Scaffolding Targeted Generalization in Natural Language Processing

Ritam Dutt

January 2024

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

Thesis Committee:
Carolyn Penstein Rose
David Mortensen
Daniel Fried
Dan Roth

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

Abstract

The holy grail of the field 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 seem finally within reach for a broader assortment of tasks and domains. However, the reality remains that current systems, designed in a task and domain-agnostic manner, are still not able to capitalize on the particulars/specific characteristics of target tasks and domains.

For many practitioners, small bespoke systems are still advantageous in terms of cost-benefit analyses. However, within the core research community, SOTA large-scale models, often designed in a sequence-to-sequence paradigm, remain at center stage. At the same time, they suffer a few shortcomings; the computational expense on multiple fronts is becoming formidable; and the transformation from raw data to input embeddings ignores some key relevant structures present in the data, which might be necessary to identify a solid basis for generalization. The end result is that crutches are learned from massive data stores that enable high performance within the training distribution, but lower beyond those frontiers.

This dissertation proposes to address both the computational expense problem and the learned crutch problem with a generalized framework that exploits different scaffolds to capture regularities between the source and target. These scaffolds are task-dependent and serve as inductive biases to facilitate generalization. We explore two main kinds of scaffolds: formal and informal. Formal scaffolds ground the information present in text to some 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 proposal, we inspect the role of formal scaffolds for information extraction in procedural text and question answering over knowledge bases (KBQA). We adopt linguistic frameworks such as dependency parses and abstract meaning representations for information extraction in procedural texts. These frameworks facilitate bridging the gap between different domains, such as cooking recipes and material science corpus and provide substantial gains in a few-shot setting. For KBQA, we leverage the schema of the underlying knowledge base to generalize to unseen entities and propose strategies to generalize to unseen schema items like relations and classes during inference.

In the second part of the proposal, we investigate the utility of LLM-generated rationales to verbalize social cues implicit in a conversation. We observe that these rationales serve as an excellent augmentation to significantly improve performance on two social meaning detection tasks both in-domain and across domains. Our proposed work explores whether these rationales will be crucial in generalizing to different social influence tasks, and whether these rationales can be generated in-house without relying on proprietary LLMs.

Finally, having shown the utility of different scaffolds, we suggest a deep dive

to understand how these scaffolds correlate with generalization performance across different cases. As a case study, we focus on natural language inference, where formal and informal scaffolds in the form of linguistic frameworks and rationales will serve as probes to analyze generalization performance across several dimensions like domains, robustness, compositionality, and the like. In a nutshell, we propose a more holistic evaluation of generalization.

Acknowledgments

.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of generalization in machine learning

In classical machine learning literature, generalization refers to the ability of systems to adapt to unseen shifts in distribution (or data) [Require 2-3 citations –RD]. We present an overview of the modelling paradigm of most machine learning systems or models in Figure 1.1 where we train a system (here as a neural network) on an observed distribution (shown in blue), that has to adapt to a new distribution (shown in green) during testing. We observe that the best curve (learned function) that minimizes the error on the training instances is not ideal for the new test instances.

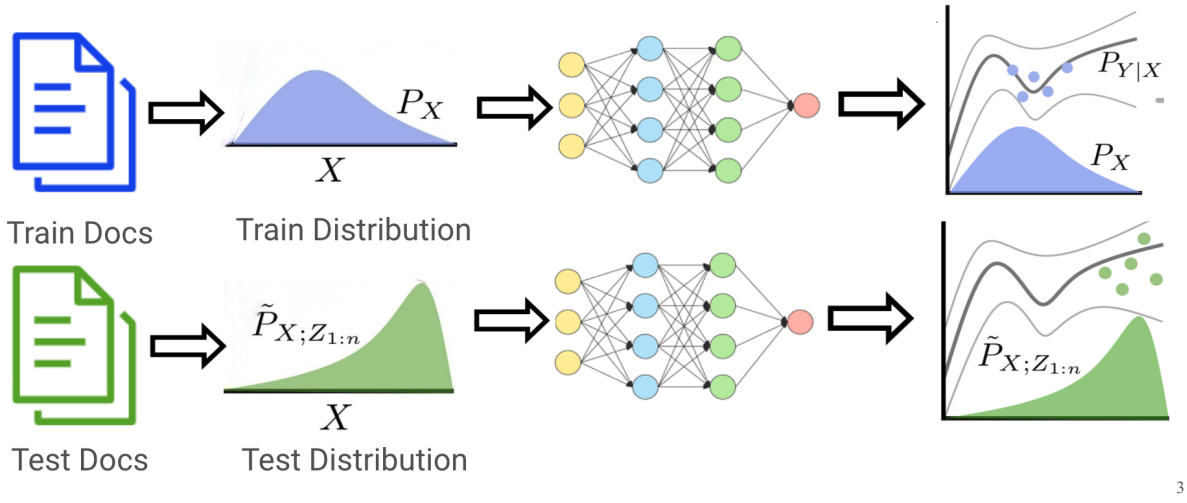


Figure 1.1: An overview of how models or systems generalize to out-of-distribution data in Fannjiang et al. [2022]

Additionally, in machine learning, generalization is characterized by the kind or type of distribution or data shifts. The prevalent shifts where the label is inferred from the data (i.e. Y is inferred from X), include (i) covariate shift Moreno-Torres et al. [2012], Storkey [2008] where the underlying data or distribution (i.e. $P(X)$) changes but the conditional of the label given the data

(i.e. $P(Y|X)$) remains the same , (ii) label shift (also called concept shift) where the distribution remains the same (i.e. $P(X)$) but the conditional (i.e. $P(Y|X)$) changes , and (iii) full shift where both the distribution and the conditional changes (i.e $P(X)$ and $P(Y|X)$ changes). A pictorial representation of these kinds of shifts are shown in Figure 1.2.

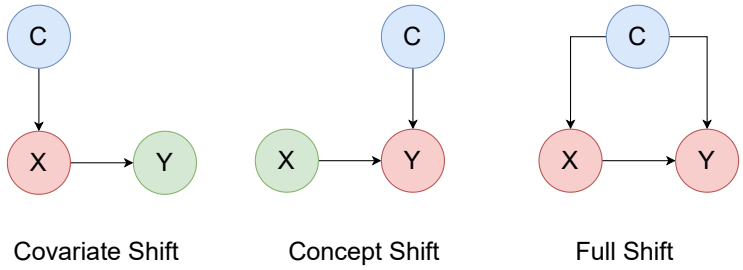


Figure 1.2: Different kinds of distribution shifts prevalant in machine learning. The red indicates the affected variable due to some external cause (C).

.

While both this overview of the generalization paradigm as well as the characterization of different kinds of shifts is crucial from a machine learning perspective, to facilitate the design of better or efficient algorithms, we argue below why it is insufficient to measure generalization for language technologies in the same vein.

1.2 Generalization in Language Technologies

1.2.1 Text as an incomplete information source

In the context of supervised machine learning, a given task is defined explicitly by providing a system with a set of input-output pairs to guide the learning process. For language technologies, the input is predominately in the form of text that needs to be processed and transformed into an intermediate representation before they are mapped into a rating or outcome. The best alignment occurs when the explicit setup or training paradigm captures the intended behaviour for a given task. Consequently, errors can creep in any of these four stages, as shown in Figure 1.3, i.e during the selection of these examples, representing the input text, mapping the input representation into a rating, and whether the training processes captures the intended behaviour or not.

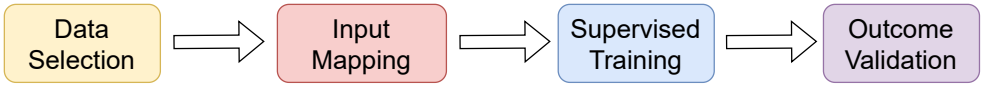


Figure 1.3: The four stages of executing a task in a supervised learning paradigm in NLP.

Recent work on machine learning and language technologies has placed a heavy emphasis on improving data selection, designing better algorithms, and validating task fidelity, [ADD CITATIONS –RD], but has taken for granted the intermediate stage of representing the input text. This has become more pronounced due to the prevalence of powerful neural models, mostly

designed in a seq2seq paradigm, to facilitate generalization for a variety of cases Raffel et al. [2020], Xie et al. [2022].

Consequently, when models adapt to different domains, the differences in distribution of the input representations are treated as natural representations of domains and as challenges that should be overcome during the supervised training stage.

1.2.2 Multi-faceted perspective of generalization

Chapter 2

Literature Review

2.1 Face Acts

Although politeness derailment and politeness evolution in dialogue have been previously investigated in the NLP literature Chang and Danescu-Niculescu-Mizil [2019], Danescu-Niculescu-Mizil et al. [2013a], the prior work is distinguished from our own in that they do not explicitly model face changes of both parties over time. Rather, Danescu-Niculescu-Mizil et al. [2013a] utilizes requests annotated for politeness to create a framework specifically to relate politeness and social power. Other previous work attempt to computationally model politeness, using politeness as a feature to identify conversations that appear to go awry in online discussions Zhang et al. [2018]. Previous work has also explored indirect speech acts as potential sources of face-threatening acts through blame Briggs and Scheutz [2014] and as face-saving acts in parliamentary debates Naderi and Hirst [2018].

The closest semblance of our work is with Klüwer [2011, 2015], which builds upon the notion of face provided by Goffman [1967] and invents its own set of face acts specifically in the context of "small-talk" conversations. In contrast, our work specifically operationalizes the notion of the positive and negative face of Brown et al. [1987], **?**, which is well established in the Pragmatics literature and heavily acknowledged in the NLP community Danescu-Niculescu-Mizil et al. [2013a], Zhang et al. [2018], Wang et al. [2012], Musi et al. [2018]. Moreover, we focus on analysing the effects of face acts in a "goal-oriented" task like persuasion, where there is an explicit threat or attack on face as opposed to small-talk scenarios, where the goal is building rapport or passing the time. Thus our work can be considered to be complementary to the prior work of Klüwer [2011] and Klüwer [2015]. It also enables us to draw insights from recent work in persuasion strategy to analyze face act exchanges in persuasion Wang et al. [2019], Yang et al. [2019].

2.2 Resisting Persuasion

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 operationalising and automatically identifying persuasion strategies Wang et al. [2019], 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. [2018a], Joshi et al. [2021]. However, there is still a dearth of research from a computational linguistic perspective investigating resisting strategies to foil persuasion.

Resisting strategies have been widely discussed in literature from various aspects such as 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]. However, these works have mostly relied on qualitative methods, unlike ours, which adopts a data-driven approach. We propose a generalised framework to characterise resisting strategies and employ state-of-the-art neural models to infer them automatically. Thus our work can be considered complementary to past research.

The closest semblance to our work in NLP literature ties in with argumentation, be it essays Carlile et al. [2018], debates Cano-Basave and He [2016], or discussions on social media platforms Al-Khatib et al. [2018], Zeng et al. [2020]. Such works have revolved mostly on analysing argumentative strategies and their effect on others.

Recently, Al Khatib et al. [2020] demonstrated that incorporating the personality traits of the resistor was influential in determining their resistance to persuasion. Such an observation acknowledges the power vested in an individual to resist change to their existing beliefs. Our work exhibits significant departure from this because we explicitly characterise the resisting strategies employed by the user. Moreover, our work focuses on the general domain of non-collaborative task-oriented dialogues, where several non-factual resisting strategies are observed, making it distinctly different from argumentation Galitsky et al. [2018]. We assert that focusing on both parties is imperative to get a complete picture of persuasive conversations.

2.3 PerKGQA

The task of KGQA has evolved from a simple-classification setting Mohammed et al. [2018] to an information retrieval paradigm Wang et al. [2020b], Saxena et al. [2020], Yasunaga et al. [2021], Sun et al. [2019], Xiong et al. [2019] that can tackle multi-hop relations or complex questions. Other approaches include 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]. We investigate graph-based information retrieval methods in this work since they achieve SOTA performance without any additional information like logical forms or semantic parses. This sets us apart from recent work on KGQA generalizability Gu et al. [2021], Chen et al. [2021] which requires such logical forms during training; information often unavailable for real-world data settings. Our work also differs from Sidiropoulos et al. [2020] which is more focused on entity-linking and relation prediction for unseen domains and leverages existing web resources, which is not applicable to us.

Most KGQA approaches that operate in an information retrieval setting over predefined (or base) knowledge graphs follow a similar procedure to make the problem computationally feasible. Sun et al. [2018, 2019], Wang et al. [2020b,a]. They first construct a smaller sub-graph for each question from the base graph, using the Personalized PageRank algorithm Haveliwala [2003]. then re-use the base graph's node representation to initialize the nodes in the sub-graph. Thus during inference, they already have prior representation of the nodes. However, in our setting, we encounter new KG during inference, and thus we need to learn the representations of those unseen nodes from scratch.

Our PATHCBR approach is closely related to 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. They then apply reasoning paths to the query entity. PATHCBR differs in two ways; (i) We operate upon complex or compositional questions and retrieve similar templates rather than entities, and (ii) We do not use a rule-based framework to generate reasoning paths. Rather, we encode the retrieved path information as an embedding and use it to score paths generated during inference to ensure generalization. In a similar vein, Das et al. [2021] uses a neuro-symbolic case-based reasoning approach for answering complex, multi-hop questions. However, their approach cannot be applied to our setting since it requires logical forms (SPARQL queries). We circumvent this requirement by designing PATHRGCN that leverages GNNs, KGs' structure, and path information between source and answers.

2.4 Few-shot 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. [2022a] 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. [2022], (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. [2022b], translated texts for multilingual RE Nag et al. [2021], or distantly supervised instances Zhao et al. [2021], Ye and Ling [2019a]. All of these techniques seek to alleviate the need of using expensive human-annotated training data. In this work, we question whether incorporating linguistic structure on existing models can aid learning robust representations which can be transferred to other domains.

2.5 Linguistic frameworks for NLP

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. [2021], 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.

2.6 Social Meaning in NLP

Social meaning is 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]). It encompasses both the linguistic agency and goals of the speaker ("the explicit") as well as their personal characteristics and dispositions ("the implicit") Goffman [2002].

While originally defined in the context of socio-linguistics, the term "social meaning" been heavily used in the computational linguistics community. It can refer to different interactional styles Jurafsky et al. [2009], or the social background and identity of a user that can be predicated from linguistic variation Nguyen et al. [2021], 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 of the same, we adopt "social meaning" as an umbrella term to refer to tasks that infer the intentions of the users or their characteristics in a social setting. Specifically, in this work we focus on two social meaning detection tasks, namely the strategies employed by an individual to resist persuasion (RES) or the emotions expressed during a conversation (ERC).

2.7 Generalization in Dialogue

Generalization in the context of dialogue tasks is a challenge because the interaction is typically organized around a task rather than the presentation of information, has multiple loci of control, and so much is implicit in it. Mehri [2022] provides an outline of different kinds 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 like 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. [2013b], Li et al. [2020a], Dutt et al. [2020]. In practice, generalizability is still challenging Khan et al. [2023], 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.

2.8 Rationales in NLP

In the context of NLP, the term "rationales" have long been used to refer to *textual explanations*, either generated by machines or humans. Rationales serve a wide variety of purposes 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], and even assisting humans in their tasks Das and Chernova [2020], Joshi et al. [2023].

Recent research has demonstrated the efficacy of LLMs in generating step-by-step explanations or rationales Gurrapu et al. [2023] that can be harnessed to bolster downstream task performance Rao et al. [2023], Wei et al. [2022b], Zelikman et al. [2022]. Rationales have also contributed to the OOD generalization of models.Majumder et al. [2022], Xiong et al. [2023], Joshi et al. [2022]

Building upon this foundation, we frame rationales as the elicited verbalization of social meaning in a conversation; they make 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. [2020] which uses rationales as means of contextualizing a task with pre-conceived social norms, whereas we use rationales to elicit the implicit intentions and assumptions of the speaker.

Chapter 3

Linguistic representations for fewer-shot relation extraction across domains

3.1 Introduction

In many specialized domains, such as healthcare or finance, one of the principal limitations for the implementation of machine learning based NLP methods is the availability of high quality data, which tends to be both time-consuming and expensive to acquire. While pre-trained language models allow impressive performance gains across a number of tasks, those gains frequently fail to generalize to specialized domains. Robust representations that allow models to both take advantage of pretrained models and generalize across domains are therefore highly desirable.

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. In addition, however, linguistic structures offer abstraction over the variation of natural language, providing representations that might express meaning in domain-general ways. We therefore extend earlier work to 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 focus on the case of automatically generated linguistic annotations, to evaluate the impact they can have on downstream tasks without expensive human annotation of parse data.

We use two linguistic formalisms, to evaluate and compare their impact: dependency parses, and abstract meaning representations (AMR). AMR [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 study the problem of relation extraction on procedural text. We use procedural text because

of what we term their *implicit schemas*. Across domains, our datasets 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 hypothesize that the underlying semantics of all of these 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 use three datasets across two domains: recipes for cooking, and materials science synthesis procedures. Each of these datasets defines the task of generating a comprehensive, descriptive graph representation of a procedure. We simplify this task into relation extraction in order to better compare the impact of different linguistic formalisms.

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.

3.2 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 3.1, and go into further detail in Section 3.2.4. 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.

3.2.1 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

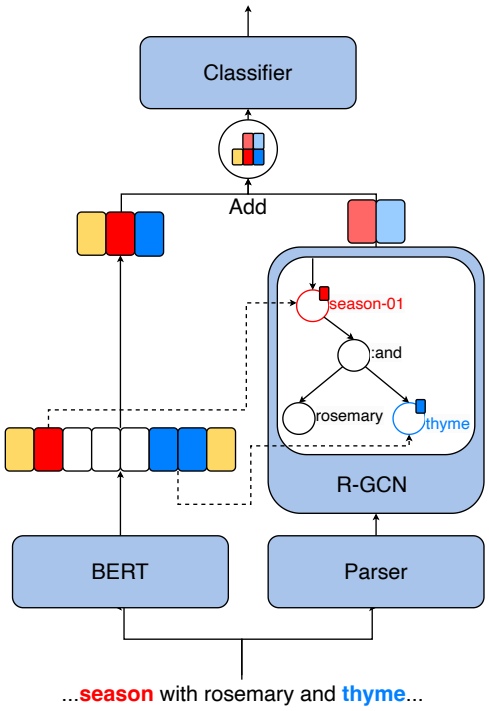


Figure 3.1: Model architecture. Yellow tokens denote BERT special tokens. Dotted lines indicate using BERT embeddings to seed the graph for the R-GCN.

heuristics to predict relations based on entity type, as observed in Rosenman et al. [2020], we exclude the entity types from the original datasets.

3.2.2 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., 2020], 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

[1] `https://github.com/bjascob/amrlib`

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.

3.2.3 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.

3.2.4 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

[2]https://amr.isi.edu/doc/amr-alignment-guidelines.html

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. [2018]). 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 3.1. The baseline architecture omits the right-hand fork, using only BERT embeddings.

3.3 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., 2020] 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" described in section 3.2.1) 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$\times$ less than the most frequent classes.

The **RISeC** dataset [Jiang et al., 2020] 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

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.

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.

3.4 Experiments

3.4.1 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 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.

3.4.2 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 $K \in$

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

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

$\{1, 5, 10, 20, 50, 100\}$. 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.

3.5 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 3.2. Overall, we do not see

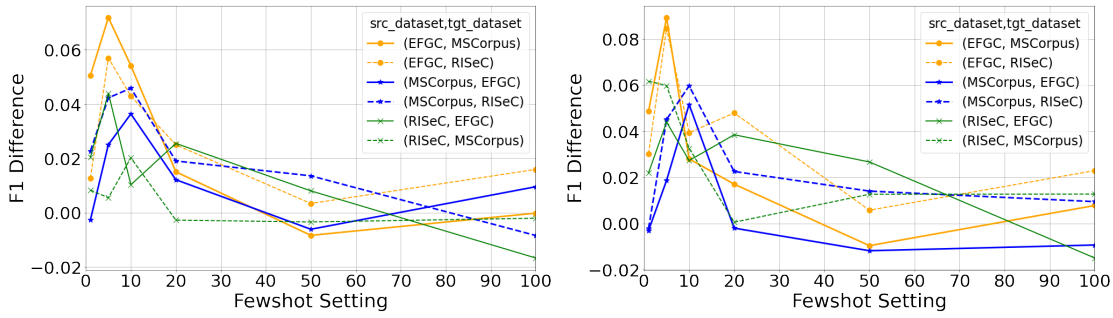
significant differences between the baseline and +Dep and +AMR cases, even though they appear to overperform the baseline case on RISeC and MSCorpus. Notably, however, these models do not overfit more than the baseline: performance on the unseen test set remains similar.

We do, however, see differences in performance between the baseline and graph-aware cases in the few-shot transfer setting. In Figure 3.2, 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 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.

			Fewshot Setting					
Target	**Source**	**Case**	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 3.3: 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.

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,



(a) +AMR vs baseline

(b) +Dep vs baseline

Figure 3.2: Differences in F1 over baseline from incorporating linguistic graphs in models.

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, 679) = 10356.25, p ¡ .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 ¡ .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 ¡ .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

			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 3.4: 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.

efficiently enabling transfer between domains. We find a significant effect of representation case: F(2, 679) = 5.26, p ¡ .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, p ¡ .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 ¡ .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 ¡ .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 3.3. 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 3.3, 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 3.4. 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 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 EFCG to RISeC case, transfer models that incorporate dependencies and AMRs overperform their in-domain counterparts by between 3 and 7 points.

3.6 Conclusion and Future Work

We experiment with using linguistic formalisms as additional context for learning robust representations that facilitate few-shot transfer among domains for the task of relation extraction. Our experiments show that the inclusion of linguistic formalisms significantly boosts models' ability

to transfer to new datasets. They additionally show that that benefit is robust to whether transfer learning helps in the baseline case. This suggests that using linguistic formalisms as a scaffold for learning in data scarce, specialized domains could be a powerful technique.

Future work could focus on several directions. With regards to the use of semantic frameworks, more work is needed to understand how best to incorporate highly abstract formalisms such as AMR. For example, how can we better use the node features in AMR, rather than just the structure? These questions also apply to more abstract syntactic frameworks like constituency parsing. With regards to our transfer learning process, we aim to understand what features of a pair of datasets make them suited to transferring by studying a wider array of datasets in diverse domains, as well as to study the impact of domain adapting our syntactic and semantic parsers to our target domains.

Chapter 4

PERKGQA: Question Answering over Personalized Knowledge Graphs

4.1 Introduction

The task of Question Answering over Knowledge Graphs (KGQA), involves answering a natural language question by querying a predefined knowledge graph (KG), such as WikiData or Freebase. 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 Kacupaj et al. [2021], 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].

Most KGQA research has focused on generalizable or generic knowledge, which assumes there is a predefined global KG for all queries. This assumes that nodes used during inference were already defined in the KG during training and holds for cases that focus on generalizable knowledge. This work proposes approaches that circumvent the need to make such an assumption.

Furthermore, using a single global KG to handle queries of different users raises additional concerns, especially when a user's query requires situated knowledge such as personal information.

- **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.

In this paper, we formulate PERKGQA or question answering over personalized knowledge graphs. Here the user has access to their specific KG, a subset of the global KG that contains only the information relevant to the user. We are restricted to the user's KG to answer their queries during training and inference. Such a setting addresses the challenges above of scalability, privacy, and generalizing over unseen KGs.

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.

We propose two approaches, PATHCBR and PATHRGCN, that are well-suited to these settings. PATHCBR is a simple non-parametric case-based reasoning approach that encodes path information of past queries to answer a new query. PATHRGCN is a parametric approach that employs graph neural networks, path information, and the KG's structure to extract answers. These approaches circumvent the need for learning prior node representations and can be readily applied to unseen KGs.

Contributions of the paper:

- We formulate PERKGQA, a new setting for KGQA where we operate over unseen KGs in the absence of any additional information. We observe that SOTA methods that need to learn underlying node embeddings fare poorly.
- To encourage research, we modify an existing academic dataset Yih et al. [2016] and make it available for research (as Mod-WebQSP).
- We propose PATHCBR and PATHRGCN, which outperform baselines on Mod-WebQSP and an internal dataset by 6.5% and 10.5% respectively.

4.2 Preliminaries

4.2.1 Task Formulation

A Knowledge Graph (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 $\mathcal{V}$.

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.

4.2.2 Running Example

We now demonstrate the applicability of PERKGQA for a cloud service provider (e.g. Microsoft Azure) in Figure 4.1. 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 (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

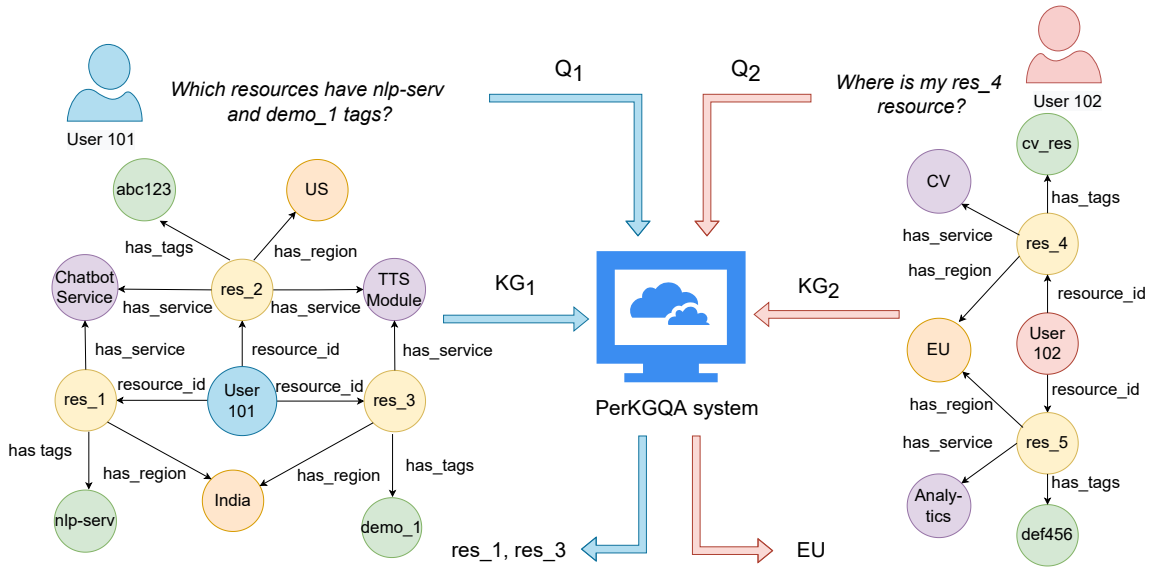


Figure 4.1: PERKGQA for a cloud service provider setting. The two users (in blue and red) create cloud resources (in yellow) in specific regions (in orange), and deploy services e.g. *Chatbot service*, or *Analytics* (in purple) on them. The users assign customized tags (in green) to the resources. Each user has their unique KG. The system should scale to support queries of new users over unseen KGs without any retraining or additional knowledge.

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 4.1, 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 4.1 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.

4.3 Datasets

We operate on two datasets: an internal dataset, CloudKGQA, built on top of CloudKG, and an academic dataset called Mod-WebQSP designed to mimic our setting. An instance in either dataset follows the same task formulation in Section 4.2, 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 linked through salient mentions of entities in q. E.g., the source entity for, "Who was responsible for Lincoln's

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

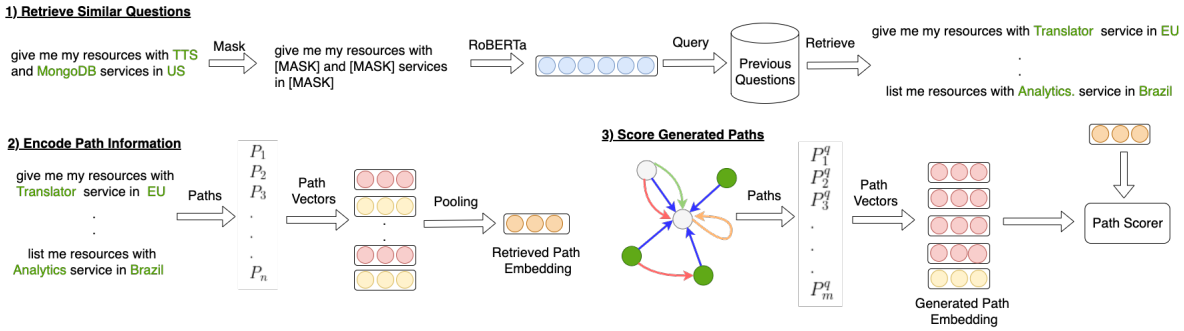


Figure 4.2: PATHCBR Overview: (1) Retrieve questions similar to a given query template from set of questions; (2) Encode path information as a path embedding; (3) Score generated paths using the retrieved path embedding.

assassination?" is the node corresponding to Abraham Lincoln.

4.3.1 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 4.1 as we present examples that outline the key characteristics of CloudKGQA.

- **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 4.1).
- **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.

4.3.2 Modified WebQSP (Mod-WebQSP)

We also operate on the publicly-available WebQSP dataset Yih et al. [2016], built over Freebase ($\mathcal{F}$). We chose WebQSP since it shares similar characteristics of CloudKGQA, namely the presence of multi-answer, multi-hop, simple and complex questions. To completely mimic our setting, we construct a KG, $\mathcal{F}_q$ for each question q, with the caveat that a significant fraction of nodes remains unseen during inference. We describe our process for creating individual KG in the Appendix **??**. Our modification achieves a low overlap of 4.0% between entities across training and test splits, implying that 96% of entities remain unseen.

4.3.3 Differences between the datasets

We present the descriptive statistics of the two datasets in Table 4.1 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 4.2. 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 4.1: 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.

4.4 Methodology

4.4.1 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 4.2) 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. [2020b]. 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

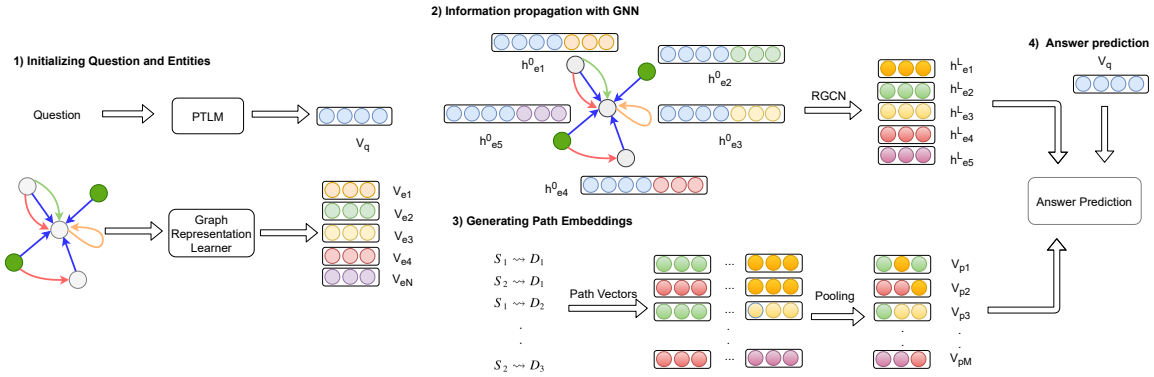


Figure 4.3: PATHRGCN Overview: (1) Initialize the question using a pretrained language model (PTLM) and the nodes in the corresponding KG; (2) Perform information propagation using RGCN to update node embeddings; (3) Encode path information from the source entities (shown in green) to all possible target nodes by pooling over the constituent node embeddings; (4) Perform answer prediction at both the path and node level.

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:

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

(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 4.5. 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:

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

(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:

$$\begin{aligned} p_j &\leftarrow [r_1, \cdots, r_{im}] \\ v_{p_j} &\leftarrow \textsc{Max-Pool}([v_{r_1}, ..., v_{r_m}]) \\ score(v_{p_j}) &\leftarrow \textsc{sim}(v_{p_j}, v_P^q). \end{aligned}$$

4.4.2 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 4.3), 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. [2020a,b], we do not use pretrained word embeddings since user-provided names can be arbitrary.

$$\begin{aligned} v_q &\leftarrow \textsc{RoBERTa}(q) \\ v_{e_1}, v_{e_2}, ..., v_{e_N} &\leftarrow \textsc{Walklet}(\mathcal{K}_q). \end{aligned}$$

(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. [2020a,b]. 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 the final representation of $h_{e_i}^L$. We use softmax as the non-linear activation and add dropout for regularization between updates. We use the RGCN model Schlichtkrull et al. [2018] to account for different relationships between nodes.

$$\begin{aligned} h_{e_i}^0 &\leftarrow v_q \oplus v_{e_i} \oplus b_i \\ h_{e_i}^{j+1} &\leftarrow \text{RGCN}(h_{e_i}^j) \end{aligned}$$

(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.

$$v_{p_j} \leftarrow \textsc{Max-Pool}(h_{e_i}^L)\ \forall e_i \in p_j$$

(iv) **Answer prediction:** We perform answer prediction both at the node and path level. We concatenate the updated representation for node e_i as $h_{e_i}^L$, 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 :

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

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.

4.5 Experiments

4.5.1 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. [2020a] 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. [2020b] 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 $\mathcal{K}_q$.

4.5.2 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]. [2] 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. [2020a], and GlobalGraph Wang et al. [2020b], we use RoBERTa-BASE Liu et al. [2019a] to encode the question, and TransE embeddings to initialize

[22] https://spacy.io/usage/spacy-101#annotations-ner

the nodes Bordes et al. [2013]. We use the publicly available PyTorch-Geometric library Fey and Lenssen [2019] to implement RGCN Schlichtkrull et al. [2018] 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] [3] 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.

4.5.3 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].

4.6 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 4.2: 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 Cloud-KGQA, outperforming the existing baselines by over 100% for Hits@1 and Accuracy in Table 4.2. 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.

[3] https://github.com/malllabiisc/EmbedKGQA

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. [2020b].

(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 4.3: 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 4.3.

(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 4.3. 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	$\underline{65.7 \pm 1.0}$	$\underline{73.0 \pm 1.1}$	$\underline{65.8 \pm 1.0}$

Table 4.4: 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 4.4.

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 4.2) 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) 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 4.4a and for Mod-WebQSP in Figure 4.5a.

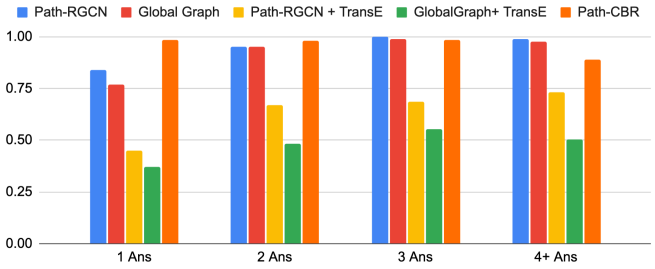
(ii) Variable size of the graph: We note the effect of for varying graph size on different methods for CloudKGQA in Figure 4.4b and for Mod-WebQSP in Figure 4.5b.

(iii) Variable Hop Distance: We investigate the performance for varying number of hops for the CloudKGQA in Figure 4.4c and for Mod-WebQSP in Figure 4.5c.

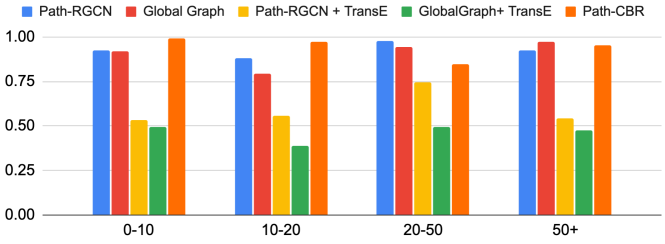
(iv) Complex Questions: We observe specifically for CloudKGQA how the accuracy across methods varies for complex questions, based on the varying number of head-nodes in Figure 4.4d and the number of logical constraints in Figure 4.4e. This information was available to us for our internal dataset but 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.

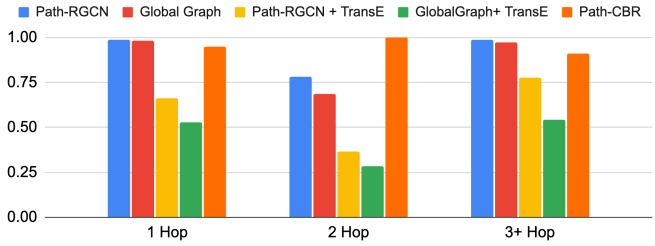
For Mod-WebQSP, we see that our PATHRGCNmodel consistently boasts the highest accuracy across all sub-cases. The trend is similar to CloudKGQA, where the PATHRGCN model can handle a larger KG size and more considerable hop distance. The only difference is the higher performance of PATHCBR when there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



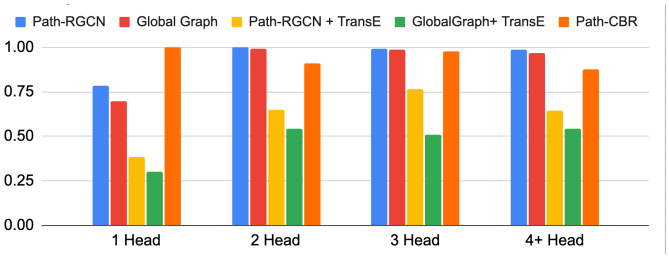
(a) CloudKGQA: Accuracy vs # Answers



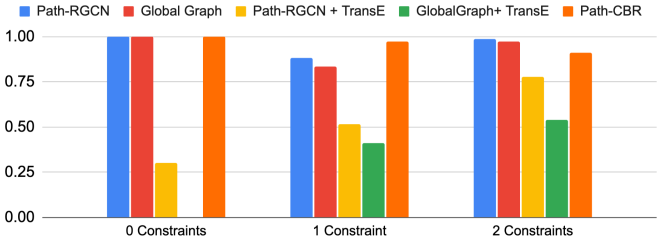
(b) CloudKGQA: Accuracy vs Subgraph size



(c) CloudKGQA: Accuracy vs # Hops

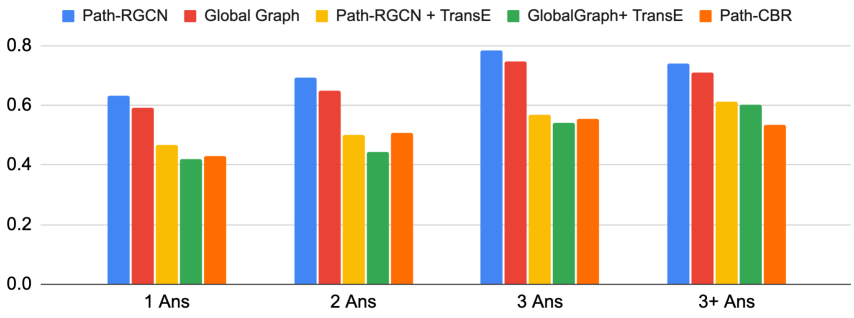


(d) CloudKGQA: Accuracy vs # source entities

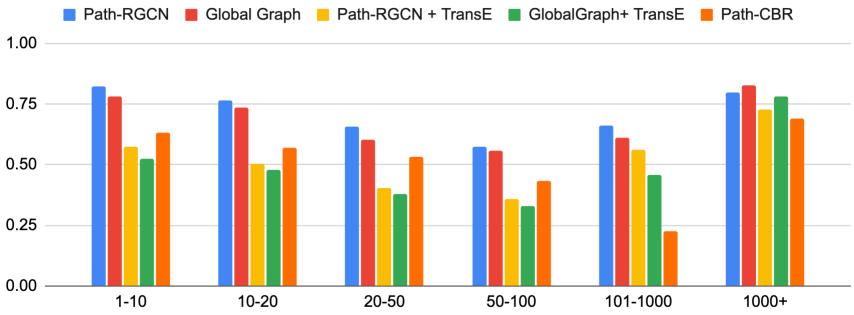


(e) CloudKGQA: Accuracy vs # Constraints

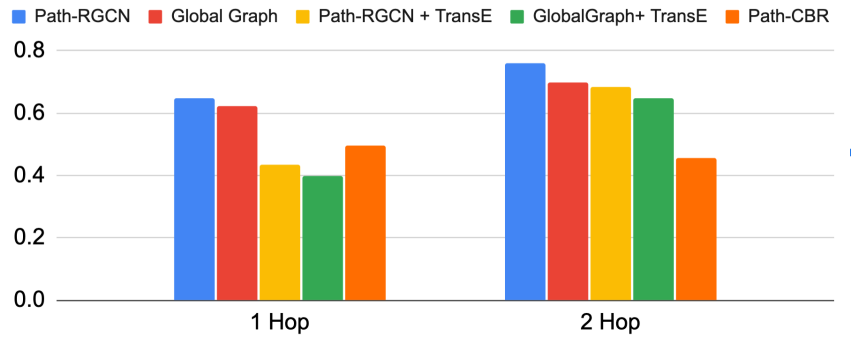
Figure 4.4: Performance of the models on the CloudKGQA dataset across different parameters such as size of the subgraph, number of answers, hops, source entities, and constraints.



(a) Mod-WebQSP: Accuracy vs # Answers



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



(c) Mod-WebQSP : Accuracy vs # Hops

Figure 4.5: Performance of the different techniques on the CloudKGQA dataset based on the number of hops, head-nodes, logical constraints

4.7 Conclusion and Future Work

We propose PERKGQA, a realistic setting for performing question answering over knowledge graphs; for each user's question, we have their corresponding KG but no additional information. Such a setting addresses the challenges of unseen nodes during inference, and prevents access to information of other users while being computationally feasible. However, state-of-the-art KGQA techniques that require learning node representations a priori fare poorly. We propose two approaches, a simple non-parametric case-based-reasoning model and a supervised neural architecture, harness path information for QA. Our approaches improve upon the baselines by 6.5% on an academic dataset and 10.5% on an internal dataset.

Having demonstrated the applicability of PERKGQA in the cloud service provider domain, we aim to explore other scenarios involving personalized or sensitive information, like healthcare. Prior work in medical NLP has focused predominately on generic or ontological knowledge such as UMLS. A personalized KG, constructed over a patient's health records, will encode information specifically for the individual and not the general population, e.g. whether the patient is allergic to certain medications. We plan to collaborate with medical professionals and create personalized KG in the healthcare domain to assist patients.

Furthermore, we seek to address certain limitations of our current approach, namely their inability to tackle spurious paths. We plan to rectify it either by explicitly providing the correct path information or incorporating some learning paradigm to detect them He et al. [2021]. Moreover, for PATHCBR, the retrieval phase is a bottleneck since one needs to compare a given query with all possible training questions and requires better indexing schemes like FAISS Johnson et al. [2019]. Likewise, the inference time for both approaches increases as the number and the length of the paths increase. However, PATHRGCN can adapt to longer paths since the node embeddings provide some degree of additional supervision. Nevertheless, we plan to explore techniques beyond embedding-based approaches, namely semantic parsing or query-graph generation, to alleviate the path-based constraint and adapt them to our PERKGQA. In the absence of gold logical forms, we plan to learn semantic parses through a weakly-supervised or distantly-supervised setting similar to Cheng et al. [2019].

Chapter 5

GrailQA++: A Challenging Zero-Shot Benchmark for Knowledge Base Question Answering

5.1 Introduction

The task of KBQA involves querying a knowledge base (KB) for a set of entities that satisfies a natural language question. Most prior work in KBQA has been restricted to an i.i.d. setting Yih et al. [2016], Talmor and Berant [2018], where the classes and relations constituting the KB remains unchanged during inference and training. However, the ubiquitous applications of KBQA in different domains such as tax, insurance, and healthcare Lüdemann et al. [2020], Huang et al. [2021], Park et al. [2020] has prompted research on KBQA generalizabilition to facilitate transfer to these domains Dutt et al. [2022], Das et al. [2021], Neelam et al. [2022], Jiang and Usbeck [2022].

The most salient work is that of Gu et al. [2021] where they propose the task of KBQA generalizability beyond the i.i.d setting, which they term "zero-shot" generalizability. In a zero-shot setting, KBQA models operate upon classes and relations which were unobserved during training. They also create a dataset called GrailQA to benchmark the generalizability of KBQA models. This dataset has garnered significant research interest with state-of-the-art KBQA models Ye et al. [2021], Yu et al. [2022a], Gu and Su [2022], Shu et al. [2022], Liu et al. [2022b] achieving remarkable performance on the leaderboard, specifically on the zero-shot setting. [1]

However, a closer inspection of the GrailQA dataset reveals that it is biased towards simpler questions and that existing KBQA systems cannot deal with complex cases in a non-i.i.d. setting. We put forward the notion of graph isomorphisms to characterize the complexity of the questions, which is similar in spirit to the idea of reasoning paths or semantic structures of Li and Ji [2022], Das et al. [2022]. We observe a pronounced skewness in the distribution of isomorphisms in the GrailQA dataset. The simplest isomorphism, where the answer is located one hop away from starting entity, comprises 78.5% of the GrailQA zero-shot samples, while a more complex isomorphism with answers three hops away accounts for only 0.53%. In this work, we leverage

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

the concept of isomorphisms to explore the generalization abilities of KBQA models on questions of varying complexity.

We propose a new zero-shot benchmark called GrailQA++ that has a balanced distribution of simple and complex isomorphisms. The dataset comprises of questions annotated by domain experts as well as questions from well-known pre-existing KBQA datasets that are built over the same Freebase database as GrailQA. We evaluate two state-of-the-art (SOTA) KBQA models on this benchmark and observe that the performance falls significantly (28.5 on GrailQA++ as opposed to 83.5 on GrailQA). Our analysis shows that this drop can be attributed partly to the skewed distribution in GrailQA and that different models fare better on different isomorphism categories.

Our contributions are the following:

- We leverage the concept of graph isomorphisms to analyze the complexity of KBQA questions.
- We create a new benchmark (GrailQA++) with complex questions to evaluate zero-shot generalizability of KBQA models.
- Our experiments show that SOTA models perform poorly on the new dataset, emphasizing that KBQA generalizability is still a challenge. [2]
- We also carry out extensive error analysis to inspect model mispredictions and non-generalizability that would serve subsequent research in creating better benchmarks.

5.2 Preliminaries

In this section, we describe the task setting and the different levels of generalization in the context of KBQA. A more detailed description can be found in Gu et al. [2022].

5.2.1 Task Formulation

Knowledge Base: We denote a Knowledge Base or a KB 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.

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, SPARQL queries, 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

[2] Our dataset is available here at `https://github.com/sopankhosla/GrailQA-PlusPlus`.

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"),

5.2.2 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.

(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.

5.3 Isomorphisms in GrailQA

In a semantic-parsing based KBQA setting, a natural language question is first converted to a logical form and then executed over the KB to yield an answer. To ensure generalization, such KBQA models need to handle different kinds of logical forms. In this section we propose a way to categorize these logical forms using the notion of isomorphisms.

5.3.1 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 5.1.

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. $\mathcal{A}_{q_j} = \psi(\mathcal{A}_{q_i})$.

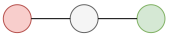
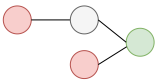
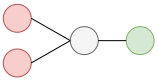
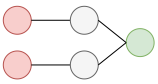
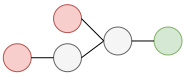
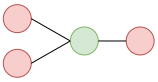
Iso-Code	Pictoral Desc.	GrailQA		GrailQA++									
				EAD		GraphQ		WebQSP		CWQ		Tot	
		Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc
Iso-0		2809	77.9	83	11.9	292	43.9	245	43.2	0	0.0	620	16.1
Iso-1		559	15.5	151	21.7	237	35.6	177	31.2	324	16.8	889	23.0
Iso-2		135	3.8	96	13.8	33	5.0	6	1.1	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 5.1: 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.

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 5.1, 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 5.7.

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 account for the fact that these graph isomorphisms can also have cycles in them. For example, in Table 5.7, we note instances of isomorphisms (CIso-0 to CIso-4) where at least one cycle is present.

5.3.2 Statistics for GrailQA

We categorize the questions in GrailQA according to the isomorphism type of the corresponding logical form. We refer to isomorphisms with fewer than 3 relations as simple and the rest as

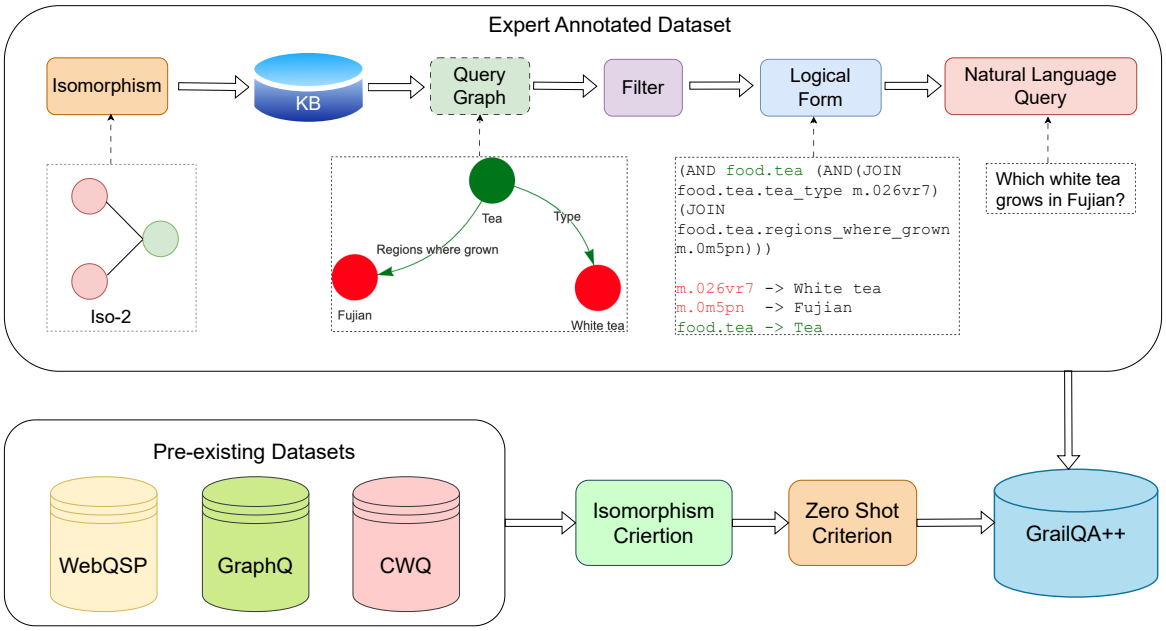


Figure 5.1: Schematic diagram that outlines the GrailQA++ dataset creation. The dataset comprises of question and corresponding logical forms, from two different sources. The former are instances which are hand-annotated by domain experts, and the latter are instances obtained from pre-existing datasets (WebQSP, CWQ, and GraphQ) which also operate over the same Freebase KB. (more details in Section 5.4).

complex isomorphisms. The simple isomorphisms for the remainder of the paper are Iso-0,1,2. We show the distribution of the isomorphisms in the zero-shot development data of GrailQA in Table 5.1.

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 5.7 in the Appendix). 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.

5.4 GrailQA++

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 5.1.

5.4.1 Expert Annotated Instances

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.[3] 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 that all three annotators agree on 75% of the

[3]Example screenshots provided in Appendix **??**.

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 5.1). All annotations were carried out by domain experts and we did not employ any crowd-workers unlike in Gu et al. [2021] for paraphrasing.

5.4.2 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).

5.4.3 Statistics of GrailQA++

We present the distribution of isomorphisms corresponding to our curated GrailQA++ in Table 5.1. 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.

	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 5.2: 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.

5.5 Experimental Setup

Baselines: We experiment with two semantic-parsing baselines for KBQA namely RNG-KBQA Ye et al. [2021] 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. [2022]. 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. [2021] 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.

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.

5.6 Results

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 5.2. We observe that models show impressive performance on GrailQA with

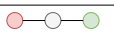
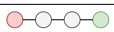
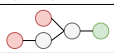
		GrailQA (Dev)		GrailQA++		EAD		GraphQ		WebQSP		CWQ	
	Iso-Codes	RNG	Arc	RNG	Arc	RNG	Arc	RNG	Arc	RNG	Arc	RNG	Arc
0		87.1/ 88.0	83.8/ 86.4	53.2/ 59.7	36.9/ 47.6	89.2/ 91.2	71.1/ 77.1	63.4/ 69.9	31.8/ 42.7	29.0/ 36.7	31.4/ 43.4	-	-
1		81.9/ 85.1	66.7/ 70.5	47.9/ 53.4	36.7/ 47.0	81.5/ 83.7	52.6/ 59.3	53.2/ 57.6	32.1/ 44.8	9.0/ 13.3	13.0/ 26.2	49.7/ 58.1	45.4/ 53.9
2		74.8/ 86.2	53.3/ 75.8	39.2/ 51.9	15.8/ 34.7	96.9/ 97.9	50.0/ 67.8	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		5.6/ 44.8	0.0/ 20.2	13.2/ 28.3	2.1/ 24.3	75.3/ 88.8	14.8/ 44.1	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		9.8/ 47.6	11.5/ 27.5	25.0/ 32.9	1.8 / 16.8	35.6/ 49.0	4.9/ 25.6	17.9/ 24.7	0.0/ 30.9	19.1/ 23.4	0.0/ 6.3	-	-
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	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	-	-	-	-	-	-	0.0/ 3.4	3.0/ 8.8
8		-	-	0.0/ 4.4	0.0/ 1.6	-	-	-	-	-	-	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	-	-	-	-	-	-

Table 5.3: 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				ArcaneQA			
Dataset	None	Count	Comparative	Superlative	None	Count	Comparative	Superlative
GrailQA (Dev)	90.1/ 90.9	91.1/ 95.3	38.6/ 73.8	0.0/ 7.8	80.2/ 83.2	68.1/ 71.7	41.2/ 65.5	72.1/ 76.5
GrailQA++	29.9/ 40.9	84.3/ 84.3	0.0/ 1.9	0.0/ 6.6	20.0/ 34.7	20.6/ 21.6	0.0/ 15.5	8.7/ 15.5

Table 5.4: 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.

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.

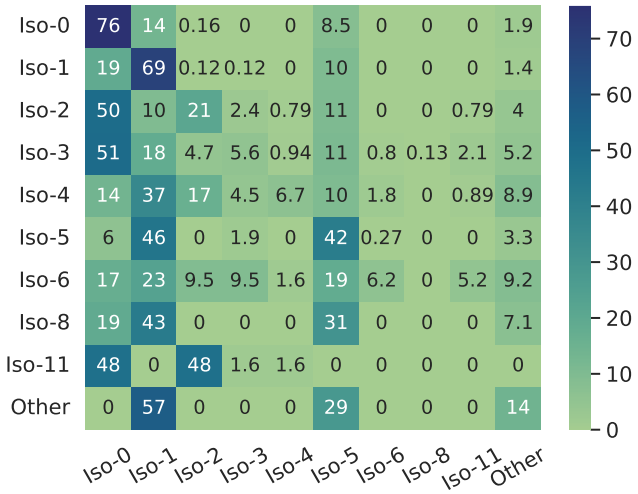
Nevertheless, even while controlling for perfect EL, the performance drops sharply on GrailQA++, resulting in an F1 score of 38.6 and 32.5 for RNG-KBQA and ArcaneQA respectively. We attribute this to the skewed distribution of ismorphisms in the original GrailQA dev split, where the simpler isomorphisms (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 5.3).

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 5.4 highlights that ArcaneQA scores higher on superlatives and comparative functions (in terms of F1 score) as opposed to RNG-KBQA for GrailQA++.

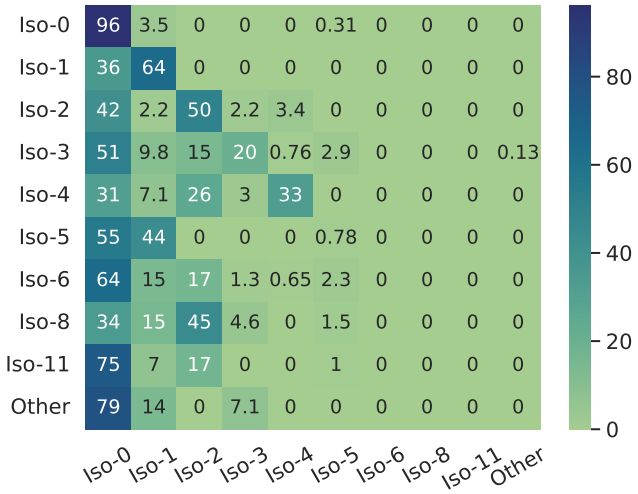
RQ2. Do models exhibit similar performance on different isomorphism types?

We present a breakdown of the model performance according to the isomorphism type for GrailQA and GrailQA++ in Table 5.3.

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



(a) ArcaneQA on GrailQA++



(b) RNG-KBQA on GrailQA++

Figure 5.2: Confusion matrices for gold Isomorphisms vs predicted Isomorphisms on the GrailQA++ dataset for ArcaneQA (top) and RNG-KBQA (bottom).

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 5.5: 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.

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 5.2.

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 5.2a) 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 5.2 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 Table5.3, 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.

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

- 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.
- Readability:We use the Flesch-reading score to characterize the readability of each question in the dataset, using the readability library in python. [4]
- 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 5.5.

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.

One interesting observation is that our constructed dataset, EAD, is most similar to GraphQues-

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

tions both in terms of the EM/F1 scores (Table 5.2) as well as coefficients of different linguistic dimensions (Table 5.5). One possible hypothesis is that both 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 5.6. All in all, we note that KBQA systems struggle with fluent and natural questions (high coherence and readability scores).

5.7 Conclusion

We propose a new dataset, GrailQA++, to benchmark the zero-shot generalization capabilities of KBQA models on complex questions. We characterize the question complexity by introducing the concept of graph isomorphisms that characterizes both the dimensions of hops and constraints. Our experiments reveal poor generalization performance of SOTA KBQA models on our proposed dataset even when gold entities are provided during inference. Our analysis also reveals complementary strengths of different KBQA models on different types of isomorphisms and how isomorphisms can be used to categorize and group model mispredictions. We also carry out extensive analysis on our proposed dataset to identify linguistic factors, that are correlated with non-generalizability such as high coherence and readability. Our research sheds light on how to design harder benchmarks to evaluate zero-shot generalization of KBQA models.

5.8 Limitations

Since the major contribution of our work is the creation of a new dataset to test the zero-shot generalization capabilities of KBQA, the limitations of the work could be stated with regards to the dataset creation. An important thing to note is that unlike other benchmarks, our proposed GrailQA++ is designed solely for the purpose of evaluation. The dataset was created keeping in mind that the only training data one has access to is the train split of GrailQA. This is because the test splits in pre-existing datasets share a huge overlap with their corresponding train splits. Consequently KBQA models trained on any additional dataset might violate the zero-shot criteria.

Additionally, we decouple the act of entity linking from question answering and thus the performance of models stated in this work are expected to be higher than using models that do not have access to gold entities. Furthermore, to aid machine understanding we avoid paraphrasing and try to construct natural language queries with explicit mention of classes and relations of interest. We intend to address these limitations in the future, but for the time being we wanted to control for complexity using isomorphisms which motivated the following design choice.

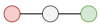
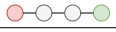
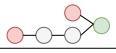
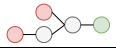
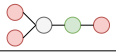
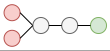
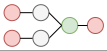
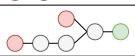
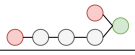
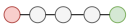
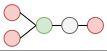
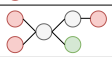
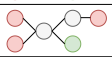
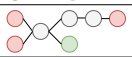
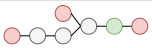
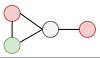
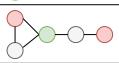
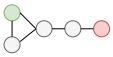
Iso-Code	Pictoral Desc	GrailQA		GraphQ		WebQSP		CWQ	
Iso-0		30496 (68.8)	2809 (77.9)	1533 (64.4)	640 (52.2)	1750 (58.3)	288 (43.8)	0 (0.0)	0 (0.0)
Iso-1		8900 (20.1)	559 (15.5)	544 (22.8)	391 (31.9)	716 (23.8)	197 (30.0)	5869 (21.8)	360 (14.5)
Iso-2		2676 (6.0)	135 (3.7)	167 (7.0)	37 (3.0)	106 (3.5)	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 5.7: 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 6

[Proposed] Enhancing inference-time zero-shot KBQA generalization with LLMs

6.1 Introduction

Of recent, with the development and proliferation of massive LLMs, there has been a trend to mould every task into a sequence to sequence (seq2seq) paradigm. Consequently, tasks that traditionally operated over or required access to structured knowledge sources like tables, databases, and graphs, have been converted to an equivalent textual format to streamline training and inference. This has led to the popularity of unified text-to-text based models Xie et al. [2022], Chen et al. [2023], Zhuang et al. [2024], Luo et al. [2023] that can operate over a wide variety of inputs. While these models have achieved competitive performance to their stand-alone-specialized counterparts, they suffer from the following drawbacks in the context of KBQA generalization.

1. The task of answering questions over a predefined knowledge base involves several steps such as recognizing the candidates of interest in the question, linking candidates to entities in the knowledge graph, retrieving a subgraph (or sub-population of schema items) from the original knowledge base, and then performing QA over the retrieved knowledge store. However, in the case of models like Unified-SKG Xie et al. [2022] and StructLM Zhuang et al. [2024], the retrieved subgraph is provided as part of the input to the model both during training and inference, and thus makes the task of KBQA exceedingly simple. Likewise, the performance of these models on unseen test instances is likely to be poor, thereby raising questions of the model's efficacy.
2. Secondly, to alleviate the stage of candidate retrieval, one might provide the entire available schema or knowledge base as input. While this strategy could work for smaller-sized or personalized knowledge bases (such as modWebQSP Dutt et al. [2022]), it is infeasible for large-scale data-sources like Freebase or WikiData that has millions of entries.
3. Finally, to reach competitive performance on par with specialized-SOTA models, one needs to instruct-tune (or supervised fine-tune) over a massive number of examples which is likely to be cost-prohibitive.

We ideally want to leverage existing off-the-shelf KBQA models and adapt them to unseen

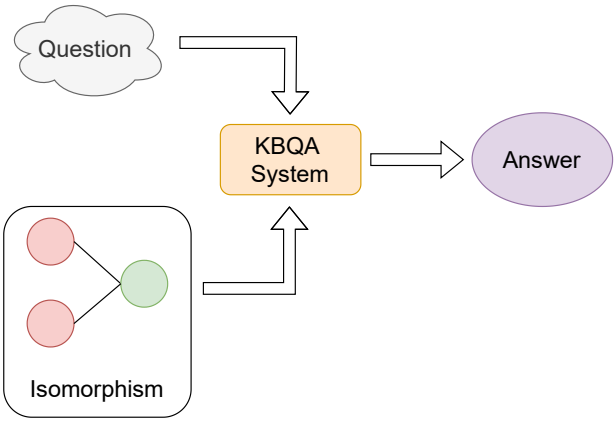


Figure 6.1: Overview of our idea of using isomorphisms for improving performance of KBQA systems.

test-cases during inference. Having demonstrated the potential pitfalls of unified systems, we propose a solution that leverages the capabilities of LLMs to improve KBQA performance.

6.2 Role of Isomorphisms

In a prior work, we had created GrailQA++ to benchmark the zero-shot generalization capabilities of different KBQA systems. We proposed the idea of isomorphisms that provides a way to characterize the complexity of a KBQA question in terms of the number of hops and constraints. It behaves similar to the idea of reasoning paths or semantic structures and provides a lens to identify which input populations are better serviced by KBQA systems. As observed in Table 5.3, we see that current KBQA systems perform well on simple isomoprhism forms but fails to generalize to more complex hops or constraints.

We acknowledge that the skewed distribution of of isomorphism categories present during training is a potential reason why systems may exhibit greater generalization prowess towards specific categories. However, as opposed to training pre-existing KBQA systems on a new distribution, we propose strategies, that uses information about the isomorphism category, to mitigate this distribution shift during inference.

Specifically, for a given question during inference, we propose to use the isomorphism category corresponding to the question to improve the retrieved/generated logical form. We can achieve this in three ways:

- For ranking based models like RNG-KBQA Ye et al. [2021] and TIARA Shu et al. [2022], we can simply filter out candidates whose logical form do not correspond with the gold isomorphism category, before they are sent to the ranker. This effectively helps to prune out the search space reducing the burden on the ranker.
- For generative models like ArcaneQA Gu and Su [2022], we plan to employ constrained decoding methods, during inference, to ensure that the generated logical form corresponds to the isomorphism category.
- For exploration based models like PANGU Gu et al. [2023], which iteratively builds up the

logical form by searching the knowledge base, we can constrain the generation to specific isomorphism category.

Model	GrailQA (dev)	GrailQA++	EAD	GraphQ	WebQSP	CWQ
RNG-KBQA	88.4/90.8	28.4/40.3	55.4/69.8	44.4/55.8	22.4/29.4	15.1/27.5
+ Gold Iso	90.6/92.8	37.2/46.3	63.3/77.4	53.8/62.8	30.5/37.8	23.7/31.5
ArcaneQA		18.5/32.5	31.5/49.9	30.2/44.8	17.6/28/7	10.2/23.2
+ Gold Iso		31.6/36.1	68.5/73.4	40.8/49.6	21.7/27.4	18.3/20.7
TIARA	86.2/88.9	29.7/40.8	56.3/71.3	47.5/57.1	22.9/29.2	15.7/27.3
+ Gold Iso	88.4/90.8	36.3/44.9	63.2/77.8	52.9/60.5	30.2/36.8	22.3/29.8

Table 6.1: EM and F1 scores for the baselines on the GrailQA and GrailQA++ datasets in absence of isomorphisms and in presence of gold isomorphisms/ classes

We observe substantial improvements in performance on our challenging GrailQA++ test-set by incorporating the isomorphism information during inference. However, a drawback of this approach is that it requires the gold isomorphism to be available during inference. In this chapter, we propose a few techniques to predict this isomorphism category that would be better utilized during inference. While this does require us to be aware of possible isomorphism classes that can exist during inference, we make no assumption about their distribution. We assume that all the isomorphism classes were seen during training.

6.3 Datasets

6.3.1 Train Dataset

For all approaches below, we aim to use the training split of the GrailQA dataset. This will ensure that the zero-shot generalization criteria holds true. Furthermore, any data augmentation strategies that we propose, will also operate upon only the observed set of classes and relations present in GrailQA's training split.

6.3.2 Inference Datasets

We evaluate the generalization performance of existing KBQA systems while incorporating isomorphism information (either gold or predicted) on the two datasets that conform with the zero-shot generalization criteria of the original GrailQA. This includes the zero-shot split of the GrailQA's development set and our curated GrailQA++ dataset. To decouple the act of entity linking and KBQA, we will consider for all cases that the gold entities are available to us both during training and inference.

6.4 Models

In this section, we propose the following techniques to predict the isomorphism category for a given question.

6.4.1 Finetune LLMs

The most obvious and straightforward way is to fine-tune a language model on the GrailQA training set to predict the isomorphism category from the natural language question. We want to focus on a few-different categories of LMs such as:

- Pre-trained LMs like BERT Devlin et al. [2019c] and T5 Raffel et al. [2020].
- LMs that were pre-trained on knowledge bases for the task of link prediction like KG-T5 Saxena et al. [2022]
- Open-sourced LLMs like LLama Touvron et al. [2023] in either a few-shot prompting or a fine-tuned setup.

Figure 6.2: Isomorphism prediction using language models.

6.4.2 Train GNNs

Since the task of isomorphism prediction involves reasoning over the knowledge base schema, graph neural networks or GNNs afford a natural means to formalize this reasoning process. For a question, whose entities are already known apriori, we will extract the corresponding subgraph from the Freebase KB. We will then train the GNN module (such as RGCN or RGAT) on the retrieved subgraph to identify the salient nodes and edges. We use the training split of GrailQA, which has the corresponding query graph for each question. Having identified the important nodes, we will aggregate their representation to predict the isomorphism category.

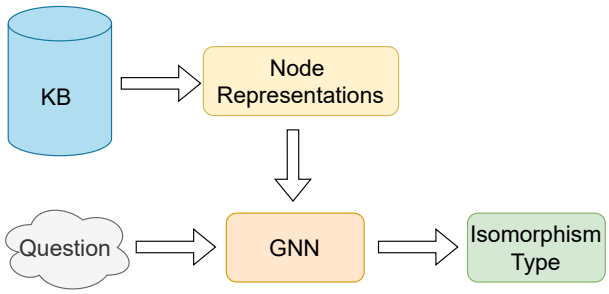


Figure 6.3: Isomorphism prediction using GNNs.

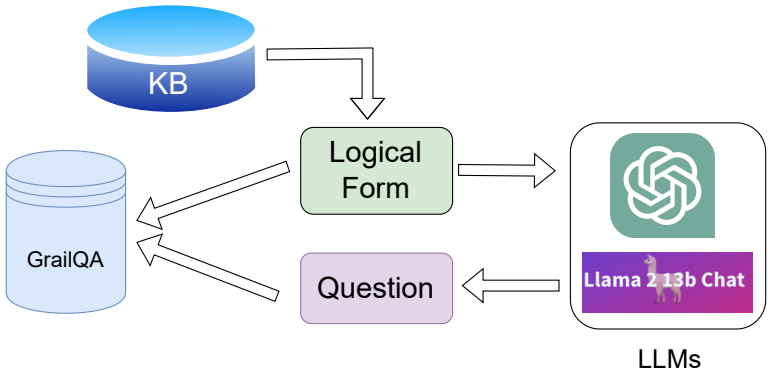


Figure 6.4: Using data augmentation to generate additional pairs of natural language questions and their corresponding logical forms.

6.4.3 LLMs for data augmentation

A shortcoming of the original GrailQA dataset is that the isomorphism distribution is biased towards simple questions. Consequently, more challenging or complex isomorphisms that occur infrequently in the dataset serve as distributional bottlenecks for the downsteam KBQA systems. One way to circumvent this bias, is to leverage the capabilities of LLMs to generate additional data from the Freebase KB.

We will follow a data collection strategy similar to our prior work while creating GrailQA++ Dutt et al. [2023]. Firstly, we will sample query graphs corresponding to a particular isomorphism category from the KB, with the additional constrain that the classes and relations that appear in the query should also be present in the original GrailQA training dataset. This extracted query graph can then be converted into its equivalent logical form using the deterministic algorithm of [cite yu su –RD]. Finally, we will use this equivalent logical form to generate the corresponding question in natural language using LLMs. LLMs have shown to be effective in converting structured language or queries into natural language, and we aim to exploit this generalization capability to generate additional data. [cite byok, gain –RD]

This augmented data can then be used in addition to the original GrailQA dataset to either finetune LMs or train GNNs, as described in the aforementioned two sections.

6.4.4 LLMs for retrieval

Another alternative to using LLMs to predict the isomorphism category directly is to use them to predict the possible skeleton of the retrieved logical form. This technique is derived from the generate-then-retrieve approach of Luo et al. [2023] where the authors first train a LLM to generate a set of logical forms given a natural language question. They observed that while the generated logical form has an exact match of 74% with the ground trurth query, if one abstracts out the names of classes and relations, the performance shoots up to 91%. This demonstrates the ability of LLMs to predict the isomorphism category with reasonable accuracy.

Chapter 7

Leveraging Machine-Generated Rationales to Facilitate Social Meaning Detection in Conversations

7.1 Introduction

> "All the world's a stage, and all the men and women merely players." Shakespeare [1623]

Beyond content focused areas of Natural Language Processing (NLP), the past two decades have witnessed a surge of interest in modeling language from a social perspective Nguyen et al. [2016]. 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].

Since social meaning is subtly encoded, traditional classification models often over-fit to context-specific linguistic elements that correlate with these subtle cues within context. Consequently, this makes transfer to unseen domains especially challenging. For example, the same strategy to resist being persuaded 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]. In this work, we propose a generalizable framework that leverages Large Language Models (LLMs) for detecting different kinds of social meaning in conversations.

We systematically investigate the generation of "rationales" by LLMs, that are designed to break through the opaque surface form of the conversation's text and make the social cues more transparent. While rationales have been utilized previously, to facilitate reasoning Rao et al. [2023], Zelikman et al. [2022], or to explain model predictions Wiegreffe et al. [2021], we use rationales to refer to the elicited social meaning, i.e. why and how an utterance was conveyed in

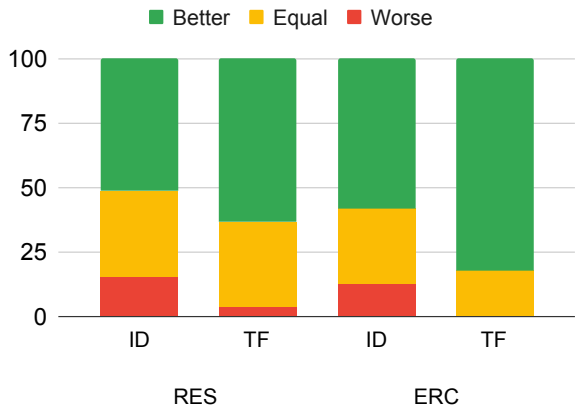


Figure 7.1: Fraction of cases where the classification performance was better, same, or worse, when rationales were augmented, for different tasks, i.e. Resistance strategies (RES) and Emotion Recognition (ERC) and settings i.e. in-domain (ID) and transfer (TF).

dialogue.

Our empirical study examines the role of augmenting rationales for two specific social meaning detection tasks: (i) Resistance Strategies (RES), which aligns with intentional and purposeful communication, and (ii) Emotion Recognition (ERC), which is characterized by habitual and subconscious responses. For each of these tasks, the evaluation is conducted over two separate corpora (different domains), but the same social meaning detection task. And thus we present results both for the in-domain (ID) and transfer (TF) settings. We illustrate in Figure 7.1 that baseline models performed significantly worse than their rationale-augmented counterparts for both tasks and settings. Our contributions are as follows :

- We investigate the role of rationales to convey social meaning by making explicit the subtle cues implicitly encoded during a conversation.
- We design a multi-faceted prompting framework, grounded in sociolinguistic theory, to generate rationales of high quality.
- We demonstrate the positive impact of adding rationales for two social meaning detection tasks across several models.
- We observe that rationales lead to greater performance gains in a cross-domain setting, especially in low data regimes, thereby highlighting the generalizability of our approach.

We make the datasets augmented with rationales and code public to encourage future research, especially for the purpose of developing open-source solutions that achieve the same functionality as the proprietary LLMs that perform best in our studies.

7.2 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 7.2.

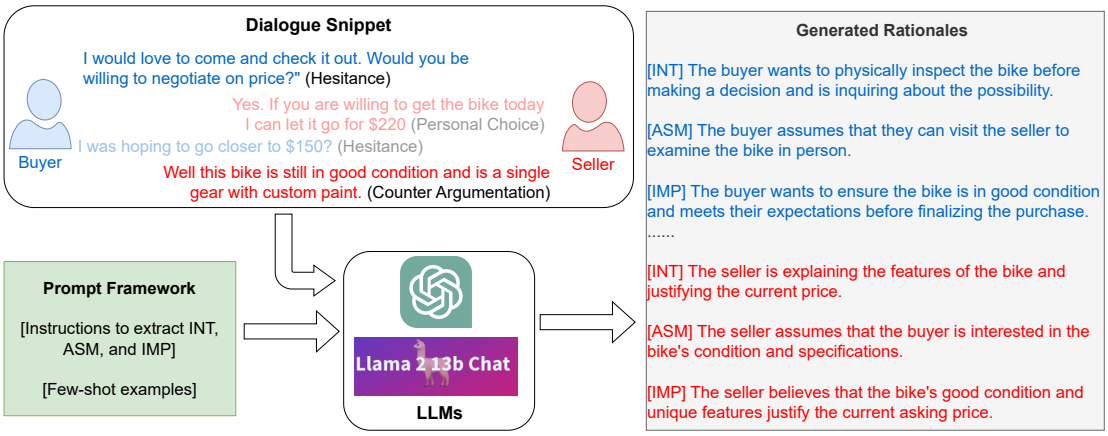


Figure 7.2: We present the prompting framework employed in this work to generate rationales that are subsequently used for dialogue understanding and transfer using pre-existing LLMs such as GPT-3.5-turbo and LLama-2 variants. We feed in the prompt (green box on the left) for a given dialogue to generate 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).

7.2.1 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.

(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.

7.2.2 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. [2022] 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 **??** and **??** in the Appendix.

7.2.3 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 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].

7.2.4 Validity of Generated Rationales

Table 7.1: 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 7.1 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

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

Table 7.2: 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

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 (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.

7.3 Experimental Setup

7.3.1 Datasets

We explore two social meaning detection tasks, namely emotion recognition in conversations or ERC Hazarika et al. [2018, 2021] and resisting strategies detection or RES Dutt et al. [2021] . We formulate both ERC and RES as utterance classification tasks, i.e. we categorize an utterance into one of several labels (8 for both ERC and RES), given its corresponding conversational context. Each task is realized via two representative datasets namely "Friends" Hsu et al. [2018] and "IEMOCAP" Busso et al. [2008] for ERC and the modified variants of the "P4G" and "CB" datasets created by Dutt et al. [2021] for RES.

For each task, the corresponding datasets (IEMOCAP and Friends for ERC, and P4G and CB for RES) operated over the same set of labels, but they exhibit different distributions (see Figure **??** in the Appendix). Thus the two datasets for both tasks exhibit a natural covariate shift making them prime candidates to investigate transfer. Furthermore, for RES, although the meaning of a given strategy remains invariant across domains, their semantic interpretation or instantiation

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

depends on the context. E.g., skepticism towards the charity in P4G and criticism of the product in CB constitutes the same resisting strategy Source Derogation.

We provide a definition for each of the eight emotions and resisting strategies along with examples for RES and ERC in Table **??** and Table **??** of the Appendix respectively. We also note the fraction of instances, for which the generated rationales were valid. We assess validity based on whether the response was a non-null string, had the appropriate speaker as its subject, and had information of all three rationales (i.e. INT, ASM, and IMP). We observe that valid generations account for $\approx 95\%$ of P4G, IEMOCAP and Friends .

7.3.2 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.

7.3.3 Models and Metrics

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

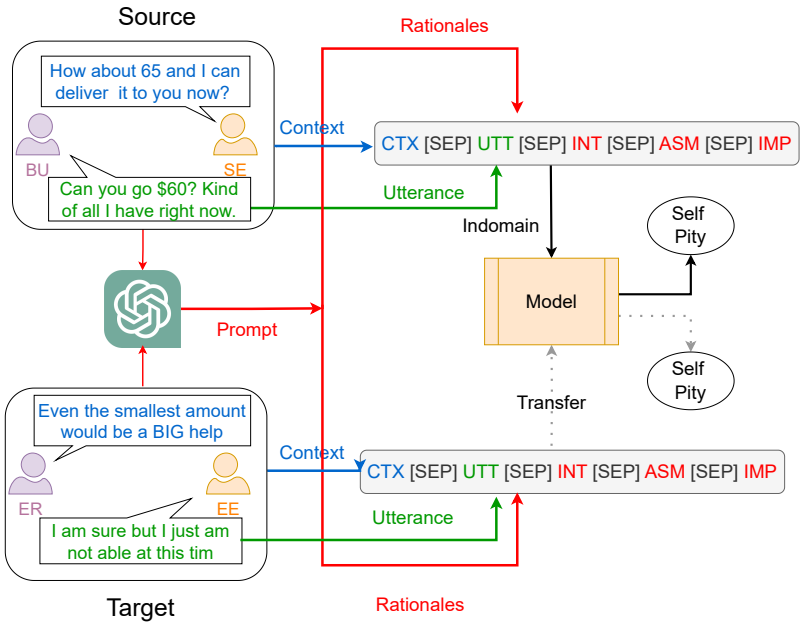


Figure 7.3: Here we illustrate the process of transfer from the source to target. The model is first fine-tuned on the source dialogues, which comprises the current utterance, the previous dialogue context, and the rationales (INT, ASM, and IMP for intentions, assumptions, and implicit information respectively). 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.

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 **??** 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].

7.4 Results

Table 7.4: 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 7.4. 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 7.5: 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 7.4. We restrict our findings to rationales

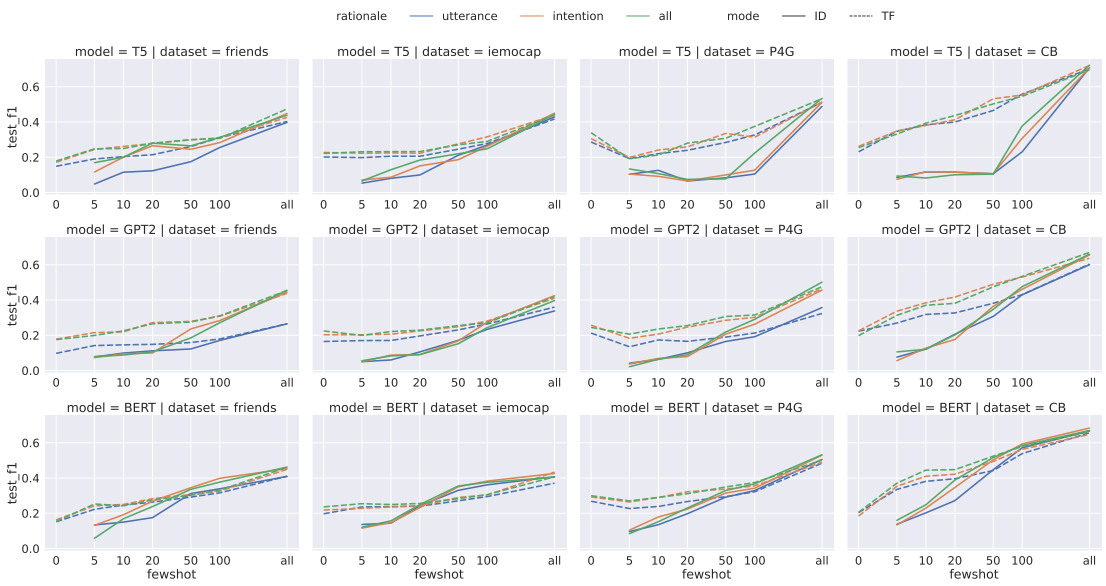


Figure 7.4: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets for different few-shot examples. The solid and dashed lines correspond to the indomain (ID) and transfer (TF) case respectively.

corresponding to intention (INT) and combination of all three (ALL) because they had the highest performance in Table 7.4. Our complete set of results are relegated to Figure **??** 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. [2023]. 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 7.5. 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.

7.5 Qualitative Analysis

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 7.4 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 **??** and cases

where their presence consistently degrades performance in Table **??**. In the following analysis we refer to instances in these Tables in the Appendix.

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

7.6 Conclusion and Future Work

We present a generalizable framework that leverages machine-generated rationales from LLMs to deduce the underlying social meaning embedded in conversations. We observe that augmenting pretrained models with the generated rationales significantly improves performance over the baseline across multiple datasets for both the tasks of emotion recognition and detecting resisting strategies. The gains are pronounced during cross-domain transfer across both zero-shot and few-shot settings thereby highlighting the generalizability of our approach. While our current

work place emphasis on domain adaptation, we believe the proposed approach is generalizable to new social meaning detection tasks (persuasion, empathy, argumentation) which we defer for future work. Furthermore, as opposed to leveraging an LLM, we intend to deploy or instruct-tune smaller models that can generate these rationales Rao et al. [2023], Zhou et al. [2023].

7.7 Limitations

Some of the main limitations of our work include:

(i) Reliance on closed-source or proprietary LLMs to generate rationales. Consequently we are not able to assure the reproducibility of generating the rationales or whether the service will be discontinued. We do however, release the entire dataset of rationales for public use for reproducibility.

(ii) We note that our proposed framework of generating rationales for fine-tuning a smaller model can be deemed more expensive than approaches that just prompts the LLM for an answer while generating these rationales during inference Wei et al. [2022c]. However, one of our contributions was to demonstrate that our approach is indeed possible. In a future work, we intend to use our created dataset, to instruction tune a smaller LM, like Flan-T5 Chung et al. [2022] to generate these rationales in-house. Prior work has demonstrated the reliability of this approach Rao et al. [2023], Zhou et al. [2023], and we intend to follow up in a future work for other social meaning detection tasks like persuasion, negotiation, empathy amongst others.

(iii) Recent studies, including Zhou et al. [2022], Sclar et al. [2023], Leidinger et al. [2023], highlight prompt sensitivity and the influence of prompt choice on downstream tasks. Our manual evaluation of 80 GPT-3.5-generated rationales, using our selected prompts, indicates they are of sufficient high quality. Potential prompt optimization avenues may exist for further enhancing rationale quality, but we defer exploration to future work.

(iv) Our choice to limit investigation to two datasets and three models is a deliberate one aimed at managing computational resources. Even within this constrained framework, we conduct 2340 experiments, highlighting the substantial computational demands of our analysis.

(v) We employ GPT-3.5-generated rationales in our study. However, we remain uncertain about their status as the ideal rationales for this purpose, or which kinds of rationales are the most effective towards this particular task.

Chapter 8

[Proposed Work] Investigating the generalizability of rationales in social conversations

8.1 Introduction

The desire to model or understand language (and human behaviour) from a social perspective has led to the formalization and subsequent adoption of several socio-linguistic principles or frameworks. Some seminal ones include the politeness framework of Brown et al. [1987], the co-operative principles and maxims of Grice, and the appraisal theory or Martin and White [Validate and add citations –RD]. These paradigms have contributed to and spurred research across several domains like education, psychology, sociology, and language technologies.

Since these frameworks are designed at a high pragmatic level, it is challenging for language technologists to computationally operationalize them directly to a new conversational scenario. Furthermore, such scenarios, might require a more nuanced understanding of the interactions at play, than that can be afforded by these frameworks, leading to the adoption of domain or task specific frameworks inspired from psychology and sociology. Some of these operationalizations include persuasion and negotiation strategies, dimensions of morality and empathy, kinds of argumentation, amongst others. We refer the reader to Chawla et al. [2023] for a comprehensive survey of different kinds of social influence in conversations.

In our prior work, we have illustrated the efficacy of incorporating rationales to facilitate generalization of two different tasks across two different domains, namely identifying resisting strategies and recognizing emotions in conversations. In this chapter, we investigate the ability of machine-generated rationales to generalize to different conversation scenarios or tasks that deal with different kinds of social influence. We analyse the importance of these rationales from two cases or perspectives; (i) from an utterance level, and (ii) from a conversation or dialogue level. The former deals with classification tasks for each utterance, whereas the latter deals with classification of a given conversation snippet.

8.2 Datasets

8.2.1 Utterance Classification

Dataset	Task	Labels
P4G	Persuasion	Logical appeal, Emotion appeal, Credibility appeal, Foot-in-the-door Self-modeling, Personal Story, Donation information, No Strategy Source-related inquiry, Task-related inquiry, Personal-related inquiry
CaSiNo	Negotitation	Self-need, Other-need, Vouch-fair, Promote-coordination,
WikiAttack	Argumentation	Comment is a personal attack, aggressive, or toxic. Show-empathy, Small-talk , or non-strategic
Diplomacy	Deception	Caught, Deceived, Cassandra, Straightforward
Therapy	Empathy	Comment expresses emotion, communicates understanding , and /or explores feeling not explicitly stated
MRFC	Morality	Non-moral, Care, Fairness, Loyality, Authority, Purity, Thin Morality

Table 8.1: Caption

8.2.2 Dialogue Classification

- **Generating the rationales in-house:** We acknowledge that our proposed approach of generating rationales by prompting proprietary LLMs may be inaccessible and expensive. Hence, given our curated dataset of rationales, we intend to instruction tune a smaller LM, like Flan-T5 Chung et al. [2022] or smaller instruct variants of open-source LLMs like LLamaTouvron et al. [2023] to generate these rationales in-house. Prior work has demonstrated the reliability of this approach Rao et al. [2023], Zhou et al. [2023], and we wish to observe how well can these rationales be generated without reliance on closed-source LLMs.
- **Role of rationales on other social influence tasks**

8.3 Proposed Experiments

Chapter 9

[Proposed Work] Evaluating Generalization

While the ability of models to generalize to out-of-domain distributions is a key desiderata of most NLP systems, this prowess has been measured in terms of absolute performance improvements on the unseen test set. Consequently, this narrow perspective of model generalizability boils down to a single metric or score. In this thesis, we wish to challenge this pre-conceived manner of measuring generalization and propose the following perspectives towards holistic NLP generalization.

9.1 Domain Robustness

9.1.1 Background

In a recent work, Calderon et al. [2023] has characterized the difference (or drop) in performance of the models in a cross-domain setup as opposed to their corresponding indomain setting across a wide-variety of NLP tasks such as NLI, sentiment analysis, QA, and over several fine-tuned and few-shot models.

They measure the in-domain performance of a model on the given source and target dataset (denoted by $\mathcal{S}$ and $\mathcal{T}$ respectively), as well as the cross-domain performance of the model when trained on the source and evaluated on the target (denoted by $\mathcal{ST}$). Consequently they are able to compute the difference (which they refer to as drop) in cross-domain performance of the model from the indomain performance on the source and target, i.e., $\mathcal{SD}$ or source-drop and $\mathcal{TD}$ or target-drop respectively.

Since their experiments consider all pairs of domains as a viable (source, target) pair, the expected performance of the model in an indomain setting is the same for all source and targets (i.e. $\mathbb{E}[\mathcal{S}] = \mathbb{E}[\mathcal{T}]$). The expected value of the source drop or target drop remains the same, i.e. $\mathbb{E}[\mathcal{SD}] = \mathbb{E}[\mathcal{TD}]$ which they term as average drop or $\bar{\Delta}$. They also compute the worst possible value of the source diff or target diff as W_{SD} and W_{TD}.

For all the experiments, these metrics are computed for a **specific task, metric, and model** in mind. Calderon et al. [2023] observed that the diff in cross-domain performance was better correlated with the target drop than the source drop.

[Write a bit more about their observations –RD]

Notation	Definition
$\mathcal{S}$	Performance of the M_t on $\mathcal{S}_t$ using μ_t. (source performance)
$\mathcal{T}$	Performance of the M_t on $\mathcal{T}_t$ using μ_t. (target performance)
$\mathcal{ST}$	Cross domain performance of M_t, using μ_t (i.e. trained on $\mathcal{S}_t$ and tested on $\mathcal{T}_t$)
$\mathcal{SD}$	$\mathcal{S}$ - $\mathcal{ST}$ (Diff in performance from the source.)
$\mathcal{TD}$	$\mathcal{T}$ - $\mathcal{ST}$ (Diff in performance from the target.)
$\mathcal{IC}$	$\mathcal{S}$ - $\mathcal{T}$ (Performance difference between source and target).
$\overline{\mathcal{S}}$	$\mathbb{E}[\mathcal{S}] = \mathbb{E}[\mathcal{T}]$ (Mean performance in an indomain setting).
$\overline{\mathcal{ST}}$	$\mathbb{E}[\mathcal{ST}]$ (Mean performance in a cross domain setting).
$\overline{\Delta}$	$\mathbb{E}[\mathcal{SD}] = \mathbb{E}[\mathcal{TD}]$ (Mean Diff in performance .)
W_{SD}	$\max_{(S_t,T_t)} \mathcal{SD}$ (Worst Diff in performance from the source)
W_{TD}	$\max_{(S_t,T_t)} \mathcal{TD}$ (Worst Diff in performance from the target)

Table 9.1: Metrics proposed by Calderon et al. [2023] for measuring domain robustness.

9.1.2 Proposed Metrics

In our proposed work, we also investigate a few metrics that aims to normalize the performance diff conditioned on the original performance of either the source or target domain, in addition to the ones proposed in Calderon et al. [2023].

Notation	Definition
$\mathcal{NSD}$	$1 - \frac{\mathcal{ST}}{\mathcal{S}}$ (Normalized Diff in performance from the source.)
$\mathcal{NTD}$	$1 - \frac{\mathcal{ST}}{\mathcal{T}}$ (Normalized Diff in performance from the target.)
$\overline{\mathcal{NSD}}$	$\mathbb{E}[NSD]$ (Mean of the normalized Diff in performance from the source)
$\overline{\mathcal{NTD}}$	$\mathbb{E}[NTD]$ (Mean of the normalized Diff in performance from the target)
$\sigma_{\mathcal{NSD}}$	Standard deviation of the normalized Diff in performance from the source
$\sigma_{\mathcal{NTD}}$	Standard deviation of the normalized Diff in performance from the target
$W_{\mathcal{NSD}}$	$\max_{(S_t,T_t)} \mathcal{NSD}$ (Worst normalized Diff in performance from the source)
$W_{\mathcal{NTD}}$	$\max_{(S_t,T_t)} \mathcal{NTD}$ (Worst normalized Diff in performance from the target)

Table 9.2: Metrics proposed by us for characterizing domain robustness.

We justify why this might be a worthwhile endeavour with an example. Imagine there exists two models, A and B which have a score of 0.40 and 0.20 on the source domain. On the other hand, when evaluated on the target domain , their corresponding cross-domain performances ($\mathcal{ST}$) are 0.50 and 0.28 respectively. Now, while model A seems to have a higher performance gain 0.10 as opposed to 0.08 of B, their relative performance gains are different 25% and 40% respectively. Thus, model B has a greater capacity to generalize on the domain, despite having a lower model capacity than A on the source domain. Another thing to note here is that the expected normalized performance diff from the source is different from the target i.e. $\mathbb{E}[\mathcal{NSD}] \neq \mathbb{E}[\mathcal{NTD}]$.

9.1.3 Proposed Experimental Setup

We propose the following experimental setup in terms of tasks, datasets and systems.

Tasks: We wish to explore the following three tasks namely natural language inference (NLI), question answering (QA), and social meaning detection tasks (SMD) like emotion recognition or resisting strategies detection.

Datasets:

- NLI: We consider five domains from MNLI dataset Williams et al. [2018] i.e. Fiction, Government, Slate, Telephone, and Travel.
- QA: We partition the SQUAD dataset Rajpurkar et al. [2016] into six different categories Geography, History, Philosophy, Science, Society, and Technology. We use the same splits as Calderon et al. [2023].
- SMD: For emotion recognition, we use datasets like Friends Hsu et al. [2018], IEMOCAP Busso et al. [2008], and MELD Poria et al. [2019]. Likewise, for resisting strategies, we can use the modified versions of the P4G and CB splits using Dutt et al. [2021].

Systems:

- **Architectures:** We wish to 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 model type we include RoBERTa Liu et al. [2019b], BERT Devlin et al. [2019a] and DeBERTa He et al. [2020] for EO, GPT-2 Radford et al. [2019] and OPT Zhang et al. [2022] for DO, and T5 Raffel et al. [2020] and BART Lewis et al. [2020] for (ED).
- **Sizes:** We also plan to experiment with varying model sizes such as small, base, and large variants for each of those models, wherever applicable.
- **Parameter-efficient models:** We also wish to explore how different parameter-efficient techniques like adapters Houlsby et al. [2019] and LoRA Hu et al. [2021], can influence domain robustness.
- **Few-shot prompting methods:** We will also explore different few-shot prompting strategies using different LLMs such as LLama2 Touvron et al. [2023] and GPT-3.5 [require citation –RD].

9.2 Multi-faceted Generalization Evaluation

9.2.1 Background

While the previous section deals with investigating generalizability from the perspective of domain robustness; we emphasize that generalization in the domain of NLP is mainly a multifaceted endeavour. A recent study of Hupkes et al. [2023] proposes a taxonomic categorization of generalization in NLP highlighting the different motivations behind generalization studies, the

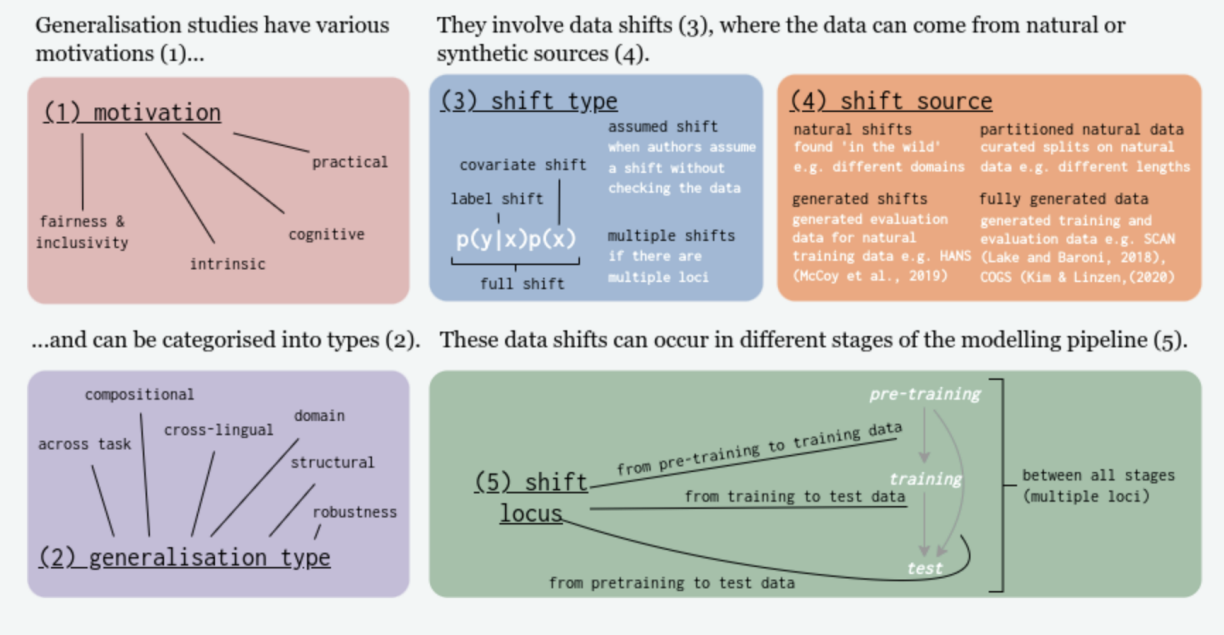


Figure 9.1: The different perspectives of generalization according to Hupkes et al. [2023]

nature/kind of data shifts, the source of the shift, as well as the stage of the modelling pipeline where the shift can take place.

Additionally, generalization in NLP encompass different tasks, domains, languages, compositions of inputs, and adversarial constructions. A pictorial representation of the different kinds of representations appear below in Figure 9.1. Inspired by this categorization, we propose to evaluate generalization across the different categorization types.

9.2.2 Proposed Research Questions

We put forward the following research questions or hypothesis in this section.

- How well does a particular model instantiation generalize across different dimensions, i.e. does the generalization ability of a model remains consistent across different axes?
- Can we extrapolate the aforementioned hypothesis for specific model classes or architectures?
- How well do certain factors like size, parameter-efficient techniques, different few-shot instances impact the overall generalization abilities?
- Do prompt-tuned models exhibit the same characteristics as these fine-tuned architectures?

9.2.3 Proposed Experimental Setup

We will follow the same experimental setup as before and focus on the tasks of natural language inference, QA, and social meaning detection in conversations.

NLI: For NLI, we consider SNLI Bowman et al. [2015] as the source domain and then carry out evaluation on different datasets that conform with the corresponding generalization

dimension. These include MNLI Williams et al. [2018] and MedNLI Romanov and Shivade [2018] to validate how well models generalize to different domains. We also use datasets such as CNLI Kaushik et al. [2019], SNLI-hard Gururangan et al. [2018], and HANS McCoy et al. [2020] to measure robustness. Finally, we use the datasets SETI Fu and Frank [2023], MoNLI Geiger et al. [2020], and ConJNLI Saha et al. [2020] to characterize generalization to different grammatical compositions.

QA: For QA, we will use the entire SQUAD dataset as the source domain and carry out evaluation on NewsQA Trischler et al. [2017] and PubmedQA Jin et al. [2019] to measure OOD generalization. Likewise adversarial SQuad Jia and Liang [2017] and MusiQue Trivedi et al. [2022], both of which are created from SQUAD would help us characterize the dimenisons of robustness and compositionality respectively.

SMD: We would like to explore how well do models generalize to different kinds of social influence in conversations or tasks; these span multiple tasks like persuasion Wang et al. [2019], Li et al. [2020b], negotiation He et al. [2018b], Chawla et al. [2021], Peskov et al. [2020], empathy detection Sharma et al. [2020], morality recognition Ziems et al. [2022], Trager et al. [2022], and argumentation Al-Khatib et al. [2018] amongst others.

We will use the same experimental-setup in terms of model architectures and design paradigms as in the previous section.

9.3 Exploring Compositional Generalization

Despite a multi-faceted view of generalization, we have still so far inspected generalization for a specific dataset as a whole and according to a given metric. This raises two important research questions.

- Are all instances in the test data equally challenging? The existence of challenge sets, and stress-test datasets like Gururangan et al. [2018], Dutt et al. [2023] illustrates that generalization abilities of most models are restricted to specific sub-populations that occur frequently during training. While this necessitates the need for designing harder benchmarks, a more fundamental question is how do we characterize this complexity of test data? How do we formalize this notion of near vs far transfer?
- Are all evaluation metrics equally important or are some metrics more important than others? While such an evaluation paradigm is an important hot bed of research for individual NLP applications, it has not yet been explored systematically in the context of NLP generalization. It raises questions such as how do we combine the results of two models acrosss different datasets or across different metrics for the same dataset or even across different instances of the same dataset?

We hope to explore these two additional research questions in this current proposal.

Bibliography

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7.2.3

Rohini Ahluwalia. Examination of psychological processes underlying resistance to persuasion. *Journal of Consumer Research*, 27(2):217–232, 2000. 2.2

Khalid Al-Khatib, Henning Wachsmuth, Kevin Lang, Jakob Herpel, Matthias Hagen, and Benno Stein. Modeling deliberative argumentation strategies on Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2545–2555, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1237. URL https://www.aclweb.org/anthology/P18-1237. 2.2, 9.2.3

Khalid Al Khatib, Michael Völske, Shahbaz Syed, Nikolay Kolyada, and Benno Stein. Exploiting personal characteristics of debaters for predicting persuasiveness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7067–7072, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.632. URL https://www.aclweb.org/anthology/2020.acl-main.632. 2.2

Alon Albalak, Yi-Lin Tuan, Pegah Jandaghi, Connor Pryor, Luke Yoffe, Deepak Ramachandran, Lise Getoor, Jay Pujara, and William Yang Wang. 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, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.751. 7.3.3

Amal Alqahtani, Efsun Sarioglu Kayi, Sardar Hamidian, Michael Compton, and Mona Diab. 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, 2022. 7.1

Ghulam Ahmed Ansari, Amrita Saha, Vishwajeet Kumar, Mohan Bhambhani, Karthik Sankaranarayanan, and Soumen Chakrabarti. 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, 7 2019. doi: 10.24963/ijcai.2019/679. URL https://doi.org/10.24963/ijcai.2019/679. 2.3

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. Semantic representation for dialogue modeling. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.342. URL https://aclanthology.org/2021.acl-long.342. 3.1, 3.2.4

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1279. URL https://aclanthology.org/P19-1279. 2.4

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, 2013. 3.1, 3.2.2

Larry M Bartels. Priming and persuasion in presidential campaigns. *Capturing campaign effects*, 1:78–114, 2006. 2.2

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in NLP. In Jun'ichi Tsujii, James Henderson, and Marius Paşca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/D12-1091. 4.5.3, 7.3.3

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*, 2021. 3.2.2

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 01 2009. ISBN 978-0-596-51649-9. 3.2.3

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013. 4.1, 4.5.1, 4.5.2

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075. 9.2.3

Gordon Briggs and Matthias Scheutz. Modeling blame to avoid positive face threats in natural language generation. In *Proceedings of the INLG and SIGDIAL 2014 Joint Session*, pages 157–161, Philadelphia, Pennsylvania, U.S.A., June 2014. Association for Computational Linguistics. doi:

10.3115/v1/W14-5001. URL https://www.aclweb.org/anthology/W14-5001. 2.1

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 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, 2020. 4.5.2

Penelope Brown, Stephen C Levinson, and Stephen C Levinson. *Politeness: Some universals in language usage*, volume 4. Cambridge university press, 1987. 2.1, 2.7, 8.1

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359, 2008. 7.3.1, 9.1.3

Nitay Calderon, Naveh Porat, Eyal Ben-David, Zorik Gekhman, Nadav Oved, and Roi Reichart. Measuring the robustness of natural language processing models to domain shifts. *arXiv preprint arXiv:2306.00168*, 2023. (document), 9.1.1, 9.1, 9.1.2, 9.1.3

Amparo Elizabeth Cano-Basave and Yulan He. A study of the impact of persuasive argumentation in political debates. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1413, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1166. URL https://www.aclweb.org/anthology/N16-1166. 2.2

Winston Carlile, Nishant Gurrapadi, Zixuan Ke, and Vincent Ng. Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1058. URL https://www.aclweb.org/anthology/P18-1058. 2.2

Jonathan P. Chang and Cristian Danescu-Niculescu-Mizil. Trouble on the horizon: Forecasting the derailment of online conversations as they develop. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 4743–4754, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1481. URL https://aclanthology.org/D19-1481. 2.1

Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3167–3185, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.254. URL https://

aclanthology.org/2021.naacl-main.254. 9.2.3

Kushal Chawla, Weiyan Shi, Jingwen Zhang, Gale Lucas, Zhou Yu, and Jonathan Gratch. Social influence dialogue systems: A survey of datasets and models for social influence tasks. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 750–766, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.53. URL https://aclanthology.org/2023.eacl-main.53. 8.1

Chih-Yao Chen and Cheng-Te Li. 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, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.272. URL https://aclanthology.org/2021.naacl-main.272. 2.4

Jifan Chen, Yuhao Zhang, Lan Liu, Rui Dong, Xinchi Chen, Patrick Ng, William Yang Wang, and Zhiheng Huang. Improving cross-task generalization of unified table-to-text models with compositional task configurations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5523–5539, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.341. URL https://aclanthology.org/2023.findings-acl.341. 6.1

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 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, 2021. 2.3

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. Learning an executable neural semantic parser. *Computational Linguistics*, 45(1):59–94, March 2019. doi: 10.1162/coli_a_00342. URL https://aclanthology.org/J19-1002. 4.7

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 7.7, 8.2.2

John W Cooley. A classical approach to mediation-part i: Classical rhetoric and the art of persuasion in mediation. *U. Dayton L. Rev.*, 19:83, 1993. 2.2

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1565. URL https://www.aclweb.org/anthology/D19-1565. 2.2

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 250–259, Sofia, Bulgaria, August 2013a. Association for Computational Linguistics. URL https://aclanthology.org/P13-1025. 2.1

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 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, 2013b. 2.7

Devleena Das and Sonia Chernova. Leveraging rationales to improve human task performance. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 510–518, 2020. 2.8

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Syg-YfWCW. 2.3

Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Non-parametric reasoning in knowledge bases. In *Automated Knowledge Base Construction*, 2020. URL https://openreview.net/forum?id=AEY9tRqlU7. 2.3, 4.4.1

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay-Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. Case-based reasoning for natural language queries over knowledge bases, 2021. 2.3, 5.1

Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Manzil Zaheer, Hannaneh Hajishirzi, Robin Jia, and Andrew Mccallum. Knowledge base question answering by case-based reasoning over subgraphs. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4777–4793. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/das22a.html. 5.1, 5.3.1

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423. 9.1.3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https:

//aclanthology.org/N19-1423. 3.2.4, 7.3.3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019c. 6.4.1

Jiwei Ding, Wei Hu, Qixin Xu, and Yuzhong Qu. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1263. URL https://aclanthology.org/D19-1263. 2.3

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1128. URL https://aclanthology.org/P18-1128. 4.5.3, 7.3.3

Ritam Dutt, Rishabh Joshi, and Carolyn Rose. Keeping up appearances: Computational modeling of face acts in persuasion oriented discussions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7473–7485, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.605. URL https://aclanthology.org/2020.emnlp-main.605. 2.7

Ritam Dutt, Sayan Sinha, Rishabh Joshi, Surya Shekhar Chakraborty, Meredith Riggs, Xinru Yan, Haogang Bao, and Carolyn Rose. ResPer: Computationally modelling resisting strategies in persuasive conversations. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 78–90, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.7. URL https://aclanthology.org/2021.eacl-main.7. 7.1, 7.3.1, 9.1.3

Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiah, Dan Roth, and Carolyn Rose. Perkgqa: Question answering over personalized knowledge graphs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 253–268, 2022. 5.1, 2

Ritam Dutt, Sopan Khosla, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 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, 2023. 6.4.3, 9.3

Clara Fannjiang, Stephen Bates, Anastasios N Angelopoulos, Jennifer Listgarten, and Michael I Jordan. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119, 2022. (document),

1.1

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 4.5.2

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. A discriminative graph-based parser for the Abstract Meaning Representation. In Kristina Toutanova and Hua Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1134. URL https://aclanthology.org/P14-1134. 3.2.3

Marieke L Fransen, Edith G Smit, and Peeter WJ Verlegh. Strategies and motives for resistance to persuasion: an integrative framework. *Frontiers in psychology*, 6:1201, 2015a. 2.2

Marieke L Fransen, Peeter WJ Verlegh, Amna Kirmani, and Edith G Smit. A typology of consumer strategies for resisting advertising, and a review of mechanisms for countering them. *International Journal of Advertising*, 34(1):6–16, 2015b. 2.2

Xiyan Fu and Anette Frank. SETI: Systematicity evaluation of textual inference. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4101–4114, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.252. URL https://aclanthology.org/2023.findings-acl.252. 9.2.3

Boris Galitsky, Dmitry Ilvovsky, and Dina Pisarevskaya. Argumentation in text: Discourse structure matters. *CICLing 2018*, 2018. 2.2

James Paul Gee. *An introduction to discourse analysis: Theory and method*. routledge, 2014. 2.6, 7.1

Atticus Geiger, Kyle Richardson, and Christopher Potts. 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, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://www.aclweb.org/anthology/2020.blackboxnlp-1.16. 9.2.3

Erving Goffman. Interaction ritual: essays on face-to-face interaction. 1967. 2.1

Erving Goffman. Front and back regions of everyday life [1959]. *The everyday life reader*, pages 50–57, 2002. 2.6, 7.1, 7.2.1

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016. 4.4.2

Yu Gu and Yu Su. Arcaneqa: Dynamic program induction and contextualized encoding for knowledge base question answering. *arXiv preprint arXiv:2204.08109*, 2022. 5.1, 5.5, 5.6, 6.2

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488, 2021. 2.3, 5.1, 5.2.2, 5.4.1, 5.4.2

Yu Gu, Vardaan Pahuja, Gong Cheng, and Yu Su. Knowledge base question answering: A semantic parsing perspective. *arXiv preprint arXiv:2209.04994*, 2022. 5.2, 5.5

Yu Gu, Xiang Deng, and Yu Su. Don't generate, discriminate: A proposal for grounding language models to real-world environments. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4928–4949, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.270. URL https://aclanthology.org/2023.acl-long.270. 6.2

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.33. URL https://aclanthology.org/2022.emnlp-main.33. 2.7

Sai Gurrapu, Ajay Kulkarni, Lifu Huang, Ismini Lourentzou, and Feras A. Batarseh. Rationalization for explainable nlp: a survey. *Frontiers in Artificial Intelligence*, 6, September 2023. ISSN 2624-8212. doi: 10.3389/frai.2023.1225093. URL http://dx.doi.org/10.3389/frai.2023.1225093. 2.8

Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rose. 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, 2023. 7.4

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 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, 2018. 9.2.3, 9.3

Taher H Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, 15(4):784–796, 2003. 2.3, 4.5.1

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 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, 2018. 7.1, 7.3.1

Devamanyu Hazarika, Soujanya Poria, Roger Zimmermann, and Rada Mihalcea. Conversational transfer learning for emotion recognition. *Information Fusion*, 65:1–12, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.06.005. URL https://www.sciencedirect.com/science/article/pii/S1566253520303018. 2.7, 7.3.1

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *WSDM*, 2021. 4.7

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, 2018a. 2.2

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. doi: 10.18653/v1/D18-1256. URL https://aclanthology.org/D18-1256. 9.2.3

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2020. 9.1.3

Teresa Heath, Robert Cluley, and Lisa O'Malley. Beating, ditching and hiding: consumers' everyday resistance to marketing. *Journal of Marketing Management*, 33(15-16):1281–1303, 2017. 2.2

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 9.1.3

Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. EmotionLines: An emotion corpus of multi-party conversations. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1252. 7.3.1, 9.1.3

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 9.1.3

Xiaofeng Huang, Jixin Zhang, Zisang Xu, Lu Ou, and Jianbin Tong. A knowledge graph based question answering method for medical domain. *PeerJ Computer Science*, 7:e667, 2021. 4.1, 5.1

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174, 2023. (document), 9.2.1, 9.1

Sahil Jayaram and Emily Allaway. Human rationales as attribution priors for explainable stance detection. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5540–5554, Online and Punta Cana, Dominican Republic, November 2021.

Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.450. URL https://aclanthology.org/2021.emnlp-main.450. 2.8

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://aclanthology.org/D17-1215. 9.2.3

Longquan Jiang and Ricardo Usbeck. Knowledge graph question answering datasets and their generalizability: Are they enough for future research? *arXiv preprint arXiv:2205.06573*, 2022. 5.1

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 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, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.aacl-main.82. 3.3, 3.3

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259. 9.2.3

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 4.7

Brihi Joshi, Aaron Chan, Ziyi Liu, Shaoliang Nie, Maziar Sanjabi, Hamed Firooz, and Xiang Ren. ER-test: Evaluating explanation regularization methods for language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3315–3336, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.242. URL https://aclanthology.org/2022.findings-emnlp.242. 2.8

Brihi Joshi, Ziyi Liu, Sahana Ramnath, Aaron Chan, Zhewei Tong, Shaoliang Nie, Qifan Wang, Yejin Choi, and Xiang Ren. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. *arXiv preprint arXiv:2305.07095*, 2023. 2.8

Rishabh Joshi, Vidhisha Balachandran, Shikhar Vashishth, Alan Black, and Yulia Tsvetkov. Dialograph: Incorporating interpretable strategy-graph networks into negotiation dialogues. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=kDnal_bbb-E. 2.2

Haein Jung, Heuiyeen Yeen, Jeehyun Lee, Minju Kim, Namo Bang, and Myoung-Wan Koo. Enhancing task-oriented dialog system with subjective knowledge: A large language model-based data augmentation framework. In Yun-Nung Chen, Paul Crook, Michel Galley, Sarik Ghazarian,

Chulaka Gunasekara, Raghav Gupta, Behnam Hedayatnia, Satwik Kottur, Seungwhan Moon, and Chen Zhang, editors, *Proceedings of The Eleventh Dialog System Technology Challenge*, pages 150–165, Prague, Czech Republic, September 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.dstc-1.18. 7.2.2

Dan Jurafsky, Rajesh Ranganath, and Dan McFarland. 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, 2009. 2.6

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 850–862, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.72. 4.1

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2019. 9.2.3

Efsun Sarioglu Kayi, Mona Diab, Luca Pauselli, Michael Compton, and Glen Coppersmith. Predictive linguistic features of schizophrenia. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 241–250, 2017. 7.1

Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. 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, 2022. 5.6

Shakir Khan, Mohd Fazil, Agbotiname Lucky Imoize, Bayan Ibrahimm Alabduallah, Bader M Albahlal, Saad Abdullah Alajlan, Abrar Almjally, and Tamanna Siddiqui. Transformer architecture-based transfer learning for politeness prediction in conversation. *Sustainability*, 15(14):10828, 2023. 2.7

Sopan Khosla and Rashmi Gangadharaiah. Benchmarking the covariate shift robustness of open-world intent classification approaches. *AACL-IJCNLP 2022*, page 14, 2022. 2.7

Sopan Khosla, Ritam Dutt, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. Exploring the reasons for non-generalizability of kbqa systems. In *The Fourth Workshop on Insights from Negative Results in NLP*, pages 88–93, 2023. 5.4.2, 5.6

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 3.4.1

Paul Kingsbury and Martha Palmer. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2002/pdf/283.pdf. 3.1, 3.3

Tina Klüwer. "i like your shirt"-dialogue acts for enabling social talk in conversational agents. In

International Workshop on Intelligent Virtual Agents, pages 14–27. Springer, 2011. 2.1

Tina Klüwer. Social talk capabilities for dialogue systems. 2015. 2.1

Silvia Knobloch-Westerwick and Jingbo Meng. Looking the other way: Selective exposure to attitude-consistent and counterattitudinal political information. *Communication Research*, 36 (3):426–448, 2009. 2.2

Yunshi Lan and Jing Jiang. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.91. URL https://aclanthology.org/2020.acl-main.91. 2.3, 4.1, 5.3.1

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. 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, 2019. 2.7

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. The language of prompting: What linguistic properties make a prompt successful? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9210–9232, 2023. 7.7

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020. 9.1.3

Mingchen Li and Jonathan Shihao Ji. Semantic structure based query graph prediction for question answering over knowledge graph. *arXiv preprint arXiv:2204.10194*, 2022. 5.1, 5.3.1

Mingyang Li, Louis Hickman, Louis Tay, Lyle Ungar, and Sharath Chandra Guntuku. Studying politeness across cultures using english twitter and mandarin weibo. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–15, 2020a. 2.7

Yu Li, Kun Qian, Weiyan Shi, and Zhou Yu. End-to-end trainable non-collaborative dialog system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8293–8302, 2020b. 9.2.3

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1362. URL https://aclanthology.org/D18-1362. 2.3

Michael K Lindell, Christina J Brandt, and David J Whitney. A revised index of interrater agreement for multi-item ratings of a single target. *Applied Psychological Measurement*, 23(2): 127–135, 1999. 7.2.4

Trond Linjordet and Krisztian Balog. Would you ask it that way? measuring and improving question naturalness for knowledge graph question answering. *arXiv preprint arXiv:2205.12768*, 2022. 5.6

Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. A simple yet effective relation information guided approach for few-shot relation extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 757–763, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.62. URL https://aclanthology.org/2022.findings-acl.62. 2.4

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. Uni-parser: Unified semantic parser for question answering on knowledge base and database. *arXiv preprint arXiv:2211.05165*, 2022b. 5.1

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019a. 4.4.2, 4.5.1, 4.5.2

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b. 9.1.3

Niklas Lüdemann, Ageda Shiba, Nikolaos Thymianis, Nicolas Heist, Christopher Ludwig, and Heiko Paulheim. A knowledge graph for assessing agressive tax planning strategies. In *International Semantic Web Conference*, pages 395–410. Springer, 2020. 4.1, 5.1

Haoran Luo, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, et al. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *arXiv preprint arXiv:2310.08975*, 2023. 6.1, 6.4.4

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. Language models of code are few-shot commonsense learners. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1384–1403, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.90. URL https://aclanthology.org/2022.emnlp-main.90. 7.2.2

Gaurav Maheshwari, Priyansh Trivedi, Denis Lukovnikov, Nilesh Chakraborty, Asja Fischer, and Jens Lehmann. Learning to rank query graphs for complex question answering over knowledge graphs. In *International semantic web conference*, pages 487–504. Springer, 2019. 2.3

Franc Mairesse, Marilyn Walker, et al. Words mark the nerds: Computational models of personality recognition through language. In *Proceedings of the annual meeting of the cognitive science society*, volume 28, 2006. 7.1

Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian Mcauley. Knowledge-grounded self-rationalization via extractive and natural language explanations. In *International Conference on Machine Learning*, pages 14786–14801. PMLR, 2022. 2.8

James R Martin and Peter R White. *The language of evaluation*, volume 2. Springer, 2003. 2.6, 7.1

James Robert Martin and David Rose. *Working with discourse: Meaning beyond the clause*. Bloomsbury Publishing, 2003. 7.1

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 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), 2020. 9.2.3

Shikib Mehri. *Towards Generalization in Dialog through Inductive Biases*. PhD thesis, Language Technologies Institute, Carnegie Mellon University, 2022. 2.7

Miriam Meyerhoff. In pursuit of social meaning. *Journal of Sociolinguistics*, 23(3):303–315, 2019. 2.6

Mayank Mishra, Prince Kumar, Riyaz Bhat, Rudra Murthy V, Danish Contractor, and Srikanth Tamilselvam. Prompting with pseudo-code instructions. *arXiv preprint arXiv:2305.11790*, 2023. 7.2.2

Salman Mohammed, Peng Shi, and Jimmy Lin. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2047. URL https://aclanthology.org/N18-2047. 2.3

José David Moreno, Jose A Martinez-Huertas, Ricardo Olmos, Guillermo Jorge-Botana, and Juan Botella. Can personality traits be measured analyzing written language? a meta-analytic study on computational methods. *Personality and Individual Differences*, 177:110818, 2021. 7.1

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012. 1.1

Elena Musi, Debanjan Ghosh, and Smaranda Muresan. Changemyview through concessions: Do concessions increase persuasion? *Dialogue & Discourse*, 9(1):107–127, 2018. 2.1

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 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, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4007. URL https://aclanthology.org/W19-4007. 3.3

Nona Naderi and Graeme Hirst. Using context to identify the language of face-saving. In *Proceedings of the 5th Workshop on Argument Mining*, pages 111–120, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5214. URL https://www.aclweb.org/anthology/W18-5214. 2.1

Arijit Nag, Bidisha Samanta, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 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, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.45. URL https://aclanthology.org/2021.conll-1.45. 2.4

Sumit Neelam, Udit Sharma, Hima Karanam, Shajith Ikbal, Pavan Kapanipathi, Ibrahim Abdelaziz, Nandana Mihindukulasooriya, Young-Suk Lee, Santosh Srivastava, Cezar Pendus, et al. A benchmark for generalizable and interpretable temporal question answering over knowledge bases. *arXiv preprint arXiv:2201.05793*, 2022. 5.1

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and robust models for biomedical natural language processing. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5034. URL https://aclanthology.org/W19-5034. 3.2.1

Dong Nguyen, A Seza Doğruöz, Carolyn P Rosé, and Franciska De Jong. Computational sociolinguistics: A survey. *Computational linguistics*, 42(3):537–593, 2016. 7.1

Dong Nguyen, Laura Rosseel, and Jack Grieve. 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, 2021. 2.6

Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. Knowledge graph-based question answering with electronic health records. *arXiv preprint arXiv:2010.09394*, 2020. 4.1, 5.1

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, 2020a. 2.7

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 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, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.298. URL https://aclanthology.org/2020.emnlp-main.298. 2.4

Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. 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, 2017. 4.4.2, 4.5.2

Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. It takes two to lie: One to lie, and one to listen. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3811–3854, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.353. URL

https://aclanthology.org/2020.acl-main.353. 9.2.3

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1050. URL https://aclanthology.org/P19-1050. 9.1.3

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. Aligning English strings with Abstract Meaning Representation graphs. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1048. URL https://aclanthology.org/D14-1048. 3.2.3

Jakob Prange, Nathan Schneider, and Lingpeng Kong. Linguistic frameworks go toe-to-toe at neuro-symbolic language modeling. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *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, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.325. URL https://aclanthology.org/2022.naacl-main.325. 3.1

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 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*, 2020. URL https://nlp.stanford.edu/pubs/qi2020stanza.pdf. 3.2.2

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. Few-shot relation extraction via Bayesian meta-learning on relation graphs. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7867–7876. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/qu20a.html. 2.4

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 7.3.3, 9.1.3

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 1.2.1, 6.4.1, 7.3.3, 9.1.3

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016. 9.1.3

Kavel Rao, Liwei Jiang, Valentina Pyatkin, Yuling Gu, Niket Tandon, Nouha Dziri, Faeze Brahman, and Yejin Choi. What makes it ok to set a fire? iterative self-distillation of contexts and rationales for disambiguating defeasible social and moral situations. In Houda Bouamor,

Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12140–12159, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.812. URL https://aclanthology.org/2023.findings-emnlp.812. 2.8, 7.1, 7.6, 7.7, 8.2.2

Hongyu Ren, Hanjun Dai, Bo Dai, Xinyun Chen, Michihiro Yasunaga, Haitian Sun, Dale Schuurmans, Jure Leskovec, and Denny Zhou. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In *International Conference on Machine Learning*, pages 8959–8970. PMLR, 2021. 2.3, 4.1

Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1187. URL https://aclanthology.org/D18-1187. 9.2.3

Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3702–3710, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.302. URL https://aclanthology.org/2020.emnlp-main.302. 3.2.1

Benedek Rozemberczki and Rik Sarkar. 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, 2020. 4.5.2

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. Do syntax trees help pre-trained transformers extract information? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.228. URL https://aclanthology.org/2021.eacl-main.228. 2.5

Amrita Saha, Ghulam Ahmed Ansari, Abhishek Laddha, Karthik Sankaranarayanan, and Soumen Chakrabarti. Complex program induction for querying knowledge bases in the absence of gold programs. *Transactions of the Association for Computational Linguistics*, 7:185–200, March 2019. doi: 10.1162/tacl_a_00262. URL https://aclanthology.org/Q19-1012. 2.3

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. ConjNLI: Natural language inference over conjunctive sentences. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.661. URL https://aclanthology.org/2020.emnlp-main.661. 9.2.3

Victor Tejedor San José. The role of humor and threat on predicting resistance and persuasion. 2019. 2.2

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. Social bias frames: Reasoning about social and power implications of language. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.486. URL https://aclanthology.org/2020.acl-main.486. 2.8

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large LMs. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.248. URL https://aclanthology.org/2022.emnlp-main.248. 2.8

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.412. URL https://aclanthology.org/2020.acl-main.412. 2.3, 4.5.1, 4.5.2

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.201. URL https://aclanthology.org/2022.acl-long.201. 6.4.1

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018. 3.2.4, 4.4.2, 4.5.2

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*, 2023. 7.7

William Shakespeare. As you like it. Act 2, Scene 7, 1623. All the world's a stage, and all the men and women merely players. 7.1

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. A computational approach to understanding empathy expressed in text-based mental health support. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5263–5276, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.425. URL https://aclanthology.org/2020.emnlp-main.425. 9.2.3

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje F Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. Tiara: Multi-grained retrieval for robust question answering over large knowledge bases. *arXiv preprint arXiv:2210.12925*, 2022. 5.1, 6.2

Georgios Sidiropoulos, Nikos Voskarides, and Evangelos Kanoulas. Knowledge graph simple

question answering for unseen domains. In *Automated Knowledge Base Construction*, 2020. 2.3

Paul Surgi Speck and Michael T Elliott. Predictors of advertising avoidance in print and broadcast media. *Journal of Advertising*, 26(3):61–76, 1997. 2.2

Saurabh Srivastava, Mayur Patidar, Sudip Chowdhury, Puneet Agarwal, Indrajit Bhattacharya, and Gautam Shroff. 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, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.300. 4.4.1, 4.4.2

Amos Storkey. When training and test sets are different: characterizing learning transfer. 2008. 1.1

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 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, 2016. 5.4.1, 5.4.2

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1455. URL https://aclanthology.org/D18-1455. 2.3

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1242. URL https://aclanthology.org/D19-1242. 2.3, 4.4.1

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL https://aclanthology.org/N18-1059. 5.1, 5.4.2

Zakary L Tormala. A new framework for resistance to persuasion: The resistance appraisals hypothesis. *Attitudes and attitude change*, pages 213–234, 2008. 2.2

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6.4.1, 7.2.4, 7.3.3, 8.2.2, 9.1.3

Jackson Trager, Alireza S Ziabari, Aida Mostafazadeh Davani, Preni Golazizian, Farzan Karimi-

Malekabadi, Ali Omrani, Zhihe Li, Brendan Kennedy, Nils Karl Reimer, Melissa Reyes, et al. The moral foundations reddit corpus. *arXiv preprint arXiv:2208.05545*, 2022. 9.2.3

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, 2017. 9.2.3

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. 9.2.3

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016. 4.5.1

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. DeepStruct: Pretraining of language models for structure prediction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.67. URL https://aclanthology.org/2022.findings-acl.67. 2.4

William Yang Wang, Samantha Finkelstein, Amy Ogan, Alan W Black, and Justine Cassell. Love ya, jerkface: using sparse log-linear models to build positive (and impolite) relationships with teens. In *Proceedings of the 13th annual meeting of the special interest group on discourse and dialogue*, pages 20–29. Association for Computational Linguistics, 2012. 2.1

Xu Wang, Shuai Zhao, Bo Cheng, Jiale Han, Yingting Li, Hao Yang, and Guoshun Nan. 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, 2020a. 2.3, 4.4.2, 4.5.1, 4.5.2

Xu Wang, Shuai Zhao, Jiale Han, Bo Cheng, Hao Yang, Jianchang Ao, and Zhenzi Li. 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), December 2020b. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.231. URL https://aclanthology.org/2020.coling-main.231. 2.3, 4.4.1, 4.4.2, 4.5.1, 4.5.2, 4.6

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1566. URL https://www.aclweb.org/anthology/P19-1566. 2.1, 2.2, 9.2.3

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040. 5.6

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and

Samuel R Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. 5.6

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. Emergent abilities of large language models, 2022a. 7.2.3

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022b. 2.8

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022c. 7.7

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. Measuring association between labels and free-text rationales. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.804. URL https://aclanthology.org/2021.emnlp-main.804. 2.8, 7.1

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101. 9.1.3, 9.2.3

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. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.39. URL https://aclanthology.org/2022.emnlp-main.39. 1.2.1, 6.1, 1

Weimin Xiong, Yifan Song, Peiyi Wang, and Sujian Li. Rationale-enhanced language models are better continual relation learners. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15489–15497, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.emnlp-main.958. 2.8

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1417. URL https://aclanthology.org/P19-1417. 2.3

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 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, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.370. URL https://aclanthology.org/2022.naacl-main.370. 2.5

Yoko Yamakata, Shinsuke Mori, and John Carroll. English recipe flow graph corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5187–5194, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.638. 3.3

Diyi Yang, Jiaao Chen, Zichao Yang, Dan Jurafsky, and Eduard Hovy. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1364. URL https://www.aclweb.org/anthology/N19-1364. 2.1, 2.2

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 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, 2020. 2.4

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 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, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.45. URL https://aclanthology.org/2021.naacl-main.45. 2.3

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. *arXiv preprint arXiv:2109.08678*, 2021. 5.1, 5.5, 6.2

Zhi-Xiu Ye and Zhen-Hua Ling. 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, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1288. URL https://aclanthology.

org/N19-1288. 2.4

Zhi-Xiu Ye and Zhen-Hua Ling. 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, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1277. URL https://aclanthology.org/P19-1277. 2.4

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 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, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2033. URL https://aclanthology.org/P16-2033. 4.1, 4.3.2, 5.1, 5.4.2

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. *arXiv preprint arXiv:2210.00063*, 2022a. 5.1

Tianshu Yu, Min Yang, and Xiaoyan Zhao. 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, October 2022b. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.205. 2.4

Omar Zaidan, Jason Eisner, and Christine Piatko. 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, 2007. 2.8

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022. 2.8, 7.1

Jichuan Zeng, Jing Li, Yulan He, Cuiyun Gao, Michael Lyu, and Irwin King. What changed your mind: The roles of dynamic topics and discourse in argumentation process. In *Proceedings of The Web Conference 2020*, WWW '20, page 1502–1513, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380223. URL https://doi.org/10.1145/3366423.3380223. 2.2

Chiyu Zhang and Muhammad Abdul-Mageed. Improving social meaning detection with pragmatic masking and surrogate fine-tuning. In Jeremy Barnes, Orphée De Clercq, Valentin Barriere, Shabnam Tafreshi, Sawsan Alqahtani, João Sedoc, Roman Klinger, and Alexandra Balahur, editors, *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 141–156, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.wassa-1.14. URL https://aclanthology.org/2022.wassa-1.14. 2.6

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua,

Dario Taraborelli, and Nithum Thain. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, 2018. 2.1

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 9.1.3

Zixuan Zhang and Heng Ji. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.4. URL https://aclanthology.org/2021.naacl-main.4. 2.5, 3.1

Zixuan Zhang, Nikolaus Nova Parulian, Heng Ji, Ahmed S Elsayed, Skatje Myers, and Martha Palmer. 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)*, 2021. 2.5

Xinyu Zhao, Shih-Ting Lin, and Greg Durrett. Effective distant supervision for temporal relation extraction. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 195–203, Kyiv, Ukraine, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.adaptnlp-1.20. 2.4

Li Zhenzhen, Yuyang Zhang, Jian-Yun Nie, and Dongsheng Li. Improving few-shot relation classification by prototypical representation learning with definition text. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 454–464, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.34. URL https://aclanthology.org/2022.findings-naacl.34. 2.4

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Prompt consistency for zero-shot task generalization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2613–2626, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.192. URL https://aclanthology.org/2022.findings-emnlp.192. 7.7

Xuhui Zhou, Hao Zhu, Akhila Yerukola, Thomas Davidson, Jena D. Hwang, Swabha Swayamdipta, and Maarten Sap. COBRA frames: Contextual reasoning about effects and harms of offensive statements. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6294–6315, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.392. URL https://aclanthology.org/2023.findings-acl.392. 7.6, 7.7, 8.2.2

Yiheng Zhou, He He, Alan W Black, and Yulia Tsvetkov. A dynamic strategy coach for effective negotiation. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 367–378, Stockholm, Sweden, September 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5943. URL https://www.aclweb.org/anthology/W19-5943. 2.2

Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. 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, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.465. URL https://aclanthology.org/2021.naacl-main.465. 4.1

Shuguang Zhu, X. Cheng, and Sen Su. Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing*, 372:64–72, 2020. 2.3

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W Huang, Jie Fu, Xiang Yue, and Wenhu Chen. Structlm: Towards building generalist models for structured knowledge grounding. *arXiv preprint arXiv:2402.16671*, 2024. 6.1, 1

Caleb Ziems, Jane Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. The moral integrity corpus: A benchmark for ethical dialogue systems. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3755–3773, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.261. URL https://aclanthology.org/2022.acl-long.261. 9.2.3

Julia Zuwerink Jacks and Kimberly A Cameron. Strategies for resisting persuasion. *Basic and applied social psychology*, 25(2):145–161, 2003. 2.2