

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 (Carnegie Mellon University)
David Mortensen (Carnegie Mellon University)
Daniel Fried (Carnegie Mellon University)
Dan Roth (University of Pennsylvania, Amazon AWS)

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

Copyright © 2024 Ritam Dutt

April 2, 2024
DRAFT

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 introduce 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

To family, friends, and well-wishers.

Contents

1	Introduction	1
1.1	Overview of generalization in NLP	1
1.2	Consequences of Seq2Seq Paradigm in LLM era	2
1.3	Scaffolds	4
1.4	Thesis Outline	5
1.5	Timelines	6
2	Literature Review	7
2.1	Formal scaffolds in NLP	7
2.2	Informal scaffolds in NLP	8
2.3	Generalization Research through the lens of NLP	9
I	Formal Scaffolds	11
3	Linguistic representations for fewer-shot relation extraction across domains	12
3.1	Introduction	12
3.2	Related Work	13
3.2.1	Few-shot Relation Extraction	13
3.2.2	Linguistic frameworks for NLP	13
3.3	Methodology	14
3.3.1	Dataset Preprocessing	14
3.3.2	Parsing	15
3.3.3	AMR Alignment	15
3.3.4	Model Architectures	16
3.4	Datasets	17
3.5	Experiments	18
3.5.1	In-Domain Experiments	18
3.5.2	Few-shot Experiments	18
3.6	Results and Discussion	19
4	PERKGQA: Question Answering over Personalized Knowledge Graphs	24
4.1	Introduction	24
4.1.1	Overview	24

4.1.2	Motivation	24
4.1.3	Contributions	25
4.2	Related work	25
4.3	Preliminaries	26
4.3.1	Task Formulation	26
4.3.2	Running Example	27
4.4	Datasets	28
4.4.1	CloudKGQA	28
4.4.2	Modified WebQSP (Mod-WebQSP)	28
4.4.3	Differences between the datasets	28
4.5	Methodology	29
4.5.1	PATHCBR	29
4.5.2	PATHRGCN	31
4.6	Experiments	33
4.6.1	Baselines	33
4.6.2	Experimental Details	34
4.6.3	Evaluation Metrics	35
4.7	Results	35
5	GrailQA++: A Challenging Zero-Shot Benchmark for Knowledge Base Question Answering	41
5.1	Introduction	41
5.2	Preliminaries	42
5.2.1	Task Formulation	42
5.2.2	KBQA Generalization	43
5.3	Isomorphisms in GrailQA	44
5.3.1	Isomorphisms	44
5.3.2	Statistics for GrailQA	44
5.4	GrailQA++	45
5.4.1	Expert Annotated Instances	45
5.4.2	Pre-existing Datasets	47
5.4.3	Statistics of GrailQA++	47
5.5	Experimental Setup	47
5.6	Results	49
6	[Proposed] Enhancing inference-time zero-shot KBQA generalization with LLMs	54
6.1	Introduction	54
6.2	Role of Isomorphisms	55
6.3	Datasets	56
6.4	Approaches	56
6.4.1	Finetune LLMs	56
6.4.2	Train GNNs	57
6.4.3	LLMs for data augmentation	57
6.4.4	LLMs for retrieval	58

II Informal Scaffolds

59

7	Leveraging Machine-Generated Rationales to Facilitate Social Meaning Detection in Conversations	60
7.1	Introduction	60
7.2	Related Work	61
7.2.1	Social Meaning in NLP	61
7.2.2	Generalization in Dialogue	62
7.2.3	Rationales in NLP	62
7.3	Prompting Framework	63
7.3.1	Prompt Design Motivation	63
7.3.2	Structured Prompting	64
7.3.3	Dialogue Context & In-Context Examples	64
7.3.4	Validity of Generated Rationales	64
7.4	Experimental Setup	66
7.4.1	Datasets	66
7.4.2	Settings: In-domain and Transfer	67
7.4.3	Models and Metrics	67
7.5	Results	68
7.6	Qualitative Analysis	71
8	[Proposed Work] Investigating the generalizability of rationales in social conversations	73
8.1	Introduction	73
8.2	Datasets	74
8.2.1	Utterance Classification	74
8.2.2	Dialogue Classification	75
8.3	Experimental Setup for Utterance Classification	75
8.3.1	Instruction-tune Framework Design	75
8.3.2	Models	76
8.3.3	Task Transfer Paradigm	77
8.4	Experimental set-up for Dialogue Classification	77
8.4.1	Prompting framework	77
8.4.2	Models	77
8.4.3	Task Setup	78
8.5	Rationale Generation	79

III Evaluating Scaffolds

80

9	[Proposed Work] Evaluating Generalization through the lens of scaffolds	81
9.1	Introduction	81
9.2	Experimental Setup	82
9.2.1	Datasets	82

9.2.2	Model Frameworks	82
9.2.3	Choice of scaffolds	83
Bibliography		85
Appendices		121
Appendix A		122

List of Figures

1.1	Fraction of papers in *CL conferences that allude to generalization in their abstract or title over time. We observe a steep increase in trend around 2018-19.	1
1.2	Distribution of different types of generalization research in NLP. Hupkes et al. [2023] . It is evident that in the past few years there has been a increased emphasis on generalization across different tasks and compositions, as opposed to merely domains.	2
1.3	A pictorial depiction of the generalization framework in NLP. We use the source data (in blue) to train a system (say a neural network) for a task (say text classification). This trained model is then expected to adapt to an unseen target distribution (as shown in green). The emphasis is that when domains differ, so do the representations that are provided to the model as inputs, and so the model needs to adapt to these changes.	3
1.4	A brief description of how scaffolds can facilitate generalization	4
3.1	Model architecture. Yellow tokens denote BERT special tokens. Dotted lines indicate using BERT embeddings to seed the graph for the R-GCN.	14
3.2	Differences in F1 over baseline from incorporating linguistic graphs in models. .	21
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. <i>Chatbot service</i> , or <i>Analytics</i> (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.	27
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.	30
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.	32

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.	38
4.5	Performance of the different techniques on the CloudKGQA dataset based on the number of hops, head-nodes, logical constraints	39
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).	45
5.2	Confusion matrices for gold Isomorphisms vs predicted Isomorphisms on the GrailQA++ dataset for ArcaneQA (top) and RNG-KBQA (bottom).	50
6.1	Overview of using isomorphisms for improving performance of KBQA systems.	55
6.2	Isomorphism prediction using language models.	57
6.3	Isomorphism prediction using GNNs.	57
6.4	Using data augmentation to generate additional pairs of natural language questions and their corresponding logical forms.	58
6.5	The generate and then retrieve framework for retrieving similar isomorphisms. .	58
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).	61
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).	63
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.	67
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.	69
8.1	Instruction Tune Paradigms for Utterance Classification	75
8.2	Cross task transfer paradigm by including rationales to bridge across different tasks.	76

8.3	Generating rationales per conversation. We will use our prompting framework to generate these rationales.	78
8.4	Proposed design framework to answer whether rationales facilitate understanding of conversational dynamics and whether the understanding can be generalized . .	78
9.1	We use the proposed SIFT architecture of Wu et al. [2021] to incorporate semantic/syntactic dependencies with the baseline model architecture for the task of NLI prediction.	83
9.2	Model Pipeline for generating rationales and infusing them for NLI	84
3	We present here the label distribution for the emotion recognition and the resisting strategies datasets.	128
4	Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets in a zero-shot transfer setting, where models trained for the similar task on a given source domain was then applied to the new target domain (e.g. P4G → CB and CB → P4G for RES and friends → iemocap and iemocap → friends for ERC.)	134
5	Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets for different few-shot examples for all rationales. The solid and dashed lines correspond to the indomain (ID) and transfer (TF) case respectively.	134
6	We present here the confusion matrices of the best performing pair of models and rationales in the in-domain setting for the 4 datasets and the corresponding model in absence of any rationale (UTT) in the in-domain setting (ID)	135
7	We present here the confusion matrices of the best performing pair of models and rationales in the transfer setting at k=20-shot case for the 4 datasets and the corresponding model in absence of any rationale (UTT).	136
8	We present here the stacked bar plots that showcases the relative percentage of times a given label was predicted correctly by the best-performing model when augmented with a particular rationale as opposed to the baseline for different datasets. The labels are arranged in increasing order of frequency, with the number inside each bar indicating the frequency of the label.	137

List of Tables

1.1	Tentative timeline for the thesis completion	6
3.1	Dataset Statistics. The label distribution column visualizes sorted frequencies of labels in each dataset.	17
3.2	Results from in-domain experiments. Each value represents the mean of runs with three random seeds, with standard deviation in parentheses.	19
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.	20
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.	22
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.	29
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.	35
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 PTLN. The best performance is highlighted in bold and the second best is underlined.	36
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.	37

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.	43
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.	48
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.	48
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.	49
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 ≤ 0.05 , 0.01, and 0.001 respectively. . . .	51
5.6	We present the mean (std) on different linguistic dimensions on the zero-shot split of GrailQA development set (Dev), and GrailQA++.	52
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.	53
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	56
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.	65
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.	65
7.3	We present here the statistics of the datasets used and the rationales generated. . .	66
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 -).	68

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.	69
8.1	Utterance Classification Datasets	74
8.2	Dialogue Classification Datasets	74
1	Fraction of times ChatGPT-3.5-turbo-16k was chosen over LLama-2-13B-chat based on the quality of the generated rationales.	122
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.	123
3	We present here the statistics of the datasets used and the rationales generated. . .	123
4	Framework describing the resisting strategies for persuasion (P4G) and negotiation (CB) datasets, as specified in Dutt et al. [2021] . Examples of each strategy are italicised. The examples for each of P4G and CB were borrowed from the original datasets of the same name from Wang et al. [2019] and He et al. [2018] respectively.	124
5	Framework describing the emotion labels in the emotion recognition datasets (IEMOCAP and Friends) Busso et al. [2008] , Poria et al. [2019] . Examples of each label are italicised.	125
6	Below is an example of our prompt for the task of emotion recognition in conversations (ERC).	126
7	Below is an example of our prompt for the task of detecting resisting strategies (RES).	127
8	Hyperparameters used for fine-tuning	128
9	Example of our prompt for the zero-shot and few-shot experiments on LLMs. We illustrate with an example from the P4G dataset.	129
10	Performance of different models on the CB (Craigslist Bargain) dataset for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.	130
11	Performance of different models on the P4G (Persuasion for Good) dataset for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.	131

12	Performance of different models on the Friends dataset for the task of ERC for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.	132
13	Performance of different models on the IEMOCAP dataset for the task of ERC for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.	133
14	Analysis of dialogue utterances with corresponding contextual information and labels when rationales (RAT) are always better.	138
15	Analysis of dialogue utterances with corresponding contextual information and labels when rationales are always worse	139

Chapter 1

Introduction

1.1 Overview of generalization in NLP

The ability to *generalize* well remains a primary desiderata of AI and NLP systems [Hupkes et al. \[2023\]](#). The past few years have seen tremendous growth and popularity of language technologies and accompanied by the massive spike in interest in generalization research [Naik et al. \[2022\]](#). Figure 1.1 illustrates the fraction of papers published in major *CL conferences on a yearly basis, that have keywords corresponding to generalization in the title or abstract. The keywords chosen for this analysis include “generalization”, “generalize”, “generalisation”, “generalise”, “domain adaptation”, “transfer learning”, “out of domain”, and “out-of-domain”. We see a sharp increase around 2018-2019 with the trend having persisted since then.

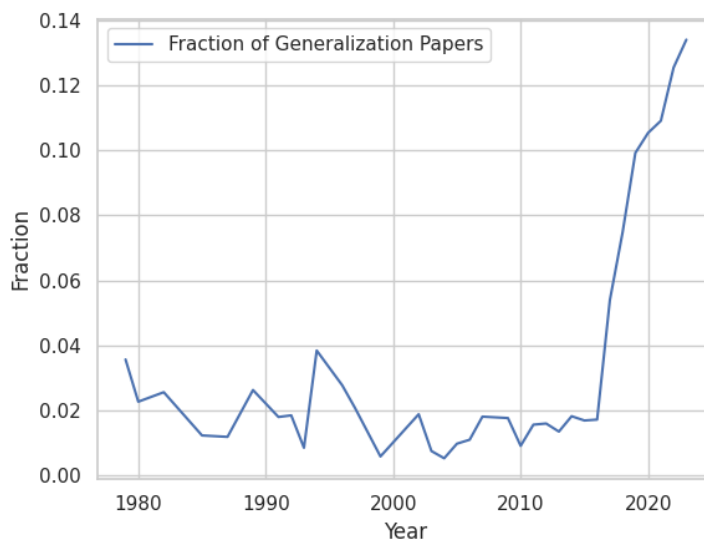


Figure 1.1: Fraction of papers in *CL conferences that allude to generalization in their abstract or title over time. We observe a steep increase in trend around 2018-19.

Traditionally, in the context of machine learning, generalization is defined as the ability of

systems to adapt to shifts in distribution or the data, which casts generalization mostly as a domain adaptation challenge. However, a deeper dive into the generalization trend reveals that in the context of language technologies, the term has broadened to span several scenarios, such as generalizing across languages, adversarial settings, tasks, and compositions beyond just domains. A recent study of [Hupkes et al. \[2023\]](#) provides a broad taxonomic categorization of generalization research in NLP, and as evident from that categorization in Figure 1.2, we observe an increased emphasis on generalization research for different tasks and compositions. Based on the timeline, the broadening of generalization research can be attributed in part to the rise and rise of foundation models spanning pre-trained LMs like BERT, GPT2, and T5 [Devlin et al. \[2019a\]](#), [Radford et al. \[2019\]](#) to recent LLMs like ChatGPT [Achiam et al. \[2023\]](#) and LLama [Touvron et al. \[2023\]](#).

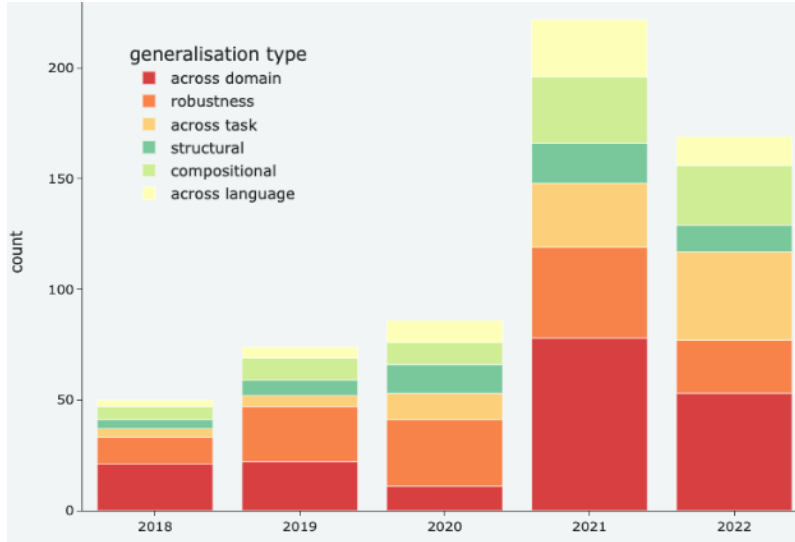


Figure 1.2: Distribution of different types of generalization research in NLP. [Hupkes et al. \[2023\]](#). It is evident that in the past few years there has been an increased emphasis on generalization across different tasks and compositions, as opposed to merely domains.

1.2 Consequences of Seq2Seq Paradigm in LLM era

The inherent power of foundation models led researchers to recast several NLP tasks as a sequence-to-sequence paradigm where both the input and output are treated as text sequences. The popularization of this paradigm has led to unification of several NLP tasks which were subsequently used to train a single model; the model in question not only achieved competitive performance on seen tasks, but also demonstrated impressive instruction following capabilities on unseen tasks [Wang et al. \[2023a, 2022c\]](#), [Chung et al. \[2022\]](#).

Nevertheless, these unified models generally fare worse than smaller stand-alone counterparts which are designed for a specific goal. This is in part, because these models end up ignoring the structure inherent in the data that is crucial for solving the task at hand. For example prior work has observed improved performance on information extraction by incorporating AMRs for both generation [Hsu et al. \[2023\]](#) and extractive tasks [Zhang et al. \[2021a\]](#). Likewise [Chen et al. \[2024\]](#)

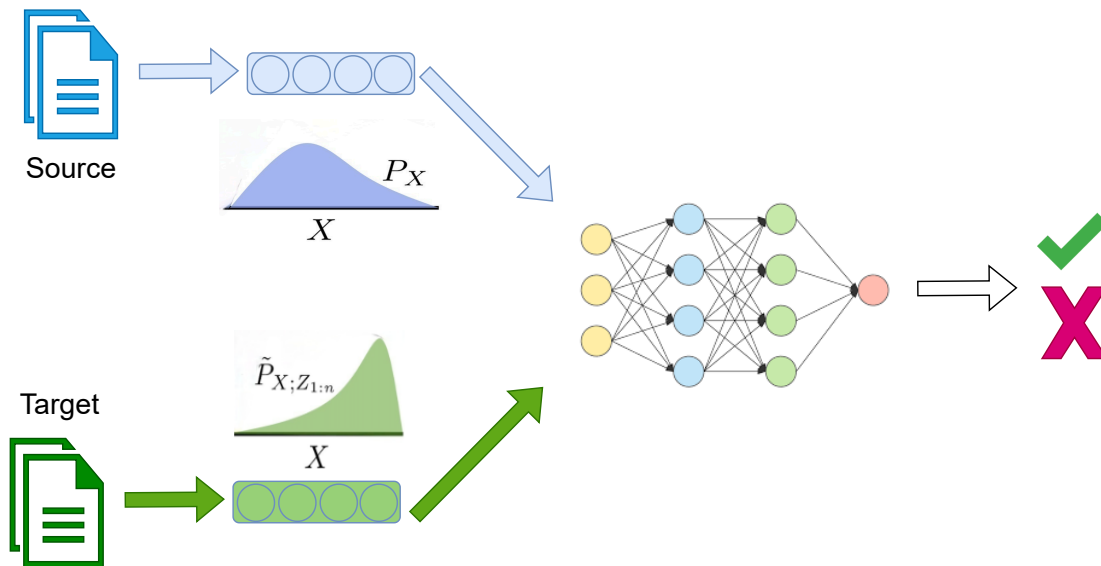


Figure 1.3: A pictorial depiction of the generalization framework in NLP. We use the source data (in blue) to train a system (say a neural network) for a task (say text classification). This trained model is then expected to adapt to an unseen target distribution (as shown in green). The emphasis is that when domains differ, so do the representations that are provided to the model as inputs, and so the model needs to adapt to these changes.

observed the importance of capturing the permutation invariances and other structural properties for efficient reasoning over tabular data.

Additionally, another drawback of these unified seq2seq models is their susceptibility to variation in representations (learned from the text) [Madaan \[2024\]](#) especially for domains and tasks that were unobserved during training. Prior work has highlighted the poor evaluation performance of ChatGPT on tasks that require reasoning over structured knowledge [Zhuang et al. \[2024\]](#). Consequently, when models are evaluated on different domains and tasks, the differences in distribution of the representations are treated as natural representations of domains and as challenges that should be overcome during training. We illustrate this phenomena pictorially how the trained model needs to adapt to changes in distribution of the input representation arising from difference in domains in Figure 1.3.

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

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

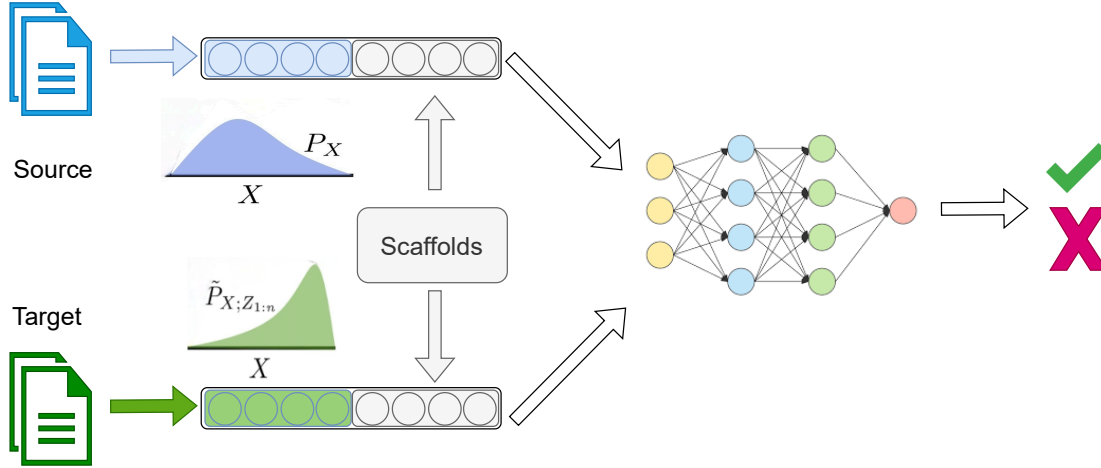


Figure 1.4: A brief description of how scaffolds can facilitate generalization

1.3 Scaffolds

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

In this work, we use “scaffolds” as an umbrella term to refer to constructs that compensates for the deficiencies in the textual representations of the input and helps bridge the differences between the source and the target. Figure 1.4 illustrates this idea as the scaffolds (shown in grey) augment the representations coming from the source and target (shown in blue and green respectively) before they are handed off to the model. Scaffolds are designed to capture a theory (or inductive bias) that remains inarticulated in the text which when supplemented with the model is crucial for solving the task at hand. In our work, we make the distinction between two main categories of scaffolds: formal and informal.

Formal scaffolds refer to architectural structures or representational frameworks that ground or supplement the information obtained from the text in some formal structure. These structures can either be external sources of ontological or encyclopedic knowledge like knowledge graphs, databases, or tables, or formalization of the internal structure of the text in the form of linguistic frameworks like syntactic and semantic parses. Formal scaffolds represent the most prevalent method of inducing structured knowledge into a model, usually in the form of graph [Wu et al. \[2021\]](#), linearized data [Zhuang et al. \[2024\]](#), or other like memory [Khandelwal et al. \[2019\]](#).

Informal scaffolds, on the other hand, augment information to the system in the form of free text. These scaffolds aims to either supplement the information in the current text with contextual or world knowledge or make explicit the subtle (or implicit) information present within the static text without grounding it any formal knowledge source. In a way, these scaffolds verbalize the knowledge encoded in the parameters of an external system or even the same system. For example, rationales, explanations, chain-of-thought reasoning all fall under this broad terminology of informal scaffolds. [Majumder et al. \[2022\]](#), [Wiegrefe et al. \[2021\]](#), [Wei et al. \[2022c\]](#).

The rest of the thesis delves into the different kinds of scaffolds and how they can facilitate generalization. We present an outline of the dissertation in the next section.

1.4 Thesis Outline

The thesis presents an overview of the different kinds of formal and informal scaffolds that can facilitate NLP generalization and the specific cases where and why they help. We also explore evaluation from a generalization research in a more principled manner. In that regard, we outline our thesis in the following chapters.

- Chapter 2 provides a literature review of the utility of different formal and informal scaffolds for several NLP tasks. Specifically, we touch upon how past work has leveraged different scaffolds for facilitating generalization and how we carry the baton forwards in not only investigating the cases or situations where these scaffolds are helpful but also why. Additionally, we explore in-depth the current generalization research landscape in NLP and how our current work addresses a current under-represented area in this space, i.e. a holistic evaluation of generalization research. While, this chapter currently presents an overview of different scaffolds, it also presents an ongoing endeavour to organize and categorize these scaffolds in the form of a meta-analysis study. This survey is still an ongoing work, and is one of the contributions of this dissertation.
- Chapter 3 explores the role of linguistic frameworks in the form of dependency parses and Abstract Meaning Representations (AMRs) as scaffolds for facilitating relation extraction in procedural text. The choice of scaffolds were inspired by the observation that while different corpora for procedural text exhibit stark lexical and topical variations, they are united in the common theme of carrying out a set of step-wise instructions for a particular goal and thus exhibit similar semantics owing to their procedural nature. We thus investigate whether incorporating these linguistic frameworks can facilitate learning the relations between the different mentions across domains, and our experiments highlight their utility especially in a constrained data setting.

Status: This work is published in the proceedings of ACL, 2023. [Gururaja et al. \[2023a\]](#)

- Chapter 4 introduces the task of question answering over knowledge graphs (KG) in a personalized setting. Unlike traditional KGQA, where there exists a single KG that is known apriori and is leveraged similarly across all queries, in PERKGQA a query is accompanied by a KG specific to the user. Consequently KGQA systems need to adapt to unseen queries over new entities for users, without having any access to prior information of the user. We demonstrate in this chapter how one can use the internal structure of the knowledge graph and the path information as scaffolds to facilitate generalization to these new settings over different cases.

Status: The work is published in the findings of NAACL, 2022. [Dutt et al. \[2022a\]](#)

- Chapter 5 extends the notion of generalization to unseen entities in a knowledge graph(or base) to unseen schemas such as classes and relations. We propose the idea of graph isomorphism to characterize the complexity of a question for KBQA and how this characterization serves as a stress test to glean the generalization capabilities of preexisting KBQA systems. We use the concept of isomorphisms to curate a stronger zero-shot generalization benchmark for KBQA and observe a significant drop in performance of systems on our dataset.

Status: The work is published in the proceedings of AACL, 2023 [Dutt et al. \[2023\]](#) and in the workshop proceedings of Insights from Negative Results in NLP [Khosla et al. \[2023a\]](#).

- Chapter 6 outlines the first of the proposed work which leverages the idea of isomorphisms as formal scaffolds to improve the zero-shot generalization performance of KBQA systems during inference. Preliminary work has shown significant improvements in including the isomorphism information to guide answer generation for several off-the-shelf KBQA systems, and this chapter proposes various ways of predicting the isomorphism type for subsequent inference.
- Chapter 7 explores the role of informal scaffolds in the form of rationales that are used to convey the implicit meaning encoded in a social conversation. We propose a prompting framework using LLMs that is grounded in socio-linguistics theory to generate rationales that capture a user’s intentions, assumptions, and biases. We hypothesize that since rationales are able to explicitly verbalize the subtle cues, models trained for social meaning detection tasks are less likely to over-fit on the context-specific linguistic elements. We carry out an extensive study over different models, few-shot settings, and tasks and observe significant gains from incorporating these rationales for both in domain and across domains, thus validating our hypothesis.

Status: The work is under submission at ACL, 2024.

- Chapter 8 is a proposed work that seeks to further explore the role of informal scaffolds as rationales for different social conversation tasks. We put forward two research questions. (i) Can rationales facilitate transfer across tasks and are some transfers more prominent than others? (ii) Are these rationales useful in predicting the outcome of a conversation? This chapter would also explore other additional ways of generating rationales beyond proprietary LLMs.
- Chapter 9 is the final proposed work in our dissertation which proposes evaluating generalization in a more principled manner. We inspect the utility of incorporating different scaffolds over different generalization conditions and scenarios using NLI as a case study. In a nutshell we hope to explore what scaffolds help, where do they help, and why.

1.5 Timelines

We also present a tentative timeline for the each of the proposed works below.

Proposed Work	Timeline
Leveraging isomorphisms for KBQA generalization (Chapter 6)	May-Jul 2024
Leveraging rationales for social conversations (Chapter 8)	Aug-Nov 2024
Evaluating Scaffolds for NLI (Chapter 9)	Dec-Mar 2025
Thesis Defense	Apr-May 2025

Table 1.1: Tentative timeline for the thesis completion

Chapter 2

Literature Review

We situate our dissertation in past work of infusing scaffolds for different NLP tasks, and explore the current space of generalization research in NLP. We review past work on each of these fronts.

2.1 Formal scaffolds in NLP

We define formal scaffolds as architectures or representations that incorporate information from some predefined formal framework. The frameworks in question can either be grounded in external ontological knowledge sources like databases and graphs, or can model the internal structure of the text in the form of semantic or syntactic parsing. The term “formal scaffolds” is closely associated with the broad idea of structure in language technologies, and hence we use these terms interchangeably while referring to past work in this space.

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

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

In a complementary sense, semantic structures in the form of frame-semantics or abstract meaning representations have also been explored for several NLP tasks. Past work has investigated the role of infusing semantic structures in for language modelling [Prange et al. \[2022\]](#), [Vashishth et al. \[2019\]](#), information extraction [Bassignana et al. \[2023\]](#), [Hsu et al. \[2023\]](#), [Zhang and Ji](#)

[2021], and other NLU tasks Wu et al. [2021], Guan et al. [2021], Ma et al. [2023].

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

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

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

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

2.2 Informal scaffolds in NLP

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

Providing textual information to NLP systems grounded in external knowledge stores has seen

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

In a complementary light, owing to the scale at which language models have been pretrained, they end up encoding information about the world in their parameters [Petroni et al. \[2019\]](#), [Zhang et al.](#), [Wang et al. \[2021a\]](#), [Singhal et al. \[2023\]](#). Consequently, as an alternate to retrieval or extraction, prior work investigates how one can generate relevant information which serves as an auxiliary input for downstream tasks. A commonplace example is to generate natural language “rationales” or “explanations” for NLU tasks like entailment or reasoning [Wiegrefe et al. \[2021\]](#), [Kumar and Talukdar \[2020\]](#), [Rajani et al. \[2019\]](#). Another line of research that has sparked wide interest is the ability of LLMs to self-rationalize about the task in a zero-shot or few-shot setting through specific prompt designs such as chain-of-thought prompting [Wei et al. \[2022b\]](#), [Zelikman et al. \[2022\]](#), [Zhou et al. \[2023a\]](#), or using the generated outputs to refine their predictions [Madaan et al. \[2024\]](#), to align themselves to unseen tasks [Wang et al. \[2023b\]](#), or improve the faithfulness and consistency of their outputs [Huang et al. \[2023\]](#), [Wang et al. \[2022b\]](#)

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

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

2.3 Generalization Research through the lens of NLP

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

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

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

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

With the prevalence of massive scale pretraining and development of large scale language models, there has been a gradual shift in paradigm in moving towards a task-agnostic set-up where several tasks are unified into a single sequence2sequence paradigm for modelling convenience. The consequence of the scale also led to the popularization of parameter efficient fine-tuning approaches (PEFT) where instead of full-model training, only a subset of it was modified such as adapters, [Houlsby et al. \[2019a\]](#), [Liu et al. \[2022a\]](#), low-rank adaptation [Hu et al. \[2021a\]](#), [Dettmers et al. \[2024\]](#), and prefix-tuning [Li and Liang \[2021\]](#), [Xu et al. \[2021\]](#). In a similar vein, prompt-tuning approaches or in-context learning where a prompt, both continuous or discrete, was tuned in few-shot or fully supervised setting to adapt to the particular task [Gu et al. \[2022b\]](#), [Hambardzumyan et al. \[2021\]](#), [Vu et al. \[2022\]](#), [Lester et al. \[2021\]](#), [Jin et al. \[2022\]](#). The popularity of such prompting PEFT based methods can be gauged since they include seven of the top 25 cited papers in *CL conferences over the past three years.

There has also been an interest in characterizing the transferability for different NLP tasks in different settings. Recent work in this space investigates how to select better source tasks for multi-task NLP [Kim et al. \[2023\]](#), [Albalak et al. \[2022\]](#), what intermediate tasks to fine-tune on [Poth et al. \[2021\]](#), [Pruksachatkun et al. \[2020\]](#), and whether and how such cross-task generalization ability can be acquired [Ye et al. \[2021a\]](#). There has also been a keen interest in measuring and characterizing domain robustness [Calderon et al. \[2023\]](#) and how the difference in performance across domains can be attributed to the choice of the target rather than the source.

We thus emphasize the rich history of generalization research in NLP, and how we aim to carry it forward in this dissertation. We explore how different scaffolds influence generalization of NLP, especially where and why.

Part I

Formal Scaffolds

Chapter 3

Linguistic representations for fewer-shot relation extraction across domains

3.1 Introduction

In this chapter, we motivate the role of linguistic frameworks in the form of dependency parses and abstract meaning representations (AMRs) as formal scaffolds for the task of few-shot relation extraction in procedural text.

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

It is precisely, why we use linguistic frameworks as scaffolds, because we hypothesize that the underlying semantics 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. Since we operate over three datasets across two domains: recipes for cooking, and materials science synthesis procedures, each defining 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.

Additionally, we hypothesize that linguistic structures offer abstraction over the way natural language varies over different domains, providing representations that might express meaning in domain-general ways. We therefore investigate whether including linguistic representations encourages learning domain-agnostic representations of relations such that models can generalize better in a few-shot setting, i.e. learning from less high-quality data in new domains.

We 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 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 Related Work

3.2.1 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. [2022b] and the “long-tail” problem of skewness in relation classes. Ye and Ling [2019b]. Salient work in that direction includes (i) designing RE-specific pretraining objectives for learning better representations Baldini Soares et al. [2019], Zhenzhen et al. [2022], Wang et al. [2022a], (ii) incorporating meta-information such as relation descriptions Yang et al. [2020], Chen and Li [2021] a global relation graph, Qu et al. [2020], or entity types Peng et al. [2020b], and (iii) leveraging additional information in the form of dependency parses Yu et al. [2022c], translated texts for multilingual RE Nag et al. [2021], or distantly supervised instances Zhao et al. [2021], Ye and Ling [2019a]. 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.

3.2.2 Linguistic frameworks for NLP

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

Supplementing training data with explicit linguistic structure, in the form of syntactic and semantic parses has led to substantial improvements in the in-domain performance on several NLP tasks. Sachan et al. [2021] challenges the utility of syntax trees over pre-trained transformers for IE and observed that one can only obtain meaningful gains with gold parses. Semantic parses, in

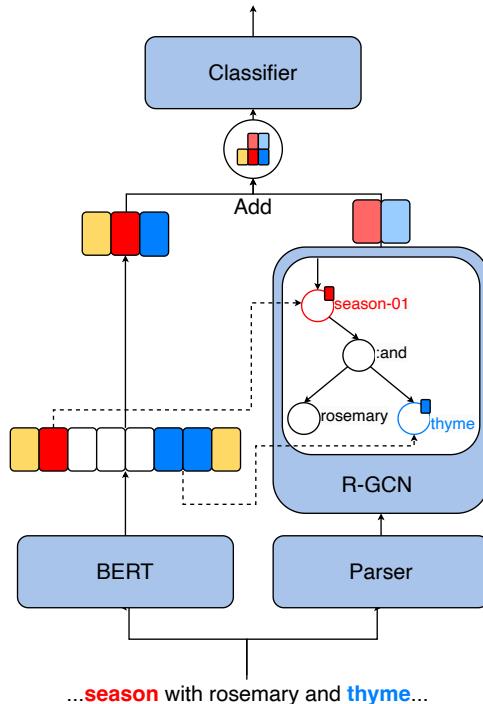


Figure 3.1: Model architecture. Yellow tokens denote BERT special tokens. Dotted lines indicate using BERT embeddings to seed the graph for the R-GCN.

the form of AMRs, have shown to be beneficial for IE [Zhang et al. \[2021b\]](#), [Zhang and Ji \[2021\]](#), [Xu et al. \[2022\]](#), even when the parses employed are not human-annotated. In this work, we raise the question of the utility of either kind of parse for few-shot RE in a cross-domain setting.

3.3 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 3.3.4. 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.

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

3.3.2 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., 2020b], which produces a dependency graph per sentence. We then create a "top" node for the graph to link the individual trees for relations that span sentences.

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

3.3.3 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 alignment. 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

¹<https://github.com/bjascob/amrlib>

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 standard² 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.

3.3.4 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 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

²<https://amr.isi.edu/doc/amr-alignment-guidelines.html>




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.

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.

3.4 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 3.3.1) 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 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. MSCor-

pus 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.

3.5 Experiments

3.5.1 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 Hub³ 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.

3.5.2 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 $K \in \{1, 5, 10, 20, 50, 100\}$. 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<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.

3.6 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.

We find that both dependency parse and AMR representations show a statistically significant positive effect on performance. In particular, we test the significance of the effect with an ANOVA model with multiple independent variables: namely, source and target dataset (EFGC, RISec, MSCorpus), representation case (Baseline, +Dep, +AMR), few-shot setting (1, 5, 10, 20, 50, 100), and transfer setting (in-domain vs out-of-domain). The dependent variable is test set F1. The data table for the analysis includes 3 runs for each combination of variables each with a separate

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

Table 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.

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

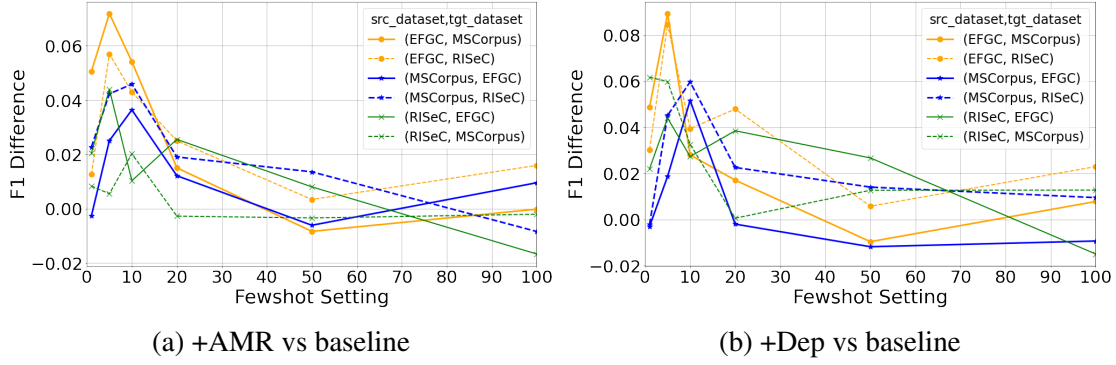


Figure 3.2: Differences in F1 over baseline from incorporating linguistic graphs in models.

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 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 linguistic 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

Target	Source	Fewshot Setting		
		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.

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

Chapter 4

PERKGQA: Question Answering over Personalized Knowledge Graphs

4.1 Introduction

4.1.1 Overview

In this chapter, we propose the task of question answering over knowledge graphs in a personalized setting, which we term PERKGQA. In such a setting, the user has access to their specific knowledge graphs or KG that contains information only relevant to the user. We are only restricted to the user’s KG to answer their queries both during training and inference.

The PERKGQA setting exhibits a wide departure of prior KGQA research which has focused more on generalizable or generic knowledge, and assumes there is a predefined global knowledge graph that exists for all queries. It also assumes that nodes used during inference were already defined in the KG during training and is thus applicable for cases that focus on generalizable knowledge, but not for cases that requires operating over new entities during inference.

In this work, we propose techniques to tackle the nuanced challenges of PERKGQA. We leverage the internal structure of a new or unseen knowledge graph and the path information between the source nodes and answers as scaffolds to prevent reliance on learning node representations of the graph apriori, and enables us to generalize over unseen KGs for new users.

4.1.2 Motivation

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. \[2020\]](#).

However prior studies on KGQA have typically operated over a single knowledge graph (KG). This KG is assumed to be known a priori and is leveraged similarly for all users’ queries during inference. Such an assumption is not applicable to real-world settings, such as healthcare, where

one needs to handle queries of new users over unseen KGs during inference, especially for user’s queries that require situated knowledge such as personal information. The main concerns of leveraging a single global KG for users’ queries introduces the following concerns.

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

PERKGQA appears deceptively 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.

4.1.3 Contributions

We propose two approaches, PATHCBR and PATHRGCN to address the challenges of PERKGQA. 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. Our contributions are as follows.

- 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 Related work

The task of KGQA has evolved from a simple-classification setting [Mohammed et al. \[2018\]](#) to an information retrieval paradigm [Wang et al. \[2020c\]](#), [Saxena et al. \[2020\]](#), [Yasunaga et al. \[2021\]](#), [Sun et al. \[2019\]](#), [Xiong et al. \[2019\]](#) 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. [2021b] 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. [2020c,b]. 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. [2021a] 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.

4.3 Preliminaries

4.3.1 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.

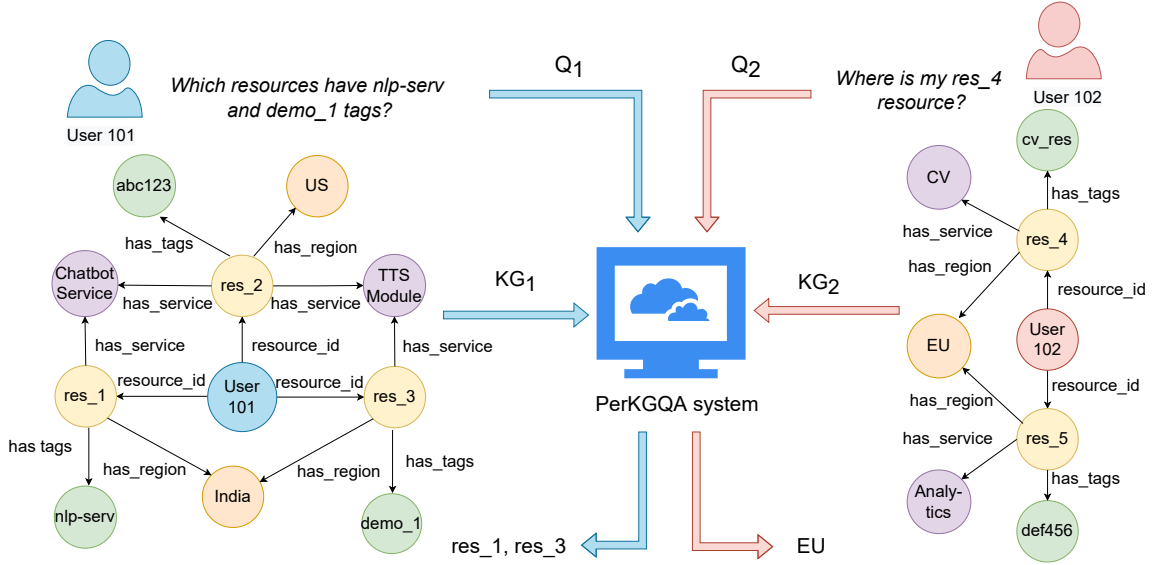


Figure 4.1: PERKGQA for a cloud service provider setting. The two users (in blue and red) create cloud resources (in yellow) in specific regions (in orange), and deploy services e.g. *Chatbot service*, or *Analytics* (in purple) on them. The users assign customized tags (in green) to the resources. Each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

4.3.2 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 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.

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

4.4 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 4.3, 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 assassination?” is the node corresponding to Abraham Lincoln.

4.4.1 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.

4.4.2 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.

4.4.3 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.

4.5 Methodology

4.5.1 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, \dots, 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, \dots, s_k in \mathcal{K}_q Sun et al. [2019], Wang et al. [2020c]. We substitute the extracted text spans with a special [MASK] token, yielding the masked query template q_{MASK} . We hypothesize that masking entities can help us learn the association of the entity with the template and could generalize to unseen entities. We employ a pretrained language model, such as RoBERTa, to create a contextualized embedding of q_{MASK} and call it v_q . We then retrieve the top n questions (q_1, \dots, q_n) and their respective KGs, ($\mathcal{K}_{q_1}, \dots, \mathcal{K}_{q_n}$)

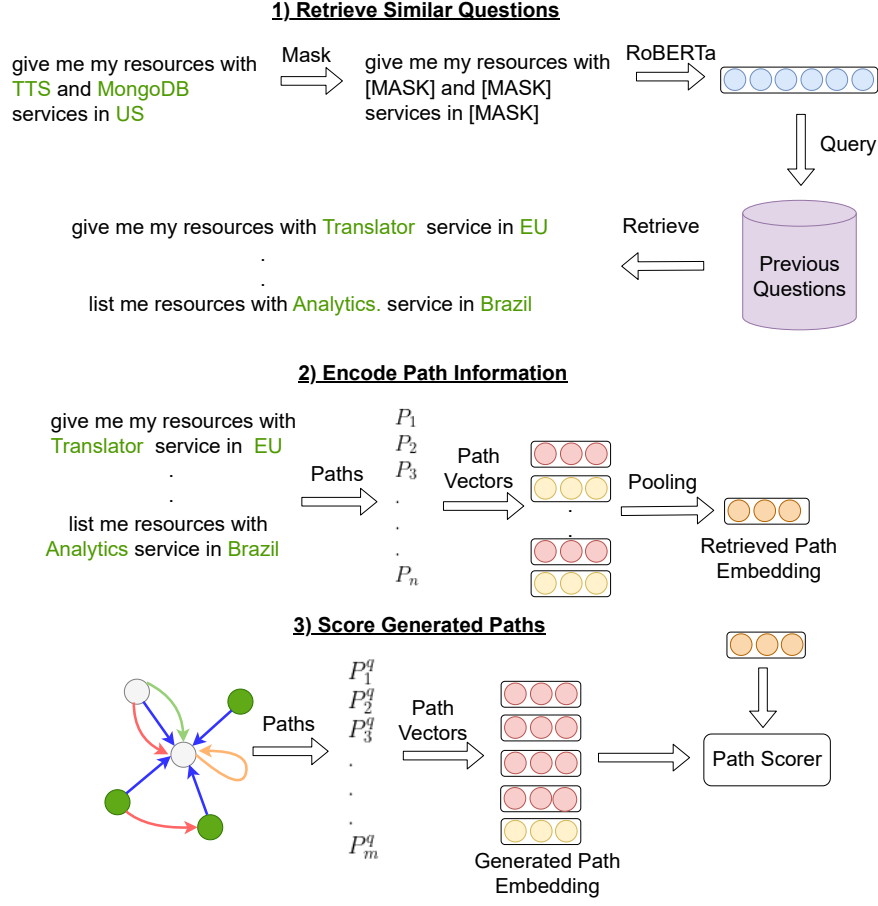


Figure 4.2: PATHCBR Overview: (1) Retrieve questions similar to a given query template from set of questions; (2) Encode path information as a path embedding; (3) Score generated paths using the retrieved path embedding.

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:

$$\begin{aligned}
 q_{\text{MASK}} &\leftarrow \text{MASK}(q) \\
 v_q &\leftarrow \text{ROBERTA}(q_{\text{MASK}}) \\
 (q_1, \mathcal{K}_{q_1}), \dots, (q_n, \mathcal{K}_{q_n}) &\leftarrow \text{RETRIEVE}(v_q)
 \end{aligned}$$

(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 $(r_{i1}, r_{i2}, \dots, r_{im})$ leading from s_{ij} to a_{ik} , or by pooling over its constituent relation embeddings $(v_{r_{i1}}, v_{r_{i2}}, \dots, v_{r_{im}})$. We describe different approaches to obtain the relation embedding v_{r_i} in Section 4.6. 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:

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

(iii) **Scoring generated paths:** For the given query q , we generate all possible paths of a certain length, arising from s_1, s_2, \dots, 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, \dots, 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:

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

4.5.2 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, \dots, 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: $v_{e_1}, v_{e_2}, \dots, v_{e_N}$ for the N nodes e_1, e_2, \dots, e_N in \mathcal{K}_q . Unlike Wang et al. [2020b,c], we do not use pretrained word embeddings since user-provided names can be arbitrary.

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

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

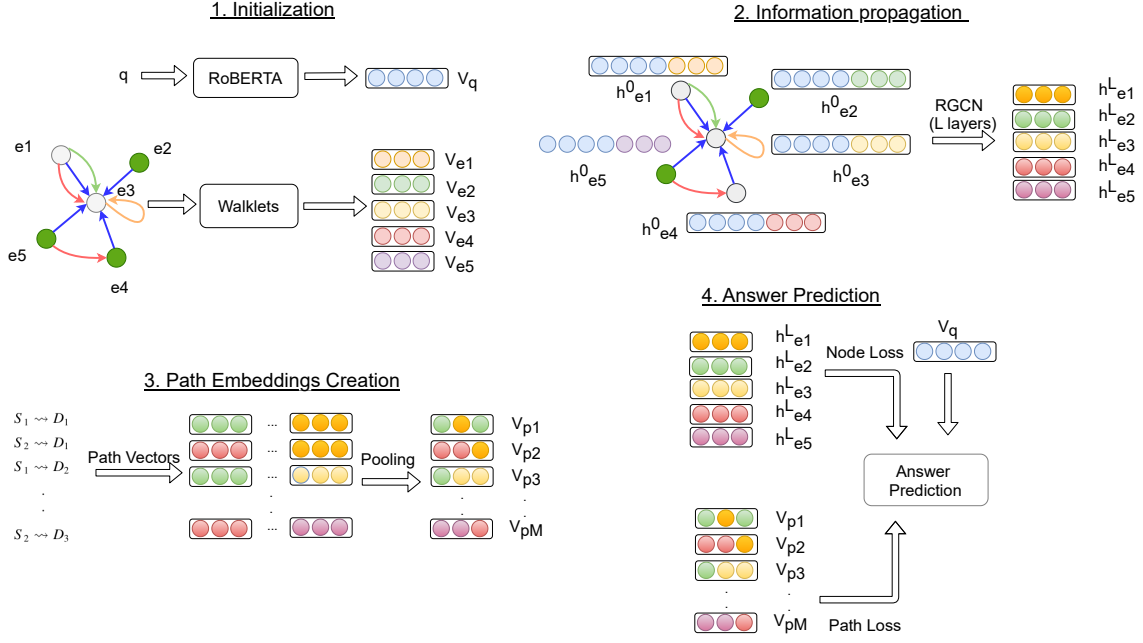


Figure 4.3: PATHRGCN Overview: (1) Initialize the question using a pretrained language model (PTLM) and the nodes in the corresponding KG; (2) Perform information propagation using RGCN to update node embeddings; (3) Encode path information from the source entities (shown in green) to all possible target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

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.

$$h_{e_i}^0 \leftarrow v_q \oplus v_{e_i} \oplus b_i$$

$$h_{e_i}^{j+1} \leftarrow \text{RGCN}(h_{e_i}^j)$$

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

$$v_{p_j} \leftarrow \text{MAX-POOL}(h_{e_i}^L) \forall e_i \in p_j$$

(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 :

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

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, \dots, e_N is an answer and (ii) probability that the paths p_1, p_2, \dots, 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.

4.6 Experiments

4.6.1 Baselines

EmbedKGQA: The EmbedKGQA model [Saxena et al. \[2020\]](#) performs Knowledge Graph Completion (KGC) on an existing knowledge graph, to learn node representations. They use ComplEx [Trouillon et al. \[2016\]](#) to generate node embeddings, to account for the anti-symmetric nature of the relations between nodes. Furthermore, they use RoBERTa [Liu et al. \[2019a\]](#) as the Pre-Trained Language Model (PTLM) to encode the question. They learn an objective function to select answers based on the similarity between question and node embeddings and further perform pruning based on the relation type to prevent over-generation of candidates. EmbedKGQA can perform arbitrary multi-hop reasoning, is not restricted to a specific neighbourhood, and can effectively handle incomplete links/edges. To ensure EmbedKGQA can be applied in our setting, we carried out KGC on the KG associated with the question instead of the entire Freebase KG. This ensures that the entity representations are distinct for each individual KG.

Rel-GCN: The Rel-GCN approach of [Wang et al. \[2020b\]](#) first constructs a smaller sub-graph \mathcal{K}_q for a given question, using PPR [Haveliwala \[2003\]](#) from the large base knowledge-graph, \mathcal{K} . They encode the question q using PTLTLM 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 PERKGQA setting we perform TransE not on the original graph, \mathcal{K} , but on each sub-graph \mathcal{K}_q .

GlobalGraph: The GlobalGraph technique of [Wang et al. \[2020c\]](#) is similar in conception to Rel-GCN, having the same steps, (i) sub-graph construction, (ii) encoding representations of question and nodes, (iii) running RGCN to update the node representations. Moreover, to capture long-dependencies between nodes, the model leverages the set of incoming and outgoing relations to assign a global type for each node. They also identify nodes that are correlated with the question

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

4.6.2 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 NER².² 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. [2020b], and GlobalGraph Wang et al. [2020c], we use RoBERTa-BASE Liu et al. [2019a] to encode the question, and TransE embeddings to initialize the nodes Bordes et al. [2013]. We use the publicly available PyTorch-Geometric library Fey and Lenssen [2019] to implement RGCN Schlichtkrull et al. [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.

²²<https://spacy.io/usage/spacy-101#annotations-ner>

³<https://github.com/malllabiisc/EmbedKGQA>

4.6.3 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. \[2018\]](#).

4.7 Results

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

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

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

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

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

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

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

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

Table 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 PERKGQA task 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?

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

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.

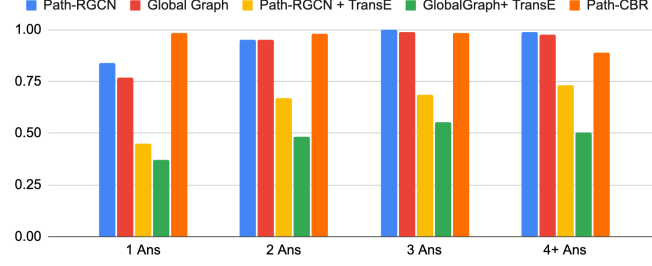
We investigate the benefits of unsupervised graph representation learning techniques to initialize node embeddings. In particular, we compare the efficacy of Walklet and TransE embeddings, when applied to Rel-GCN, GlobalGraph, and PATHRGCN. We see significant improvements for all models when TransE embeddings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

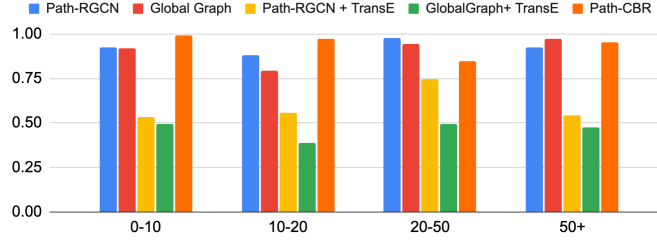
We also investigate the importance of incorporating node loss (NL in Table 4.2) for additional supervision. This aids Mod-WebQSP, where multiple relations between entities give rise to several possible paths between source and answer, most of which are spurious. Since multiple paths do not exist for CloudKGQA, removing the node loss does not deteriorate performance.

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

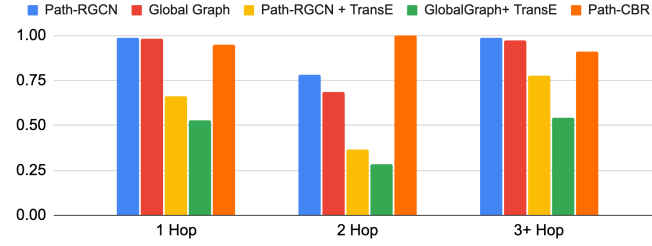
We investigate the performance of the different methods (accuracy) on the PERKGQA task for different properties of the dataset. The methods we investigated were (i) PATHRGCN (ii)



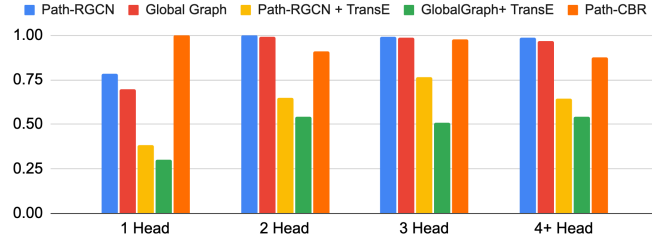
(a) CloudKGQA: Accuracy vs # Answers



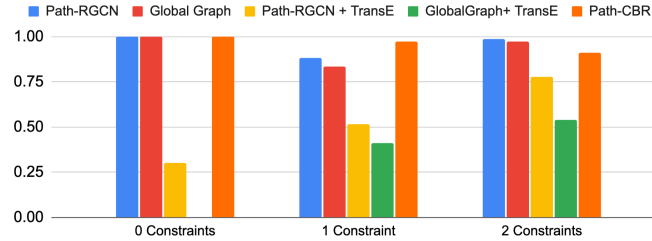
(b) CloudKGQA: Accuracy vs Subgraph size



(c) CloudKGQA: Accuracy vs # Hops

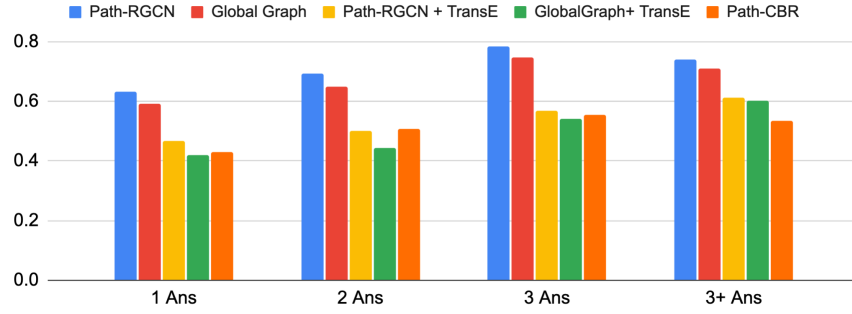


(d) CloudKGQA: Accuracy vs # source entities

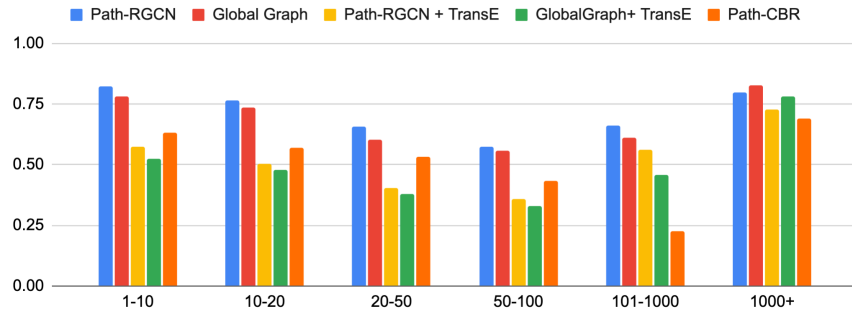


(e) CloudKGQA: Accuracy vs # Constraints

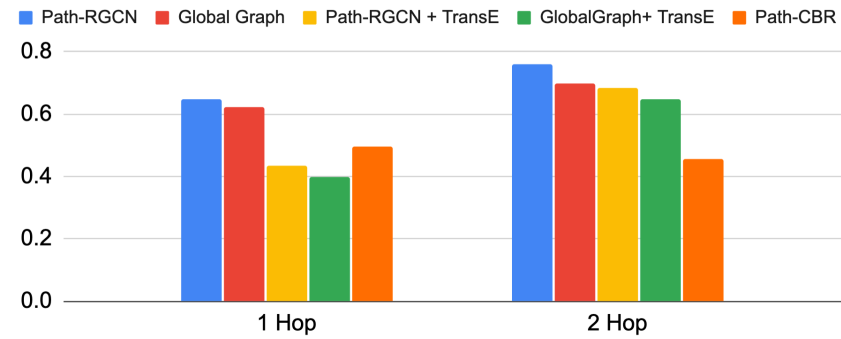
Figure 4.4: Performance of the models on the CloudKGQA dataset across different parameters such as size of the subgraph, number of answers, hops, source entities, and constraints.



(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

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

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

Chapter 5

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

5.1 Introduction

In this chapter, we explore the capabilities of KBQA in generalizing beyond the i.i.d setting, where one might encounter unseen entities or KBs during inference, but still operate over the same schema as discussed in Chapter 4. This challenging “zero-shot generalization” setting requires operating over schema items such as classes and relations that were unobserved during training. The most salient work is that of Gu et al. [2021] where they 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. [2021b], Yu et al. [2022a], Gu and Su [2022], Shu et al. [2022], Liu et al. [2022c] achieving remarkable performance on the leaderboard, specifically on the zero-shot setting.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 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

¹<https://dki-lab.github.io/GrailQA/>

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. \[2022a\]](#).

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 exist 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 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*”),

²Our dataset is available here at <https://github.com/sopankhosla/GrailQA-PlusPlus>.

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.




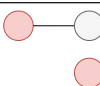
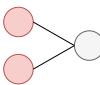

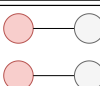
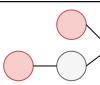
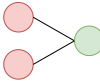
				GrailQA++									
Iso-Code	Pictorial Desc.	GrailQA		EAD		GraphQ		WebQSP		CWQ		Tot	
		Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc
Iso-0		2809	77.9	83	11.9	292	43.9	245	43.2	0	0.0	620	16.1
Iso-1		559	15.5	151	21.7	237	35.6	177	31.2	324	16.8	889	23.0
Iso-2		135	3.8	96	13.8	33	5.0	6	1.1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

Table 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.

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

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 complex isomorphisms. The simple isomorphisms for the remainder of the chapter are Iso-0,1,2. We show the distribution of the isomorphisms in the zero-shot development data of GrailQA in [Table 5.1](#).

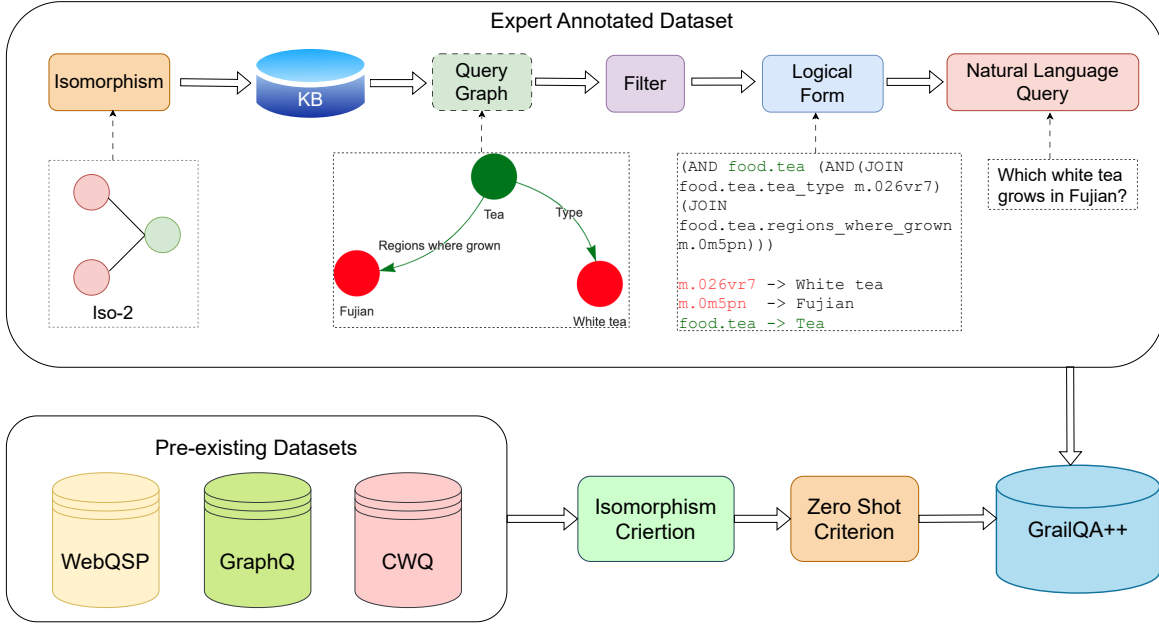


Figure 5.1: Schematic diagram that outlines the GrailQA++ dataset creation. The dataset comprises of question and corresponding logical forms, from two different sources. The former are instances which are hand-annotated by domain experts, and the latter are instances obtained from pre-existing datasets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

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 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

³Example screenshots provided in Appendix ??.

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), conjunctions (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 criteria 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. \[2023b\]](#), 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 criterion: 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 isomorphisms 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.

5.5 Experimental Setup

Baselines: We experiment with two semantic-parsing baselines for KBQA namely RNG-KBQA [Ye et al. \[2021b\]](#) and ArcaneQA [Gu and Su \[2022\]](#). We chose these models because they encapsulate two different strategies of carrying out semantic parsing in the context of KBQA [Gu et al. \[2022a\]](#). Furthermore, they achieve impressive performance on the GrailQA leaderboard and

	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.










Iso-Codes	GrailQA (Dev)		GrailQA++		EAD		GraphQ		WebQSP		CWQ	
	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.

also have publicly available checkpoints which can be used for evaluation. We follow the inference setting mentioned in their Github repositories, with the single exception that for RNG-KBQA we do not restrict ourselves to the subset of Freebase domains for GrailQA.

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

ArcaneQA [Gu and Su \[2022\]](#) employ a seq2seq generative LM to obtain the final logical form from the natural language query. They leverage a constrained decoding paradigm that leverages the information in the KB during query generation to ensure executability.

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

	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.

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 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 isomorphisms 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).

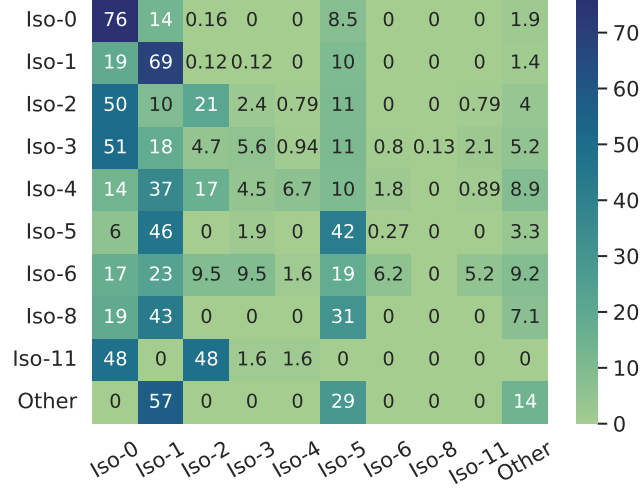
We also investigate the models’ performance on questions with additional functions. These functions are (i) comparatives (ex. greater than, less than), (ii) superlatives (argmax, argmin), (iii) counting or aggregation, and (iv) none (absence of any specific operation). The results in Table 5.4 highlights 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?

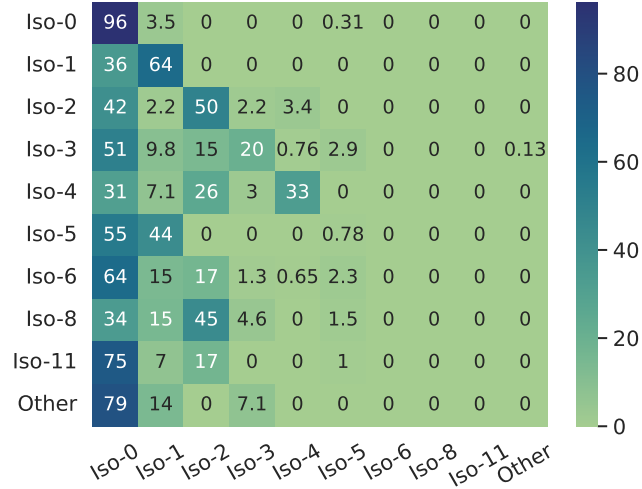
We present a breakdown of the model performance according to the isomorphism type for GrailQA and GrailQA++ in Table 5.3.

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

We hypothesize that different KBQA models are biased towards generating/retrieving logical forms that conform to specific isomorphisms. To delve deeper, we categorize the models mispre-



(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.

dictions 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 Table 5.3, 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. [2023b] on our proposed dataset.

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

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++.

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 CTRL Eval 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 Linjordnet and Balog [2022] where the fluency and naturalness of questions degrades KBQA performance. Moreover, the negative correlation with #Z implies that questions with a greater proportion of unseen classes and relations are harder for models to answer. Furthermore, a positive correlation with #N signifies that being more explicit in framing questions is beneficial for model performance. We see similar trends in #N and #Z across most datasets.

An interesting observation is that our constructed dataset, EAD, is most similar to GraphQuestions both in terms of the EM/F1 scores (Table 5.2) as well as coefficients of different linguistic dimensions (Table 5.5). One hypothesis is that these datasets were created in a similar fashion.

We observe that GrailQA mostly follows a similar trend to All since it accounts for 50% of the entire dataset. However, the readability metric for All is influenced by the pre-existing datasets (CWQ, GraphQ, and WebQSP) which have comparatively higher scores in Table 5.6. All in all, we note that KBQA systems struggle with fluent and natural questions (high coherence and readability scores).

⁴<https://pypi.org/project/py-readability-metrics/>

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 does not lead to a fair comparison with other KBQA systems. 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. \[2022b\]](#)), 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

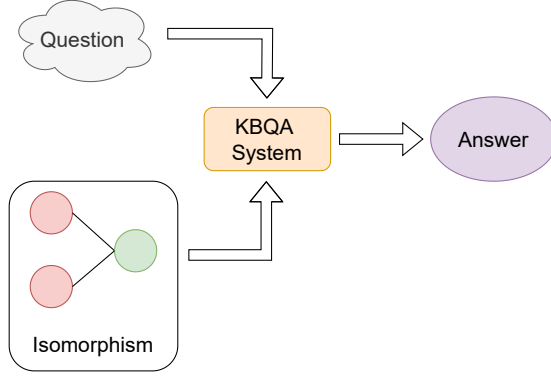


Figure 6.1: Overview of using isomorphisms for improving performance of KBQA systems.

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 isomorphism forms but fails to generalize to more complex hops or constraints.

We acknowledge that the skewed distribution 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. [2021b] 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.

We observe substantial improvements in performance on our challenging GrailQA++ test-set

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

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

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.

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 Approaches

We propose the following techniques to predict the isomorphism category for a given question.

6.4.1 Finetune LLMs

The most 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.

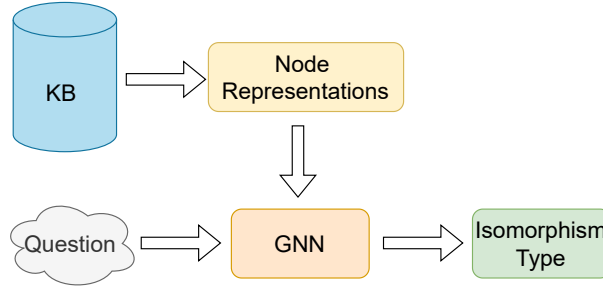


Figure 6.3: Isomorphism prediction using GNNs.

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 downstream 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 [Su et al. \[2016\]](#). 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

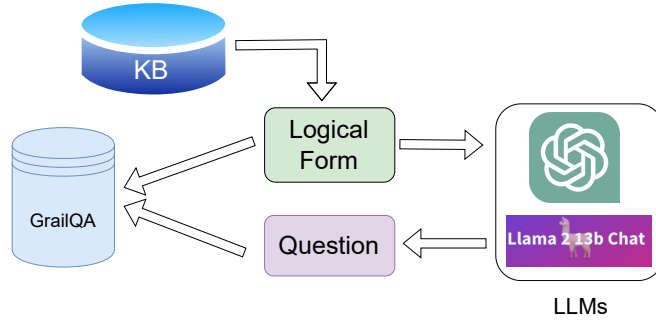


Figure 6.4: Using data augmentation to generate additional pairs of natural language questions and their corresponding logical forms.

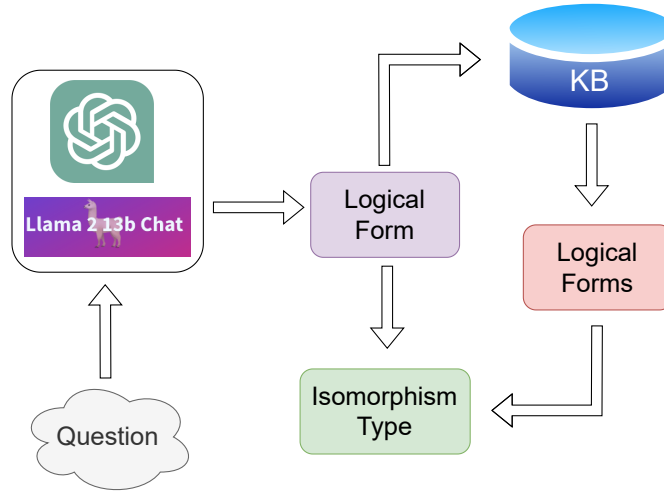


Figure 6.5: The generate and then retrieve framework for retrieving similar isomorphisms.

or queries into natural language, and we aim to exploit this generalization capability to generate additional data [Shu and Yu \[2024\]](#), [Agarwal et al.](#).

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 truth 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.

Part II

Informal Scaffolds

Chapter 7

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

7.1 Introduction

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. \[2016\]](#), such as predicting emotions and mental states of users to identifying what strategies are being employed to solve a particular task like persuasion and negotiation. In this chapter, we propose the idea of informal scaffolds in the form of rationales, that convey the implicit meaning encoded in a social conversation in order to facilitate detection of social meaning.

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 persuasion attempts would manifest in different ways, depending on whether one is negotiating the price of a commodity, or one is hesitating donating to charity [Dutt et al. \[2021\]](#). In this work, we propose a generalizable framework that leverages the rationales generated by Large Language Models (LLMs) and use them as scaffolds for detecting social meaning in conversations.

The idea behind the “rationales” is that they are designed to break through the opaque surface form of the conversation’s text and make the social cues more transparent. While rationales have been utilized previously, to facilitate reasoning [Rao et al. \[2023\]](#), [Zelikman et al. \[2022\]](#), or to explain model predictions [Wiegrefe et al. \[2021\]](#), we use rationales to refer to the elicited social meaning, i.e. why and how an utterance was conveyed in 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



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).

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.

7.2 Related Work

7.2.1 Social Meaning in NLP

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

Social meaning is thus defined as the signaling people do during interactions to maintain positioning in terms of identity and relationship (e.g., practices of signaling are defined in detail in [Gee \[2014\]](#), with additional operationalizations in [Martin and White \[2003\]](#) and [Meyerhoff](#)

[2019]).

While originally defined in the context of socio-linguistics, the term “social meaning” been heavily used in the computational linguistics community. It can refer to different ways or styles people interact [Jurafsky et al. \[2009\]](#), or the social background and identity of a user that can be predicated from linguistic variation [Nguyen et al. \[2021\]](#), 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 myriad 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).

7.2.2 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. \[2013\]](#), [Li et al. \[2020\]](#), [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.

7.2.3 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 [Wiegrefe 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

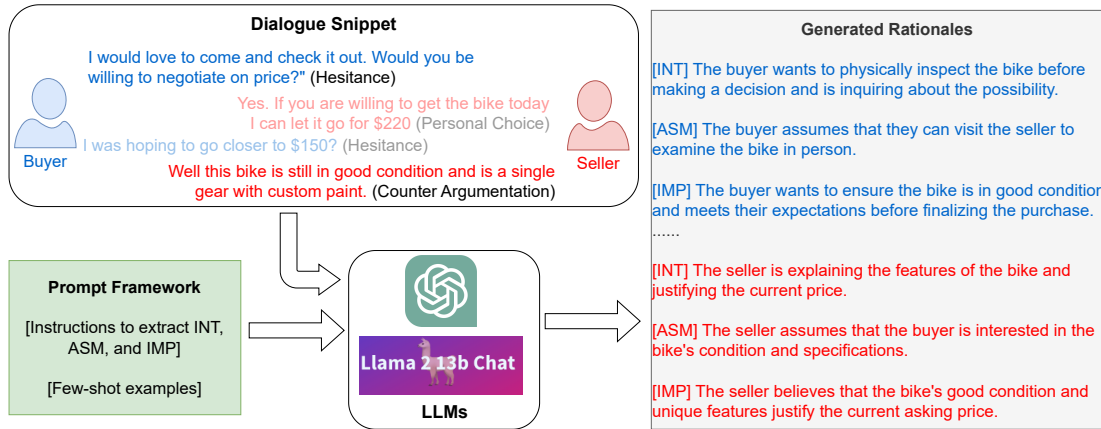


Figure 7.2: We present the prompting framework employed in this work to generate rationales that are subsequently used for dialogue understanding and transfer using pre-existing LLMs such as GPT-3.5-turbo and Llama-2 variants. We feed in the prompt (green box on the left) for a given dialogue to generate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

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.

7.3 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.

7.3.1 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 in-

formation 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.

7.3.2 Structured Prompting

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

We observed that prompting LLM to generate all three rationales (INT, ASM, and IMP) together, facilitated instruction following. Hence we term our approach as “multi-faceted prompting”. These rationales were augmented with the conversational text for two downstream social meaning detection tasks. We provide examples of prompts for the two tasks in Tables 6 and 7 in the Appendix.

7.3.3 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. \[2023\]](#).

7.3.4 Validity of Generated Rationales

To assess the quality of the generated rationales, we prompted two prevalent pre-trained LLMs in contemporary NLP research; GPT-3.5-turbo-16k or ChatGPT¹ 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

¹<https://platform.openai.com/docs/models/gpt-3-5>

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

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

the output with a higher quality, was carried out by three graduate students proficient in English. The results of our experiments present in Table 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 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.

	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.

7.4 Experimental Setup

7.4.1 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 3 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 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 4 and Table 5 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 .

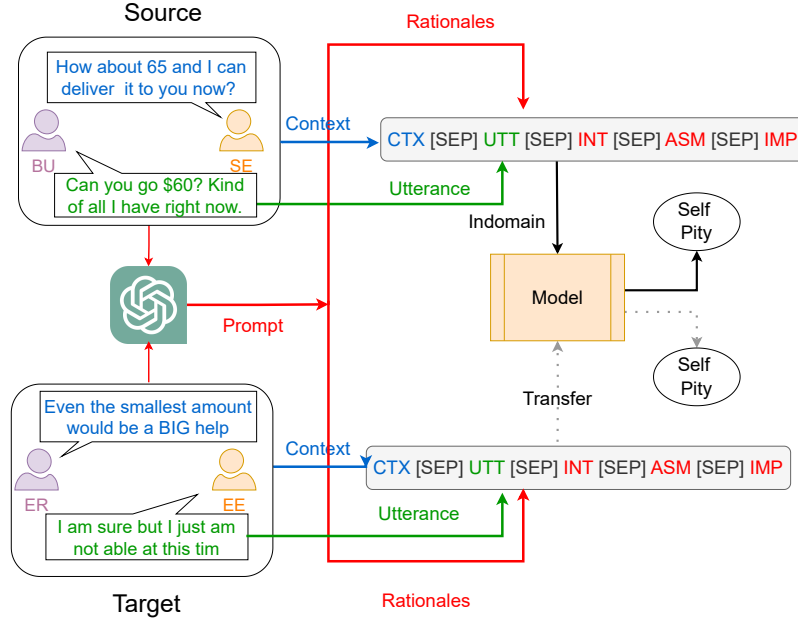


Figure 7.3: Here we illustrate the process of transfer from the source to target. The model is first fine-tuned on the source dialogues, which comprises the current utterance, the previous dialogue context, and the rationales (INT, ASM, and IMP for intentions, assumptions, and implicit information respectively).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

7.4.2 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 $k \in \{5, 10, 20, 50, 100\}$. This enables us to diagnose the impact of adding rationales while controlling for data sparsity.

7.4.3 Models and Metrics

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

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 9 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. \[2018\]](#).

7.5 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 \pm 3.6	60.0 \pm 0.9	70.8 \pm 1.8	50.6 \pm 2.5	35.7 \pm 4.4	48.8 \pm 0.9	40.9 \pm 0.9	26.5 \pm 0.8	39.8 \pm 3.4	40.7 \pm 1.5	35.3 \pm 2.4	42.8 \pm 1.7
INT	68.4\pm1.7	65.6 \pm 2.0*	70.6 \pm 2.8	53.0 \pm 1.6	45.7 \pm 1.6*	51.2 \pm 1.4	45.3 \pm 0.8*	44.5 \pm 1.0*	44.8\pm2.6*	42.6\pm1.3	42.5\pm2.4*	45.0\pm0.7*
ASM	66.6 \pm 0.7	65.3 \pm 1.3*	69.0 \pm 1.8	49.4 \pm 8.1	47.7 \pm 2.4*	51.1 \pm 0.8	44.6 \pm 0.1*	43.4 \pm 1.2*	39.8 \pm 0.6	41.0 \pm 1.8	39.3 \pm 3.2*	43.1 \pm 0.6
IMP	66.9 \pm 0.3	64.9 \pm 1.6*	69.1 \pm 2.6	52.3 \pm 1.7	50.1 \pm 2.6*	51.7 \pm 3.0*	44.7 \pm 1.7*	43.3 \pm 1.9*	44.1 \pm 3.3	42.0 \pm 1.2	39.9 \pm 0.9*	42.0 \pm 0.8
ALL	67.0 \pm 0.7	66.0\pm1.5*	72.2\pm0.5	53.2\pm1.4	50.1\pm1.4*	53.4\pm2.7*	46.2\pm1.3*	45.5\pm0.8*	43.8 \pm 3.1	40.4 \pm 1.0	39.7 \pm 1.8*	44.2 \pm 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

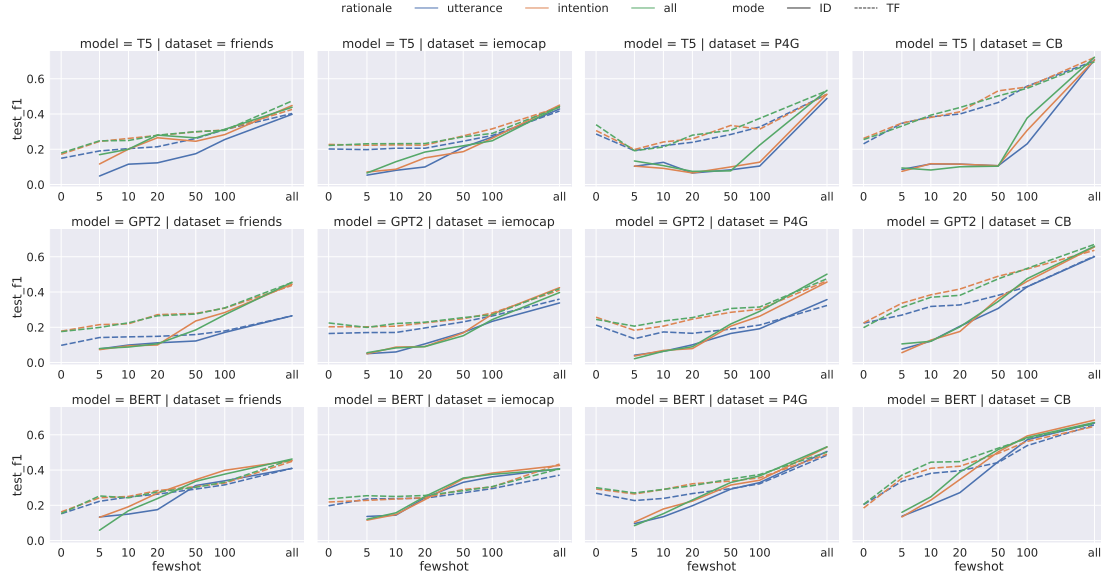


Figure 7.4: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets for different few-shot examples. The solid and dashed lines correspond to the indomain (ID) and transfer (TF) case respectively.

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

dataset combinations. However, the gain diminishes as the model fine-tuned on the entire dataset (denoted by 'all').

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

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

This observation is echoed vividly in 0-shot transfer where we observe a significant gain 83.3% of the times for ERC as opposed to 41.7% for RES. Nevertheless, in a few-shot setting when the model is exposed to instances from the corresponding target domain, the gains start racking up. We emphasize that across all experiments, rationales perform significantly worse than the baseline fewer than 10%. Thus, from a big picture view, rationales can indeed facilitate task performance and transfer.

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

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

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

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

We present our results of using rationales for few-shot prompting in LLMs in Table 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.

7.6 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 14 and cases where their presence consistently degrades performance in Table 15. 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.

Chapter 8

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

8.1 Introduction

In the previous chapter, 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.

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 [Bernsen et al. \[1996\]](#), and the appraisal theory or Martin and White [Martin and White \[2003\]](#). 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.

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.

8.2 Datasets

In this section, we describe the two broad settings where we wish to investigate the role of rationales; the former is an utterance classification task where each utterance is associated with a given label, while the latter is a dialogue classification task where we are expected to classify a given dialogue or conversational snippet.

8.2.1 Utterance Classification

For our utterance classification, we explore six different kinds of social interactions; namely persuasion, negotiation, argumentation, deception, and demonstration of empathy and morality. Each of these six tasks are instantiated with a different dataset to avoid overlap of textual content. We present the tasks, their corresponding datasets, and the labels or strategies corresponding to the task in Table 8.1. At first glance, we observe that no two datasets are equivalent, although they might share some overlap such as expressing emotion is a common strategy in P4G and the Empathy dataset. We describe each dataset in detail later in the Appendix, along with a distribution of labels and their corresponding definitions.

Task	Dataset	Labels
Persuasion	P4G Wang et al. [2019]	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
Negotitation	CaSiNo Chawla et al. [2021]	Self-need, Other-need, Vouch-fair, Promote-coordination, Show-empathy, Small-talk, or non-strategic
Argumentation	WikiAttack Wulczyn et al. [2017]	Comment is a personal attack, aggressive, or toxic.
Deception	Diplomacy Peskov et al. [2020]	Caught, Deceived, Cassandra, Straightforward
Empathy	Therapy Sharma et al. [2020]	Expresses emotion, communicates understanding, and /or explores feeling not explicitly stated
Morality	MFRC Trager et al. [2022]	Non-moral, Care, Fairness, Loyalty, Authority, Purity, Thin Morality

Table 8.1: Utterance Classification Datasets

Task	Dataset	Outcome Prediction
Persuasion	P4G Wang et al. [2019]	Persuader convinces the persuadee to donate.
Negotiation	CB He et al. [2018]	Seller sold at the targeted price.
Argumentation	CGA Zhang et al. [2018]	Conversation does not derail.

Table 8.2: Dialogue Classification Datasets

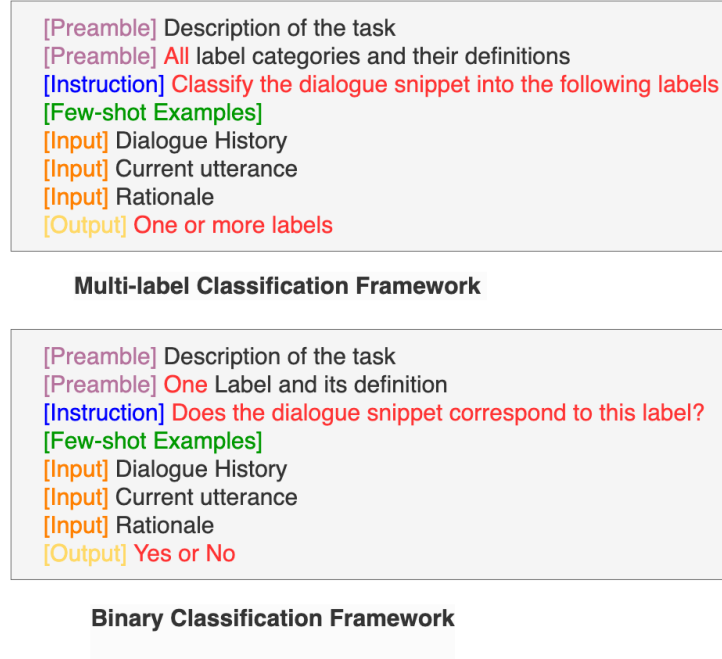


Figure 8.1: Instruction Tune Paradigms for Utterance Classification

8.2.2 Dialogue Classification

For the dialogue classification set-up, we explore three datasets that correspond to three different tasks with different conversational outcomes.

- CGA: We intend to use the Conversations Gone Awry Corpus (CGA) of [Zhang et al. \[2018\]](#) to predict the probability of derailment of a given conversation. The dataset constructed from Wikipedia’s talk page discussions aims to identify whether a discussion between two interlocutors can devolve into a personal attack or go awry.
- P4G: We will also use the Persuasion for Good (P4G) corpus of [Wang et al. \[2019\]](#). We seek to predict whether the persuader was able to convince the persuadee to donate to charity.
- CB: Similar to the aforementioned persuasion task, we also plan to use the Craigslist Bargain (CB) negotiation dataset of [He et al. \[2018\]](#) to predict whether the seller was able to meet the targeted price requirement or not.

The choice behind each of these datasets is that they represent three different task-oriented dialogue datasets, with varying definitions for conversational outcome.

8.3 Experimental Setup for Utterance Classification

8.3.1 Instruction-tune Framework Design

We explore two different ways of formalizing the instruct-tune/prompting paradigm for our task, as shown in Figure 8.1. Since we deal with multi-label classification task, we propose two different

ways to obtain the output from our model architectures.

The first is a multi-label classification paradigm (shown in top), wherein we provide for each instance in a given dataset and task, all the labels or categories that are possible for the tasks alongside their corresponding definitions. We provide few-shot examples to the framework in the same manner, and the framework has to output one or more correct label /categories. In the binary classification framework (shown below), the idea is to identify whether a given dialogue utterance correspond to a particular label or category.

While the binary-classification approach does not require one to provide all the label categories and does decreases the size of the input prompt, it has an unintended consequence of providing more negative examples to instruct-tune the model. Imagine, for a multi-class classification problem amongst 8 possible classes, one needs to provide 8 different instances to the model in the second case as opposed to one in the former. We intend to do a cost-benefit analysis of each approach via performance accuracy and speed.

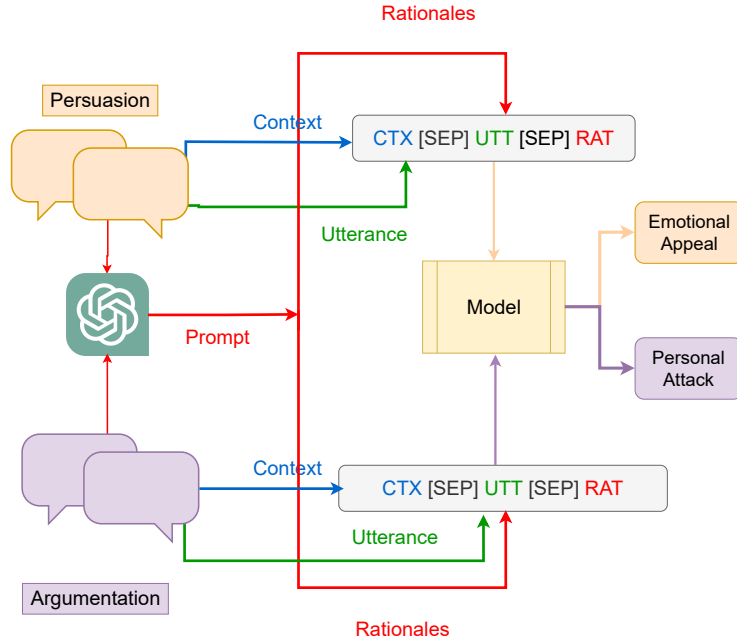


Figure 8.2: Cross task transfer paradigm by including rationales to bridge across different tasks.

8.3.2 Models

Since we care about generalizability across different tasks and domains, with each task spanning different labels and categories, it would be unwise to use the same models as in our past approach (i.e. using BERT, T5, GPT2, etc.), where the domains were different but the label space remained the same. We explore different instruction tune and few-shot prompting models that could facilitate generalization.

- Flan-T5: The instruct-tuned version of T5 model or Flan-T5 [Chung et al. \[2022\]](#) serves as an obvious starting point and has been instruct tuned to show impressive performance for a wide variety of tasks.

- InstructDial: The InstructDial model of [Gupta et al. \[2022\]](#) is well suited for our setting, having been instruct tuned for a wide variety of dialogue tasks. This would enable us to evaluate the gains obtained from pretraining or instruct-tuning on a wide variety of dialogue tasks in both a few-shot as well as a fully-supervised trained setting.
- LLMs: We also want to explore the capability of different LLMs like Llama2 and Mixtral which have excellent instruction following capabilities. We intend to explore both few-shot training and fine-tuning approaches for these models.

8.3.3 Task Transfer Paradigm

We propose two main research hypothesis for utterance classification.

- How well does social influence transfer across different tasks or settings?
- What is the role of the rationales in this transfer or generalization across tasks?

To answer these, we will modify the experimental setup that we had proposed in our previous chapter for cross-domain dialogue classification, as shown in Figure 8.2.

Here, we will first instruct-tune a given model on a given source task, say persuasion which corresponds to the P4G dataset. We will pass to the model a given dialogue utterance, its context and corresponding rationale, obtained by prompting an LLM, here ChatGPT. Having obtained an instruct-tune version of the model, we will use it to observe the performance on a new target dataset (here Argumentation) in a few-shot or zero-shot setting.

To answer Q1 in our hypothesis, we will simply carry out the experiments without the inclusion of the rationales.

To answer Q2, we will investigate all possible source-target pairs and observe whether gains are symmetric, are certain datasets more powerful than others in providing consistent gains, and whether different models show similar outcomes.

8.4 Experimental set-up for Dialogue Classification

In a similar vein, we will investigate the ability of rationales in predicting the success of conversations, which can be varied based on whether there was a successful donation [Wang et al. \[2019\]](#) or whether the conversation remained civil [Zhang et al. \[2018\]](#).

8.4.1 Prompting framework

We will use prompting framework to generate rationales for a given conversation in the manner described in Chapter 7. The main difference is that instead of generating the rationale for a given utterance, we will extract the rationales for all the utterances in the conversation as in Figure 8.3.

8.4.2 Models

We plan to use the same model architectures as for the utterance classification task, which includes instruct-tuned models like Flan-T5 and InstructDial, as well as few-shot prompted LLMs like Llama2, Mixtral and ChatGPT.

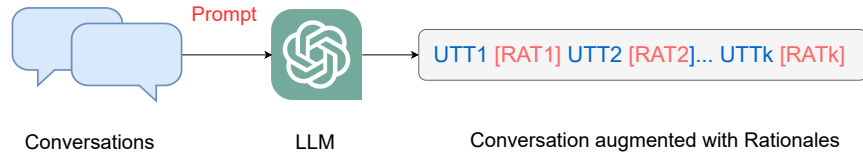


Figure 8.3: Generating rationales per conversation. We will use our prompting framework to generate these rationales.

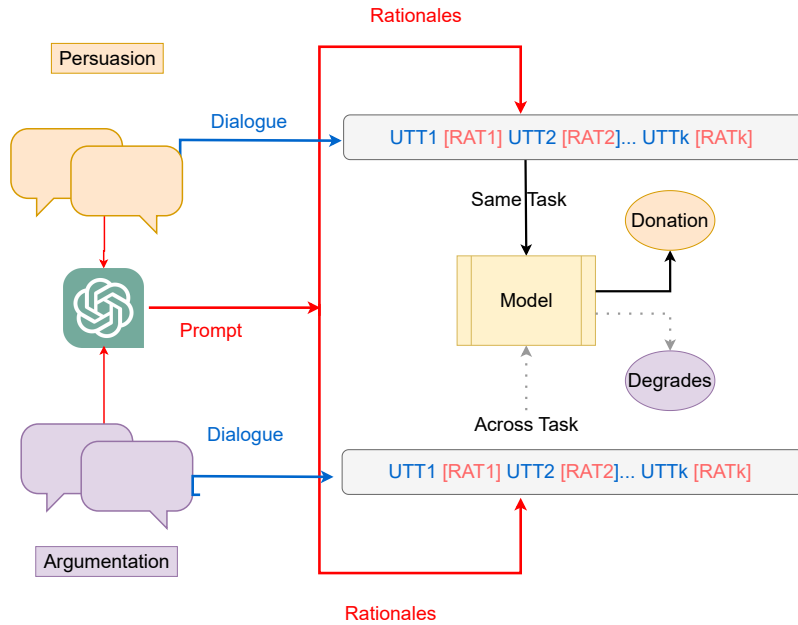


Figure 8.4: Proposed design framework to answer whether rationales facilitate understanding of conversational dynamics and whether the understanding can be generalized

8.4.3 Task Setup

We propose the following three research hypothesis.

RQ1: Do rationales facilitate understanding of conversational dynamics?

RQ2: Can the task of understanding conversational success be transferred?

Similar to the past section, we adapt the task framework to answer these two questions, as shown in Figure 8.4. To answer RQ1, we simply observe the change in model performance in presence and absence of the rationales for the three different datasets and settings as described above. To answer RQ2, we will train the model on one prediction task and use the fine-tuned version of the model to predict the success on the second task. For both cases, we need to pass to the model, the definition of conversation success. Similar to the past idea, we can glean the performance improvements both in a few-shot and zero-shot setting.

8.5 Rationale Generation

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 LLaMA [Touvron 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. \[2023b\]](#), and we wish to observe how well can these rationales be generated without reliance on closed-source LLMs.

Part III

Evaluating Scaffolds

Chapter 9

[Proposed Work] Evaluating Generalization through the lens of scaffolds

9.1 Introduction

In our past endeavours, we have established the ability of scaffolds, both formal and informal, to generalize to different kinds of natural language understanding tasks. We have also leveraged these scaffolds as some form of stress-tests to identify or characterize certain sub-populations that are more or less amenable to generalization by different models.

For example, in our past work GrailQA++, we have used isomorphisms as a lens to inspect model performance and discovered how different KBQA systems have complementary performance; for example ArcaneQA was better suited to deal with longer hops whereas RNG-KBQA was better for questions with more constraints. Likewise, in the PerKGQA work, we also discovered how adding in path information and node embeddings improved generalization performance across multiple dimensions, such as number of hops, number of head/starting entities, and the size of the retrieved subgraph amongst others.

In this section, we aim to understand the utility of different scaffolds through a principled and rigorous evaluation process. We use natural language inference or textual entailment (NLI) as the case study to investigate where and why different rationales help to generalize. We focus on NLI specifically as the task of our choice for several reasons.

Firstly, NLI is an important natural language understanding task that has garnered significant attention in the language technologies community over the past decade. Consequently, it has resulted not only in the creation of standardized massive datasets like SNLI and MNLI for indomain evaluation, but has also spurred research across different generalization axes; namely domains (like medicine and legal), robustness and, and input compositionality.

Moreover, several works have also underscored the benefits of incorporating scaffolds like linguistic frameworks and rationales on NLI performance. This enables us to inspect the cases where scaffolds are beneficial and why, without worrying about the basic premise, (i.e. Do scaffolds help NLI?) Additionally certain NLI datasets enable us to organize the input into different taxonomic categories and map performance gains to different input populations, whereas other datasets are constructed in a contrastive/ counterfactual manner and thus facilitates analysing

the output differences across differences in inputs.

9.2 Experimental Setup

In this section formalize the different datasets that correspond to certain generalization criteria (dimensions), model architectures, and formal/informal scaffolds that we intend to leverage.

9.2.1 Datasets

For NLI, we want to explore the various types of generalization as noted in [Hupkes et al. \[2023\]](#). Subsequently, 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.

- **Domains:** To see the impact of incorporating scaffolds across domains, we first models trained on a given source domains, and evaluate it on other targets. We consider SNLI as our initial source domain for all settings and conditions. We include both the matched and mis-matched splits of the Multi-Genre NLI (MNLI) dataset to test generalization to other generic or general domains. We will also experiment on the MedNLI [Romanov and Shivade \[2018\]](#) and the Contract-NLI [Koreeda and Manning \[2021\]](#) to evaluate performance on the medical and legal domains respectively.
- **Robustness or Challenge sets:** We also want to investigate the ability of scaffolds to generalize to different adversarial constructions or challenge sets. We thus use SNLI-Hard [Gururangan et al. \[2018\]](#), and the counter-factual NLI or CNLI [Kaushik et al. \[2019\]](#) dataset as valid targets.
- **Compositionality:** We aim to use the datasets SETI [Fu and Frank \[2023\]](#), MoNLI [Geiger et al. \[2020\]](#), and ConJNLI [Saha et al. \[2020\]](#) to characterize generalization to different input compositions.
- **Input categorization:** To characterize different properties of the inputs, we will also use the TaxiNLI dataset [Joshi et al. \[2020\]](#) that provides an hierarchical taxonomy of a subset of the MNLI dataset and categorizes the inputs on the basis of whether it requires linguistic, logical, or world knowledge. Likewise, we also want to evaluate on the HANS [McCoy et al. \[2020\]](#) that inspects different syntactic heuristics in NLI on the basis of lexical overlap, subsequence, and constituents.

Overall, this large collection of datasets makes the task of NLI more amenable to evaluate generalization.

9.2.2 Model Frameworks

Since we are interested in a thorough evaluation paradigm, we inspect the role of incorporating scaffolds to a wide variety of model frameworks, which can be categorized as follows.

- **Architectures:** We will 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. [2020a] for (ED).

- **Sizes:** We also plan to experiment with varying model sizes such as small, base (or medium), and large variants for each of those models, wherever applicable, to observe the impact of scale on generalization capabilities.
- **Parameter-efficient models:** We also wish to explore how different parameter-efficient techniques like adapters Houlsby et al. [2019b] and LoRA Hu et al. [2021b], influence generalizability across different settings.

All these models, will be trained initially on the SNLI source dataset and will then be used for evaluation on all of these target domains as discussed above. We will inspect each model in isolation (i.e. absence of any scaffold which serves as a baseline), and also after incorporating the corresponding scaffold. We now discuss our choice of scaffolds and how we leverage them below.

9.2.3 Choice of scaffolds

We explore the role of both formal and informal scaffolds for NLI in our work.

Formal scaffolds: The formal scaffolds correspond to different linguistic frameworks that encapsulate different syntactic and semantic information. We use dependency graphs to model syntax and with parses obtained from the Stanza CoreNLP library Qi et al. [2020a]. For modelling semantics, we leverage of Abstract Meaning Representations or AMRs, and will experiment with the AMR parser of Vasylenko et al. [2023] which incorporates the structural information at training time or the neural IBM Transition Parser Drozdov et al. [2022].

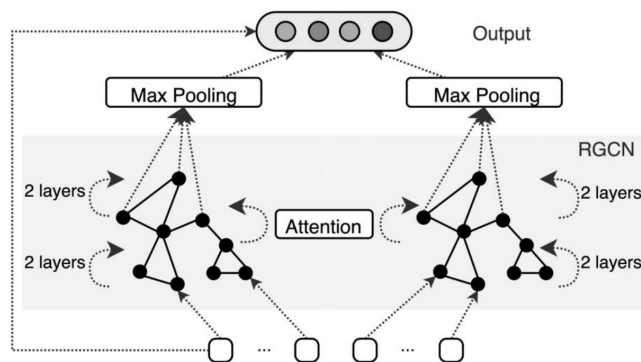


Figure 9.1: We use the proposed SIFT architecture of Wu et al. [2021] to incorporate semantic/syntactic dependencies with the baseline model architecture for the task of NLI prediction.

We adopt the same model architecture as that of Wu et al. [2021] to infuse the corresponding syntactic or semantic parses with the different pre-trained language models. Specifically, given a pair of premise and hypothesis, the concatenated pair of sentences is first passed through an LM (such as BERT or T5) to obtain a representation for each token. These tokens are then used to initialize the corresponding nodes in the semantic or syntactic parse for the premise and the

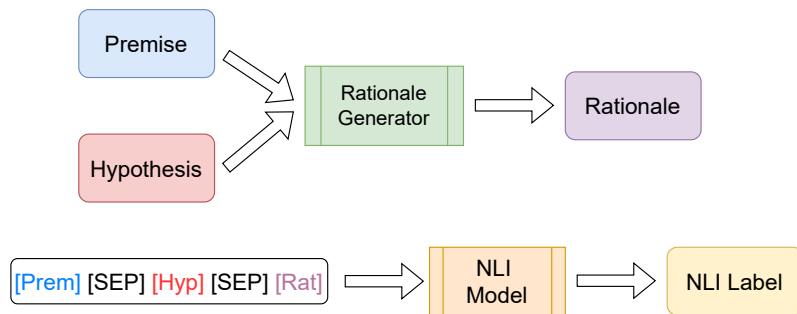


Figure 9.2: Model Pipeline for generating rationales and infusing them for NLI

hypothesis separately. These parses are then passed through a GNN specifically RGCN to update their representations. This step is followed by cross-attention over the updated representations after which they are subsequently passed through another GNN module and max pooled to obtain the final representations for both the premise and hypothesis. These updated representations are concatenated with the [CLS] representation of the original module and then passed through the NLI module to obtain the final output. We provide a pictorial representation of our chosen architecture in Figure 9.1.

Informal scaffolds: Past research has demonstrated the impact of adding natural language explanations or rationales to improve the reasoning ability and facilitate reasoning in NLI. To that end, we explore different kinds of rationales or explanations that we provide as additional input to the model. We showcase a simple representation of the rationale generation process and its subsequent inclusion into the NLI prediction module in Figure 9.2.

We will follow the pipelined approach of generating rationales for NLI where the rationales are produced by a separate generator model and is then provided as input to the classification model, as opposed to generating the rationales and predicting the NLI labels in a joint fashion. We propose this approach since it affords the flexibility of experimenting with different model architectures (like encoder models) which are be suitable for text generation. In this case, we will experiment with recent fine-tuned models like Flan-T5 [Chung et al. \[2022\]](#) and LLama-2 [Touvron et al. \[2023\]](#) for the task of rationale generation. We will use the e-SNLI dataset of [Camburu et al. \[2018\]](#) that contains human-written natural language rationales as the dataset for rationale generation, since it is a natural extension of SNLI, the source dataset for all our experiments.

Measuring the utility of scaffolds: For formal scaffolds, we will use a probe-based classifier to observe how the presence of certain syntactic and semantic features (nodes) or relations (edges) in the parse can be attributed to performance improvements. For informal scaffolds or rationales, we explore different attribution methods that inspect the rationales text to assess their importance. These methods include Shapley Value [Lundberg and Lee \[2017\]](#), integrated gradients [Wiegrefe et al. \[2021\]](#), and attention-based approaches [Wiegrefe and Pinter \[2019\]](#).

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. 1.1, 7.3.3
- Dhruv Agarwal, Rajarshi Das, Sopan Khosla, and Rashmi Gangadharaiyah. Bring your own kg: Self-supervised program synthesis for zero-shot kgqa. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*. 6.4.3
- 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>. 2.3, 7.4.3
- 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. 7.2.1
- 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>. 4.2
- Jean-michel Attendu and Jean-philippe Corbeil. NLU on data diets: Dynamic data subset selection for NLP classification tasks. In Nafise Sadat Moosavi, Iryna Gurevych, Yufang Hou, Gyuwan Kim, Young Jin Kim, Tal Schuster, and Ameeta Agrawal, editors, *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustainLP)*, pages 129–146, Toronto, Canada (Hybrid), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.sustainlp-1.9. URL <https://aclanthology.org/2023.sustainlp-1.9>. 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>. 3.2.2, 3.3.4
- 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>. 3.2.1
- 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, 3.3.2
- Elisa Bassignana, Filip Ginter, Sampo Pyysalo, Rob van der Goot, and Barbara Plank. Silver syntax pre-training for cross-domain relation extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6984–6993, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.436. URL <https://aclanthology.org/2023.findings-acl.436>. 2.1
- Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima’an. Graph convolutional encoders for syntax-aware neural machine translation. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1209. URL <https://aclanthology.org/D17-1209>. 2.1
- 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>. 4.6.3, 7.4.3
- Niels Ole Bernsen, Hans Dybkjær, and Laila Dybkjær. Cooperativity in human-machine and human-human spoken dialogue. *Discourse processes*, 21(2):213–236, 1996. 8.1
- 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. 3.3.2
- Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 01 2009. ISBN 978-0-596-51649-9. 3.3.3
- John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In Dan Jurafsky and Eric Gaussier, editors, *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July 2006. Association for Computational Linguistics. URL <https://aclanthology.org/D06-1011>. 3.2.1

- [//aclanthology.org/W06-1615](https://aclanthology.org/W06-1615). 2.3
- Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004. 2.1
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008. 2.1
- 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. 4.1.2, 4.6.1, 4.6.2
- 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>. 9.2.1
- Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. Libkg—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. 4.6.2
- Penelope Brown, Stephen C Levinson, and Stephen C Levinson. *Politeness: Some universals in language usage*, volume 4. Cambridge university press, 1987. 7.2.2, 8.1
- Jan Buchmann, Max Eichler, Jan-Micha Bodensohn, Ilia Kuznetsov, and Iryna Gurevych. Document structure in long document transformers. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1056–1073, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.eacl-long.64>. 2.1
- 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. (document), 7.4.1, 5
- Rui Cai and Mirella Lapata. Syntax-aware semantic role labeling without parsing. *Transactions of the Association for Computational Linguistics*, 7:343–356, 2019. doi: 10.1162/tacl.a.00272. URL <https://aclanthology.org/Q19-1022>. 2.1
- 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. 2.3
- Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In

- Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3829–3839, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.408>. 2.1
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31, 2018. 9.2.3
- Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Retrieve, rerank and rewrite: Soft template based neural summarization. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1015. URL <https://aclanthology.org/P18-1015>. 2.2
- 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>. ??
- 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>. 3.2.1
- Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. Improved neural machine translation with a syntax-aware encoder and decoder. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1177. URL <https://aclanthology.org/P17-1177>. 2.1
- Jifan Chen, Yuhao Zhang, Lan Liu, Rui Dong, Xinchu 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
- Mingda Chen, Zewei Chu, Karl Stratos, and Kevin Gimpel. Mining knowledge for natural language inference from Wikipedia categories. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3500–3511, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.313. URL <https://aclanthology.org/2020.findings-emnlp.313>. 2.2
- Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. Hytrel: Hypergraph-enhanced tabular data representation learning. *Advances in Neural Information Processing Systems*, 36, 2024. 1.2
- Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. KACE: Generating knowledge aware contrastive explanations for natural language inference. 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 2516–2527, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.196. URL <https://aclanthology.org/2021.acl-long.196>. 2.2
- 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, 2021b. 4.2
- Tianyu Chen, Shaohan Huang, Furu Wei, and Jianxin Li. Pseudo-label guided unsupervised domain adaptation of contextual embeddings. In Eyal Ben-David, Shay Cohen, Ryan McDonald, Barbara Plank, Roi Reichart, Guy Rotman, and Yftah Ziser, editors, *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 9–15, Kyiv, Ukraine, April 2021c. Association for Computational Linguistics. URL <https://aclanthology.org/2021.adaptnlp-1.2>. 2.3
- Xiang Chen, Yue Cao, and Xiaojun Wan. WIND: Weighting instances differentially for model-agnostic domain adaptation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2366–2376, Online, August 2021d. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.209. URL <https://aclanthology.org/2021.findings-acl.209>. 2.3
- Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1053. URL <https://aclanthology.org/D16-1053>. 2.1

- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 1.2, 8.3.2, 8.5, 9.2.3
- Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural scaffolds for citation intent classification in scientific publications. 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 3586–3596, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1361. URL <https://aclanthology.org/N19-1361>. 1.3
- William Croft. *Morphosyntax: constructions of the world’s languages*. 2022. 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, 2013. 7.2.2
- 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. 2.2, 7.2.3
- 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>. 4.2
- 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>. 4.2, 4.5.1
- 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, 2021a. 4.2
- 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. 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 9594–9611, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.755. URL <https://aclanthology.org/2021.emnlp-main.755>. 2.1
- 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, 2021. 4.2

- 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/das22a.html>. 5.1, 5.3.1
- Hal Daumé III. Frustratingly easy domain adaptation. In Annie Zaenen and Antal van den Bosch, editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <https://aclanthology.org/P07-1033>. 2.3
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024. 2.3
- 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 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>. 1.1, 2.3, 9.2.2
- 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>. 3.3.4, 7.4.3
- 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>. 4.2
- Yijiang Dong, Lara Martin, and Chris Callison-Burch. CoRRPUS: Code-based structured prompting for neurosymbolic story understanding. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13152–13168, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.832. URL <https://aclanthology.org/2023.findings-acl.832>. 2.1
- 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 Compu-*

- tational 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>. 4.6.3, 7.4.3
- Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramón Astudillo. Inducing and using alignments for transition-based AMR parsing. 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 1086–1098, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.80. URL <https://aclanthology.org/2022.naacl-main.80>. 9.2.3
- Wenyu Du, Zhouhan Lin, Yikang Shen, Timothy J. O’Donnell, Yoshua Bengio, and Yue Zhang. Exploiting syntactic structure for better language modeling: A syntactic distance approach. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6611–6628, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.591. URL <https://aclanthology.org/2020.acl-main.591>. 2.1
- Zhichao Duan, Xiuxing Li, Zhenyu Li, Zhuo Wang, and Jianyong Wang. Not just plain text! fuel document-level relation extraction with explicit syntax refinement and subsentence modeling. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1941–1951, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.140. URL <https://aclanthology.org/2022.findings-emnlp.140>. 2.1
- 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>. 7.2.2
- 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>. (document), 7.1, 7.4.1, 4
- Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiyah, Dan Roth, and Carolyn Rose. PerKGQA: Question answering over personalized knowledge graphs. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 253–268, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.19. URL <https://aclanthology.org/2022.findings-naacl.19>. 1.4

- Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiiah, 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, 2022b. 2
- Ritam Dutt, Sopan Khosla, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiiah. GrailQA++: A challenging zero-shot benchmark for knowledge base question answering. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–909, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.58. URL <https://aclanthology.org/2023.ijcnlp-main.58>. 1.4, 6.4.3
- Santiago Egea Gómez, Euan McGill, and Horacio Saggion. Syntax-aware transformers for neural machine translation: The case of text to sign gloss translation. In Reinhard Rapp, Serge Sharoff, and Pierre Zweigenbaum, editors, *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 18–27, Online (Virtual Mode), September 2021. INCOMA Ltd. URL <https://aclanthology.org/2021.bucc-1.4>. 2.1
- Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 933–952, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.59. URL <https://aclanthology.org/2023.emnlp-main.59>. 2.1
- Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. MultiDoc2Dial: Modeling dialogues grounded in multiple documents. 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 6162–6176, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.498. URL <https://aclanthology.org/2021.emnlp-main.498>. 2.2
- Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 4.6.2
- 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>. 3.3.3
- Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. Social chemistry 101: Learning to reason about social and moral norms. 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 653–670, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.48. URL <https://aclanthology.org/2020.emnlp-main.48>. 2.1

- 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>. 9.2.1
- James Paul Gee. *An introduction to discourse analysis: Theory and method*. routledge, 2014. 7.2.1
- 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>. 9.2.1
- Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. 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 154–164, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1015. URL <https://aclanthology.org/D19-1015>. 2.1
- Deepanway Ghosal, Somak Aditya, and Monojit Choudhury. Prover: Generating intermediate steps for NLI with commonsense knowledge retrieval and next-step prediction. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 872–884, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.56. URL <https://aclanthology.org/2023.ijcnlp-main.56>. 2.2
- Erving Goffman. Front and back regions of everyday life [1959]. *The everyday life reader*, pages 50–57, 2002. 7.2.1, 7.3.1
- Ralph Grishman. The role of syntax in information extraction. In *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop held at Vienna, Virginia, May 6-8, 1996*, pages 139–142, Vienna, Virginia, USA, May 1996. Association for Computational Linguistics. doi: 10.3115/1119018.1119051. URL <https://aclanthology.org/X96-1029>. 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. 4.5.2
- 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. 4.2, 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*, 2022a. 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
- Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. PPT: Pre-trained prompt tuning for few-shot learning. 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 8410–8423, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.576. URL <https://aclanthology.org/2022.acl-long.576>. 2.3
- Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hongye Tan. Frame semantic-enhanced sentence modeling for sentence-level extractive text summarization. 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 4045–4052, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.331. URL <https://aclanthology.org/2021.emnlp-main.331>. 2.1
- 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>. 7.2.2, 8.3.2
- 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>. 7.2.3
- Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rosé. Linguistic representations for fewer-shot relation extraction across domains. 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 7502–7514, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.414. URL <https://aclanthology.org/2023.acl-long.414>. 1.3, 1.4
- 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, 2023b. 2.1, 7.5
- 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. 9.2.1
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL <https://aclanthology.org/2020.acl-main.740>. 2.3
- Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. 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 4921–4933, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.381. URL <https://aclanthology.org/2021.acl-long.381>. 2.3
- Jiale Han, Bo Cheng, and Xu Wang. Open domain question answering based on text enhanced knowledge graph with hyperedge infusion. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1475–1481, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.133. URL <https://aclanthology.org/2020.findings-emnlp.133>. 2.1
- Boran Hao, Henghui Zhu, and Ioannis Paschalidis. Enhancing clinical BERT embedding using a biomedical knowledge base. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 657–661, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.57. URL <https://aclanthology.org/2020.coling-main.57>. 2.1
- 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. 4.2, 4.6.1
- 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. 7.2.1, 7.4.1
- 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. 7.2.2, 7.4.1

- 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 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1256. URL <https://aclanthology.org/D18-1256>. (document), ??, 8.2.2, 4
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2020. 9.2.2
- 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, 2019a. 2.3
- 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, 2019b. 9.2.2
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, 2023. URL <https://aclanthology.org/2023.findings-acl.507.pdf>. 2.2
- 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>. 7.4.1
- I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. AMPERE: AMR-aware prefix for generation-based event argument extraction model. 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 10976–10993, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.615. URL <https://aclanthology.org/2023.acl-long.615>. 1.2, 2.1
- 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*, 2021a. 2.3
- 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*, 2021b. 9.2.2
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.67. URL <https://aclanthology.org/2023.emnlp-main.67.22>
- 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. 4.1.2
- 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), 1.1, 1.1, 1.2, 2.3, 9.2.1
- Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. 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 874–880, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74. URL <https://aclanthology.org/2021.eacl-main.74.22>
- Parag Jain and Mirella Lapata. Memory-based semantic parsing. *Transactions of the Association for Computational Linguistics*, 9:1197–1212, 2021. doi: 10.1162/tacl.a_00422. URL <https://aclanthology.org/2021.tacl-1.71.2.1>
- 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.450.2.2,7.2.3>
- Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In Annie Zaenen and Antal van den Bosch, editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <https://aclanthology.org/P07-1034.2.3>
- Yiwei Jiang, Klim Zaporozhets, 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.82.3.4,3.4>
- Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R. Lyu. HiGRU: Hierarchical gated

- recurrent units for utterance-level emotion recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 397–406, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1037. URL <https://www.aclweb.org/anthology/N19-1037>. 2.1
- Wenxiang Jiao, Michael Lyu, and Irwin King. Real-time emotion recognition via attention gated hierarchical memory network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8002–8009, 2020. 2.1
- Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. 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 2763–2775, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.197. URL <https://aclanthology.org/2022.acl-long.197>. 2.3
- 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>. 2.2, 7.2.3
- 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. 2.2, 7.2.3
- Pratik Joshi, Somak Aditya, Aalok Sathe, and Monojit Choudhury. TaxiNLI: Taking a ride up the NLU hill. In Raquel Fernández and Tal Linzen, editors, *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 41–55, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.4. URL <https://aclanthology.org/2020.conll-1.4>. 9.2.1
- Haein Jung, Heuiyeen Yeen, Jeehyun Lee, Minju Kim, Namoo 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>. 7.3.2
- 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. 7.2.1

- 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>. 4.1.2
- Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. Syntax-aware neural semantic role labeling with supertags. 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 701–709, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1075. URL <https://aclanthology.org/N19-1075>. 2.1
- 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. 9.2.1
- 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. 7.2.1
- Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. Ctrlval: 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 Ibrahim Alabdullah, 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. 7.2.2
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019. 1.3, 2.1
- Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In *International Conference on Learning Representations*, 2020. 2.2
- Sopan Khosla and Rashmi Gangadharaiah. Benchmarking the covariate shift robustness of open-world intent classification approaches. *AACL-IJCNLP 2022*, page 14, 2022. 7.2.2
- Sopan Khosla, Ritam Dutt, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. Exploring the reasons for non-generalizability of KBQA systems. In Shabnam Tafreshi, Arjun Akula, João Sedoc, Aleksandr Drozd, Anna Rogers, and Anna Rumshisky, editors, *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 88–93, Dubrovnik, Croatia, May 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.insights-1.11. URL <https://aclanthology.org/2023.insights-1.11>. 1.4

- 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, 2023b. 5.4.2, 5.6
- Joongwon Kim, Akari Asai, Gabriel Ilharco, and Hannaneh Hajishirzi. TaskWeb: Selecting better source tasks for multi-task NLP. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11032–11052, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.680. URL <https://aclanthology.org/2023.emnlp-main.680>. 2.3
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 3.5.1
- 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, 3.4
- Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.164. URL <https://aclanthology.org/2021.findings-emnlp.164>. 9.2.1
- Fajri Koto, Jey Han Lau, and Timothy Baldwin. Improved document modelling with a neural discourse parser. In Meladel Mistica, Massimo Piccardi, and Andrew MacKinlay, editors, *Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association*, pages 67–76, Sydney, Australia, 4–6 December 2019. Australasian Language Technology Association. URL <https://aclanthology.org/U19-1010>. 2.1
- Sawan Kumar and Partha Talukdar. NILE : Natural language inference with faithful natural language explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.771. URL <https://aclanthology.org/2020.acl-main.771>. 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>. 4.1.2, 4.2, 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

- Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. 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 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL <https://aclanthology.org/2021.emnlp-main.243>. 2.3
- 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, 2020a. 9.2.2
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020b. 2.2
- 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, 2020. 7.2.2
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL <https://aclanthology.org/2021.acl-long.353>. 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>. 4.2
- 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. 7.3.4
- 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
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022a. 2.3

- 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 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.62. URL <https://aclanthology.org/2022.findings-acl.62>. 3.2.1
- 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*, 2022c. 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. 4.5.2, 4.6.1, 4.6.2
- 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. 9.2.2
- Niklas Lüdemann, Ageda Shiba, Nikolaos Thymianis, Nicolas Heist, Christopher Ludwig, and Heiko Paulheim. A knowledge graph for assessing aggressive tax planning strategies. In *International Semantic Web Conference*, pages 395–410. Springer, 2020. 4.1.2
- Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 9.2.3
- 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
- Fukun Ma, Xuming Hu, Aiwei Liu, Yawen Yang, Shuang Li, Philip S. Yu, and Lijie Wen. AMR-based network for aspect-based sentiment analysis. 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 322–337, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.19. URL <https://aclanthology.org/2023.acl-long.19>. 2.1
- Aman Madaan. *Enhancing Language Models with Structured Reasoning*. PhD thesis, Language Technologies Institute, Carnegie Mellon University, 2024. 1.2
- Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. Memory-assisted prompt editing to improve GPT-3 after deployment. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861, Abu Dhabi, United Arab Emirates, December 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.183. URL <https://aclanthology.org/2022.emnlp-main.183>. 2.1
- 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 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.90. URL <https://aclanthology.org/2022.emnlp-main.90>. 2.1, 7.3.2
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024. 2.2
- Gaurav Maheshwari, Priyansh Trivedi, Denis Lukovnikov, Nilesch 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. 4.2
- 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. 7.2.1
- 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. 1.3, 2.2, 7.2.3
- James R Martin and Peter R White. *The language of evaluation*, volume 2. Springer, 2003. 7.2.1, 8.1
- James Robert Martin and David Rose. *Working with discourse: Meaning beyond the clause*. Bloomsbury Publishing, 2003. 7.2.1
- 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. 9.2.1
- Shikib Mehri. *Towards Generalization in Dialog through Inductive Biases*. PhD thesis, Language Technologies Institute, Carnegie Mellon University, 2022. 7.2.2
- Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. Mixture-of-partitions: Infusing large biomedical knowledge graphs into BERT. 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 4672–4681, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.383. URL <https://aclanthology.org/2021.emnlp-main.383>. 2.1
- Miriam Meyerhoff. In pursuit of social meaning. *Journal of Sociolinguistics*, 23(3):303–315, 2019. 7.2.1
- 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. 7.3.2
- 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 Confer-*

- ence 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>. 4.2
- 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. 7.2.1
- 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>. 3.4
- 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>. 3.2.1
- Aakanksha Naik, Jill Lehman, and Carolyn Rosé. Adapting to the long tail: A meta-analysis of transfer learning research for language understanding tasks. *Transactions of the Association for Computational Linguistics*, 10:956–980, 2022. doi: 10.1162/tacl_a_00500. URL <https://aclanthology.org/2022.tacl-1.56>. 1.1, 2.3
- Rungsiman Nararatwong, Natthawut Kertkeidkachorn, and Ryutaro Ichise. KIQA: Knowledge-infused question answering model for financial table-text data. In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 53–61, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.6. URL <https://aclanthology.org/2022.deelio-1.6>. 2.1
- 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>. 3.3.1
- 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. 7.2.1

- Noriki Nishida and Yuji Matsumoto. Out-of-domain discourse dependency parsing via bootstrapping: An empirical analysis on its effectiveness and limitation. *Transactions of the Association for Computational Linguistics*, 10:127–144, 2022. doi: 10.1162/tacl_a.00451. URL <https://aclanthology.org/2022.tacl-1.8.23>
- Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.313. URL <https://aclanthology.org/2023.emnlp-main.313.2.1>
- 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. 4.1.2
- 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. 7.2.2
- 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.298.3.2.1>
- 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. 4.5.2, 4.6.2
- 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.??>
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL <https://aclanthology.org/D19-1250.2.2>
- 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>. (document), 5
- Matt Post and Daniel Gildea. Parsers as language models for statistical machine translation. In *Proceedings of the 8th Conference of the Association for Machine Translation in the Americas: Research Papers*, pages 172–181, Waikiki, USA, October 21-25 2008. Association for Machine Translation in the Americas. URL <https://aclanthology.org/2008.amta-papers.16>. 2.1
- Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. What to pre-train on? Efficient intermediate task selection. 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 10585–10605, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.827. URL <https://aclanthology.org/2021.emnlp-main.827>. 2.3
- 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>. 3.3.3
- 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>. 2.1, 3.2.2
- Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. Intermediate-task transfer learning with pretrained language models: When and why does it work? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.467. URL <https://aclanthology.org/2020.acl-main.467>. 2.3
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.14. URL <https://aclanthology.org/2020.acl-demos.14>. 9.2.3

- 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*, 2020b. URL <https://nlp.stanford.edu/pubs/qi2020stanza.pdf>. 3.3.2
- 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>. 3.2.1
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1.1, 7.4.3, 9.2.2
- 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. 6.4.1, 7.4.3, 9.2.2
- Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1487. URL <https://aclanthology.org/P19-1487>. 2.2
- Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.603. URL <https://aclanthology.org/2020.coling-main.603>. 2.3
- 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>. 2.2, 7.1, 7.2.3, 8.5
- 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. 4.1.2, 4.2
- 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>. 9.2.1
- 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>. 3.3.1
- 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. 4.6.2
- Qian Ruan, Malte Ostendorff, and Georg Rehm. HiStruct+: Improving extractive text summarization with hierarchical structure information. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.102. URL <https://aclanthology.org/2022.findings-acl.102>. 2.1
- Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2.3
- 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>. 2.1, 3.2.2
- 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>. 4.2
- 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>. 9.2.1
- 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>. 2.2, 7.2.3
- 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>. 2.2, 7.2.3
- Gabriel Sarch, Yue Wu, Michael Tarr, and Katerina Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3468–3500, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.226. URL <https://aclanthology.org/2023.findings-emnlp.226>. 2.1
- 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>. 4.2, 4.6.1, 4.6.2
- 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. 3.3.4, 4.5.2, 4.6.2
- Hendrik Schuff, Hsiu-Yu Yang, Heike Adel, and Ngoc Thang Vu. Does external knowledge help explainable natural language inference? automatic evaluation vs. human ratings. In Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad, editors, *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 26–41, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.blackboxnlp-1.3. URL <https://aclanthology.org/2021.blackboxnlp-1.3>. 2.2
- 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>. ??

- Yikang Shen, Shawn Tan, Alessandro Sordoni, Siva Reddy, and Aaron Courville. Explicitly modeling syntax in language models with incremental parsing and a dynamic oracle. 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 1660–1672, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.132. URL <https://aclanthology.org/2021.naacl-main.132>. 2.1
- Yiheng Shu and Zhiwei Yu. Distribution shifts are bottlenecks: Extensive evaluation for grounding language models to knowledge bases. In Neele Falk, Sara Papi, and Mike Zhang, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 71–88, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.eacl-srw.7>. 6.4.3
- Yiheng Shu, Zhiwei Yu, Yuhao 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. 4.2
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. 2.2
- Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 2.1
- 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>. 4.5.1, 4.5.2
- 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, 6.4.3
- 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>.

org/D18-1455. 4.2

- 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>. 4.2, 4.5.1
- Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. Syntactic scaffolds for semantic structures. 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 3772–3782, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1412. URL <https://aclanthology.org/D18-1412>. 1.3
- Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. 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 9275–9293, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.746. URL <https://aclanthology.org/2020.emnlp-main.746>. 2.3
- 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>. 5.4.2
- Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 339–352, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.26. URL <https://aclanthology.org/2022.findings-naacl.26>. 2.1
- Dung Thai, Dhruv Agarwal, Mudit Chaudhary, Wenlong Zhao, Rajarshi Das, Jay-Yoon Lee, Hannaneh Hajishirzi, Manzil Zaheer, and Andrew McCallum. Machine reading comprehension using case-based reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8414–8428, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.564. URL <https://aclanthology.org/2023.findings-emnlp.564>. 2.2
- Yuanhe Tian, Yan Song, and Fei Xia. Improving relation extraction through syntax-induced pre-training with dependency masking. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*,

- pages 1875–1886, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.147. URL <https://aclanthology.org/2022.findings-acl.147>. 2.1
- 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. 1.1, 6.4.1, 7.3.4, 7.4.3, 8.5, 9.2.3
- Jackson Trager, Alireza S. Ziabari, Aida Mostafazadeh Davani, Prezi Golazazian, Farzan Karimi-Malekabadi, Ali Omrani, Zhihe Li, Brendan Kennedy, Nils Karl Reimer, Melissa Reyes, Kelsey Cheng, Mellow Wei, Christina Merrifield, Arta Khosravi, Evans Alvarez, and Morteza Dehghani. The moral foundations reddit corpus, 2022. ??
- 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. 4.6.1
- Marlies van der Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1147. URL <https://aclanthology.org/D17-1147>. 2.3
- Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. RESIDE: Improving distantly-supervised neural relation extraction using side information. 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 1257–1266, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1157. URL <https://aclanthology.org/D18-1157>. 2.1
- Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1320. URL <https://aclanthology.org/P19-1320>. 2.1
- Pavlo Vasylenko, Pere Lluís Hugué Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli. Incorporating graph information in transformer-based AMR parsing. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1995–2011, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.125. URL <https://aclanthology.org/2023.findings-acl.125>. 9.2.3
- Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. 2.1
- Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou’, and Daniel Cer. SPoT: Better frozen model adaptation through soft prompt transfer. 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 5039–5059, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.346. URL <https://aclanthology.org/2022.acl-long.346>. 2.3
- 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 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.67. URL <https://aclanthology.org/2022.findings-acl.67>. 3.2.1
- Cunxiang Wang, Pai Liu, and Yue Zhang. Can generative pre-trained language models serve as knowledge bases for closed-book QA? 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 3241–3251, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.251. URL <https://aclanthology.org/2021.acl-long.251>. 2.2
- Ke Wang, Jiayi Wang, Niyu Ge, Yangbin Shi, Yu Zhao, and Kai Fan. Computer assisted translation with neural quality estimation and automatic post-editing. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2175–2186, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.197. URL <https://aclanthology.org/2020.findings-emnlp.197>. 2.1
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021b. doi: 10.1162/tacl_a_00360. URL <https://aclanthology.org/2021.tacl-1.11>. 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, 2020b. 4.2, 4.5.2, 4.6.1, 4.6.2
- 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 2020c. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.231. URL <https://aclanthology.org/2020.coling-main.231>. 4.2, 4.5.1, 4.5.2, 4.6.1, 4.6.2, 4.7
- 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 Anna Korhonen, David Traum, and Lluís Màrquez, editors, *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://aclanthology.org/P19-1566>. (document), ??, ??, 8.2.2, 8.4, 4
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2022b. 2.2
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022c. 1.2, 1.2
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. 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 13484–13508, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL <https://aclanthology.org/2023.acl-long.754>. 1.2
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. 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 13484–13508, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL <https://aclanthology.org/2023.acl-long.754>. 2.2
- 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. 7.3.3
- 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. 2.2, 7.2.3
- 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. 1.3, 2.2
- Sarah Wiegrefe and Yuval Pinter. Attention is not not explanation. 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 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL <https://aclanthology.org/D19-1002>. 9.2.3
- Sarah Wiegrefe, 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>. 1.3, 2.2, 7.1, 7.2.3, 9.2.3
- Zhaofeng Wu, Hao Peng, and Noah A. Smith. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242, 2021. doi: 10.1162/tacl.a_00363. URL <https://aclanthology.org/2021.tacl-1.14>. (document), 1.3, 2.1, 9.1, 9.2.3
- Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, pages 1391–1399, 2017. ??
- 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>. 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>. 2.2, 7.2.3
- 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>. 4.2
- Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. 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 9514–9528, Online and Punta Cana, Dominican Republic, November 2021.

- Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.749. URL <https://aclanthology.org/2021.emnlp-main.749>. 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>. 3.2.2
- 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>. 3.4
- 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. 3.2.1
- 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>. 4.2
- Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. 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 7163–7189, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.572. URL <https://aclanthology.org/2021.emnlp-main.572>. 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*, 2021b. 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>. 3.2.1
- 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>. 3.2.1
- 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>. 4.1.3, 4.4.2, 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
- Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain 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 4961–4974, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.340. URL <https://aclanthology.org/2022.acl-long.340>. 2.1
- 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 2022c. International Committee on Computational Linguistics. URL <https://aclanthology.org/2022.coling-1.205>. 3.2.1
- 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. 2.2, 7.2.3
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022. 2.2, 7.1, 7.2.3
- 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>. 7.2.1
- 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 Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1125. URL <https://aclanthology.org/P18-1125>. ??, 8.2.2, 8.4
- 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. 9.2.2
- Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models for question answering. 2.2
- Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics*, 5:515–528, 2017. doi: 10.1162/tacl_a.00077. URL <https://aclanthology.org/Q17-1036>. 2.3
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL <https://aclanthology.org/P19-1139>. 2.1
- Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. Abstract, rationale, stance: A joint model for scientific claim verification. 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 3580–3586, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.290. URL <https://aclanthology.org/2021.emnlp-main.290>. 1.2
- 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>. 2.1, 3.2.2
- 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)*, 2021b. 3.2.2
- 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>. 3.2.1
- 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>. 3.2.1
- Ruiqi Zhong, Mitchell Stern, and Dan Klein. Semantic scaffolds for pseudocode-to-code generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2283–2295, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.208. URL <https://aclanthology.org/2020.acl-main.208>. 1.3
- Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, et al. How far are large language models from agents with theory-of-mind? 2023a. 2.2
- 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 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.392. URL <https://aclanthology.org/2023.findings-acl.392>. 8.5
- 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>. 4.1.2
- Shuguang Zhu, X. Cheng, and Sen Su. Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing*, 372:64–72, 2020. 4.2
- Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W Huang, Jie Fu, Xiang Yue, and Wenhui Chen. Structlm: Towards building generalist models for structured knowledge grounding. *arXiv preprint arXiv:2402.16671*, 2024. 1.2, 1.2, 1.3, 6.1, 1
- Caleb Ziems, Jane Dwivedi-Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. NormBank: A knowledge bank of situational social norms. 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 7756–7776, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.429. URL <https://aclanthology.org/2023.acl-long.429>. 2.1

Appendices

Appendix A

Dataset Statistics

Figure 3 provides a distribution of labels for the two tasks of ERC and RES across the respective two datasets. Furthermore, Table 4 and Table 5 provide additional insight into the definition of the categories/strategies for the corresponding datasets, as well as representative examples of the same.

Table 3 also presents statistics of the datasets and the corresponding rationales. Each dialog is broken into multiple datapoints, one for each turn in it. The average number of turns per dialogue and the number of words per turn are reported, with IEMOCAP seen to have significantly longer dialogues compared to the rest. The number of rationales generated for the dataset are reported – For P4G and CB, we encounter parsing issues with GPT-3.5’s generated rationales for some instances, which are ignored during training. The average number of words per generated intention/assumption/implicit information is higher for the emotion datasets compared to the resisting strategies ones, which may have been influenced by the choice of the one-shot example in the prompt. The generated implicit information is found to be longer than intention and assumption, and assumption is found to be longer than intention, across all datasets.

Table 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

Qualitative Analysis of Rationales

We present qualitative analysis of the responses generated by LLMs (GPT-3.5-turbo-16k and LLama2-13B-chat-hf) here with Table 1 highlighting the fraction of times the annotators preferred the quality of response generations of ChatGPT to LLama2. Table 2 highlights the average score

Table 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	Grammar	Relevance	Factuality
Friends	5.00	4.55	4.75
IEMOCAP	4.98	4.92	4.34
P4G	5.00	4.52	4.92
CB	5.00	4.55	5.00

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

Table 3: We present here the statistics of the datasets used and the rationales generated.

of two annotators on the quality of responses generated across different datasets in terms of grammaticality, relevance, and factuality.

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 which are not present in the text.

We also provide examples of the actual prompt framework for the ERC and RES in Table 6 and 7 respectively.

Hyperparameter Tuning

We present the hyperparameters for our experiments in Table 8. We carry out the experiments over 3 seeds on a A6000 GPU with early stopping with patience of 5 over the validation set for all experiments. We implement the entire experiments in Python, with help of the Pytorch library and use the pre-trained models as specified in Huggingface under the agreed upon license agreements.

Table 4: Framework describing the resisting strategies for persuasion (P4G) and negotiation (CB) datasets, as specified in [Dutt et al. \[2021\]](#). Examples of each strategy are italicised. The examples for each of P4G and CB were borrowed from the original datasets of the same name from [Wang et al. \[2019\]](#) and [He et al. \[2018\]](#) respectively.

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

Table 5: Framework describing the emotion labels in the emotion recognition datasets (IEMOCAP and Friends) [Busso et al. \[2008\]](#), [Poria et al. \[2019\]](#). Examples of each label are italicised.

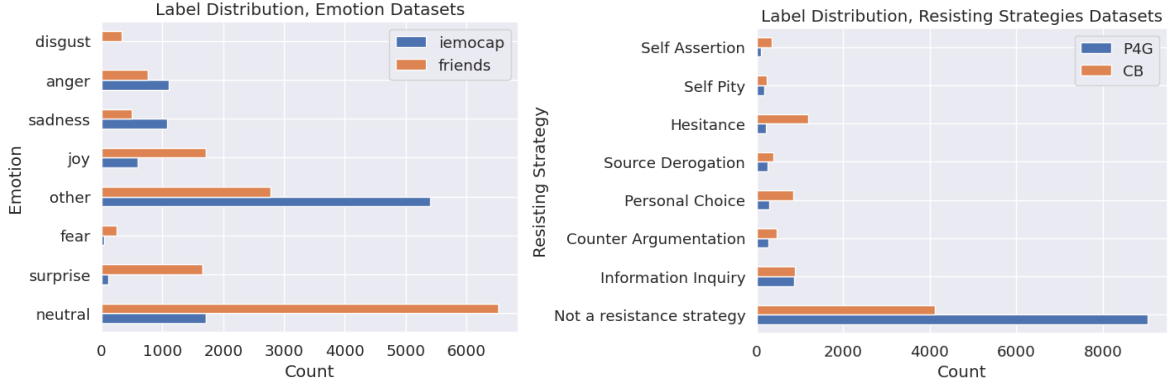
Emotion	IEMOCAP	Friends
Neutral	Neutral emotion is characterized by the absence of strong feelings or emotions. <i>I'll go to basketball games.</i>	Neutral emotion is characterized by the absence of strong feelings or emotions. <i>Yeah, apparently they're turning it into some kinda coffee place.</i>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

Table 6: Below is an example of our prompt for the task of emotion recognition in conversations (ERC).

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

Table 7: Below is an example of our prompt for the task of detecting resisting strategies (RES).

Part 1: High level description of the objective	Analyze this dialogue, focusing on any underlying assumptions and implicit information. Ensure that you address each line individually without skipping or grouping.
Part 2: Step-wise guide	<p>For each line:</p> <ol style="list-style-type: none"> 1. Provide a comprehensive and concise explanation for: <ol style="list-style-type: none"> a)Speaker’s Intention b)Assumptions about the conversation c)Implicit Information 2. Continue until you have analyzed every line.
Part 3: Output Template	<p>Please format your response as follows:</p> <p>Speaker’s Intention: ¿your response¿</p> <p>Assumptions about the conversation: ¿your response¿</p> <p>Implicit Information: ¿your response¿</p>
Part 4: Examples for ICL	<p>INPUT:</p> <p>...</p> <p>Persuadee: They are hungry and injured and also short.</p> <p>Persuader: I’m so sorry, what a terrible thing.</p> <p>...</p> <p>Output:</p> <p>...</p> <p>Speaker’s Intention: The Persuadee provides additional details about their child’s situation, emphasizing the child’s needs.</p> <p>Assumptions about the conversation: The Persuadee assumes that sharing these specific details will elicit a stronger empathetic response from the Persuader.</p> <p>Implicit Information: The Persuadee seeks empathy and understanding from the Persuader regarding their child’s dire circumstances.</p> <p>Speaker’s Intention: The Persuader expresses sympathy and acknowledges the gravity of the Persuadee’s situation.</p> <p>Assumptions about the conversation: The Persuader assumes that offering sympathy and acknowledging the seriousness of the situation is an appropriate response.</p> <p>Implicit Information: The Persuader expresses compassion and understanding toward the Persuadee’s plight.</p> <p>...</p>



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

Figure 3: We present here the label distribution for the emotion recognition and the resisting strategies datasets.

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

Table 8: Hyperparameters used for fine-tuning

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

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

Additional Results

We present the comprehensive results that could not make it into the main paper for lack of space, namely ID and TF results over all datasets with line-plots showing a direct visualization of the same in Figure 5 and zero-shot results in Figure 4. We also highlight model mispredictions in terms of confusion matrices (Figures 6 and 7) and highlight labels where models perform consistently better / worse in Figure 8 and Tables 14 and 15.

Table 9: Example of our prompt for the zero-shot and few-shot experiments on LLMs. We illustrate with an example from the P4G dataset.

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

Table 10: Performance of different models on the **CB (Craigslist Bargain)** dataset for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

Model	Mode	Rationale	5	10	20	50	100	All
bert	ID	-	13.8±4.7	20.2±1.4	27.2±6.8	44.7±2.4	57.2±2.0	66.7±3.6
		INT	13.4±5.5	22.9±0.7	34.6±3.4	50.3±2.5	59.3±1.8	68.4±1.7
		ASM	13.0±5.9	22.3±5.0	30.4±1.7	47.2±1.4	60.4±3.0	66.6±0.7
		IMP	13.6±5.0	23.8±3.1	31.6±6.7	50.9±2.5	60.1±1.9	66.9±0.3
		ALL	16.0±6.4	24.8±4.4	38.8±2.2	51.6±1.2	58.5±1.9	67.0±0.7
	TF	-	33.5±2.1	38.1±3.0	39.6±3.2	44.2±3.3	53.8±1.2	65.7±0.9
		INT	35.3±0.4	41.0±4.0	42.2±1.9	49.4±2.9	56.3±1.2	64.8±3.5
		ASM	35.1±0.8	39.7±2.4	41.7±1.4	48.3±1.8	54.3±2.1	66.8±0.6
		IMP	37.5±1.4	42.4±1.6	42.8±0.5	50.2±4.5	55.0±1.9	66.1±2.9
		ALL	37.1±1.9	44.5±2.9	44.8±0.7	52.4±2.0	57.9±0.6	66.3±1.6
gpt2	ID	-	7.6±6.3	12.1±5.7	20.6±4.4	30.7±6.3	43.0±1.0	60.0±0.9
		INT	5.6±1.7	12.7±5.1	17.6±2.8	36.4±5.4	46.1±0.2	65.6±2.0
		ASM	9.8±5.1	12.6±3.1	14.8±3.7	30.1±0.2	43.5±3.5	65.3±1.3
		IMP	6.3±2.8	11.3±5.1	19.9±5.9	35.5±2.8	48.2±4.3	64.9±1.6
		ALL	10.6±6.1	12.1±5.8	20.2±4.5	34.7±3.5	47.6±2.5	66.0±1.5
	TF	-	26.9±2.4	31.8±1.3	32.7±2.3	38.1±0.6	43.0±2.7	60.4±0.4
		INT	33.7±7.4	38.4±1.7	41.7±3.1	48.9±0.9	53.0±0.4	63.7±3.2
		ASM	25.6±6.4	33.1±1.8	34.9±2.6	46.2±1.9	52.1±1.0	63.4±2.9
		IMP	33.6±7.3	35.8±4.2	39.8±2.7	48.0±4.6	53.8±3.0	64.0±1.1
		ALL	31.3±4.8	37.0±5.0	38.1±3.4	47.4±1.8	53.4±1.8	67.0±2.8
t5-base	ID	-	8.5±3.2	11.7±0.6	11.6±1.3	10.6±3.2	23.1±11.4	70.8±1.8
		INT	7.3±2.5	11.8±1.9	11.5±1.7	10.7±3.4	30.7±1.6	70.6±2.8
		ASM	9.2±2.6	7.9±0.9	11.2±2.3	7.0±0.3	23.4±3.1	69.0±1.8
		IMP	8.0±4.3	7.8±1.6	11.3±1.1	10.8±3.0	29.8±2.8	69.1±2.6
		ALL	9.4±2.5	8.2±2.2	10.1±1.5	10.4±4.0	37.7±3.9	72.2±0.5
	TF	-	34.7±1.8	38.4±2.0	40.0±1.1	46.5±4.4	55.8±1.8	70.1±2.9
		INT	34.9±4.3	38.1±2.2	41.3±3.5	53.1±0.8	55.3±3.3	72.1±0.7
		ASM	33.9±3.0	38.9±0.5	42.5±2.6	50.2±3.0	53.0±3.1	70.3±3.4
		IMP	28.5±2.2	37.8±3.1	39.6±5.3	45.7±0.8	50.7±1.6	70.5±1.3
		ALL	33.2±4.5	39.4±2.0	43.7±2.2	50.3±3.9	54.6±3.7	69.6±1.7

Table 11: Performance of different models on the **P4G (Persuasion for Good)** dataset for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

Model	Mode	Rationale	5	10	20	50	100	ALL
bert	ID	-	9.6±0.2	13.5±3.8	19.8±0.6	29.2±0.7	32.9±1.1	50.6±2.5
		INT	10.4±5.0	17.9±2.9	22.4±3.6	31.5±1.4	34.2±1.8	53.0±1.6
		ASM	6.3±3.2	16.4±1.6	17.2±5.8	32.1±0.5	34.3±1.4	49.4±8.1
		IMP	6.8±4.6	15.1±2.2	22.0±1.9	32.2±0.7	35.5±1.5	52.3±1.7
		ALL	8.4±7.5	15.3±6.6	23.0±1.1	32.9±0.7	36.7±1.0	53.2±1.4
	TF	-	22.7±0.3	23.9±0.9	26.7±1.5	29.5±2.6	32.2±0.4	48.4±1.4
		INT	26.4±1.1	29.0±2.7	32.2±1.0	33.7±0.4	35.6±2.0	49.0±0.6
		ASM	24.4±2.8	26.2±2.0	26.9±1.0	30.0±0.6	33.0±1.6	47.0±3.5
		IMP	22.2±3.3	25.1±2.3	28.0±1.2	32.4±0.6	34.2±1.5	48.2±0.7
		ALL	27.0±0.9	29.0±2.2	31.1±0.7	34.9±2.0	37.5±2.3	50.2±3.5
gpt2	ID	-	4.2±2.5	6.3±4.1	10.0±3.2	16.5±1.1	19.2±1.9	35.7±4.4
		INT	3.7±2.4	6.9±2.7	7.9±3.0	20.6±1.4	26.3±3.3	45.7±1.6
		ASM	2.7±1.1	3.7±1.2	7.3±1.9	16.2±5.3	28.4±5.7	47.7±2.4
		IMP	3.9±2.5	6.4±3.5	8.2±4.5	20.2±5.5	27.0±2.6	50.1±2.6
		ALL	2.1±0.8	6.4±1.9	9.0±3.9	21.8±1.7	29.0±4.5	50.1±1.4
	TF	-	13.5±1.8	16.3±0.5	16.6±2.9	19.0±0.6	21.3±1.3	32.4±3.6
		INT	18.3±2.0	20.7±0.4	24.6±0.6	28.5±2.1	30.2±0.3	46.4±1.8
		ASM	20.4±1.2	21.0±1.1	23.6±1.5	26.6±1.9	29.6±0.9	45.0±2.0
		IMP	18.8±3.6	22.4±1.8	23.9±1.6	29.1±1.7	29.7±2.3	48.4±2.1
		ALL	20.6±2.7	23.6±0.3	25.5±2.2	30.6±0.2	31.5±2.0	47.5±2.0
t5-base	ID	-	10.3±0.9	12.6±2.6	6.5±2.3	8.3±2.6	10.5±1.9	48.8±0.9
		INT	10.4±1.0	9.2±5.6	6.6±0.2	10.0±0.8	12.7±0.7	51.2±1.4
		ASM	11.7±2.1	10.2±4.3	8.7±3.9	6.8±0.8	12.0±3.2	51.1±0.8
		IMP	11.1±1.8	7.7±3.5	7.0±2.7	8.0±4.2	11.7±8.9	51.7±3.0
		ALL	13.4±1.1	10.7±4.6	7.4±3.9	7.7±1.4	22.4±7.5	53.4±2.7
	TF	-	19.2±1.6	22.0±2.0	23.9±1.6	28.4±0.9	32.6±0.9	51.2±2.3
		INT	19.9±3.5	24.1±2.1	25.9±2.6	33.5±2.6	31.4±4.4	51.3±1.6
		ASM	19.6±2.0	24.7±3.9	26.0±1.3	29.1±1.3	32.6±1.6	49.3±0.8
		IMP	21.5±1.5	24.4±0.5	29.1±1.8	30.9±1.0	33.5±3.6	51.4±2.9
		ALL	19.2±2.1	21.3±1.7	28.0±2.8	30.8±3.0	37.5±0.8	53.2±1.2

Table 12: Performance of different models on the **Friends** dataset for the task of ERC for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

Model	Mode	Rationale	5	10	20	50	100	All
bert	ID	-	13.4±2.1	15.0±2.1	17.5±3.7	31.2±0.8	33.9±0.7	40.9±0.9
		INT	13.2±1.2	19.2±2.6	26.9±5.8	34.5±3.0	39.9±1.8	45.3±0.8
		ASM	11.5±5.0	16.4±3.4	18.6±3.8	30.2±0.8	35.4±1.3	44.6±0.1
		IMP	12.5±4.9	13.2±3.2	22.5±3.9	32.1±1.6	36.0±1.0	44.7±1.7
		ALL	5.8±3.7	16.9±3.4	23.8±5.3	33.6±2.0	37.7±1.0	46.2±1.3
	TF	-	22.3±1.1	24.7±0.8	26.3±2.1	29.2±2.0	31.6±1.6	41.0±1.3
		INT	24.2±2.0	25.0±2.4	28.3±1.5	30.6±1.0	32.6±1.1	44.9±0.4
		ASM	23.2±3.0	23.9±2.4	24.9±2.7	27.3±1.4	30.8±1.0	40.9±0.8
		IMP	21.4±1.2	24.2±1.5	25.1±1.6	28.1±0.9	31.5±1.5	45.0±0.6
		ALL	25.3±2.2	24.3±1.9	27.6±1.2	30.2±1.3	33.1±1.0	46.1±1.8
gpt2	ID	-	7.7±0.9	9.9±0.9	11.2±0.2	12.2±1.0	17.1±1.1	26.5±0.8
		INT	7.3±1.8	9.5±0.3	10.0±1.5	23.6±2.1	28.4±3.2	44.5±1.0
		ASM	5.7±0.8	7.6±0.5	10.0±1.4	14.0±1.3	20.6±3.4	43.4±1.2
		IMP	7.9±2.4	9.0±1.1	10.1±0.9	15.2±1.7	24.0±1.4	43.3±1.9
		ALL	7.9±1.1	8.8±0.4	10.6±3.6	18.5±1.3	27.1±1.6	45.5±0.8
	TF	-	14.2±1.2	14.6±0.2	14.9±0.9	15.9±1.0	17.9±1.4	26.5±1.3
		INT	21.5±2.6	22.0±1.1	27.2±1.5	27.9±0.8	30.8±1.5	43.7±1.7
		ASM	14.5±3.1	16.4±3.8	18.7±0.9	20.6±1.6	26.3±1.8	40.7±0.7
		IMP	16.9±2.3	16.9±3.3	20.6±1.6	23.2±1.5	27.7±2.5	42.6±1.1
		ALL	19.9±3.1	22.5±1.5	26.5±0.9	27.5±1.8	31.0±2.9	45.4±1.1
t5-base	ID	-	4.8±4.3	11.5±0.3	12.3±1.4	16.2±3.8	25.5±0.4	39.8±3.4
		INT	11.6±5.4	20.1±1.5	26.5±2.7	24.5±2.5	28.3±2.3	44.8±2.6
		ASM	11.3±1.5	13.4±1.3	18.5±2.6	20.0±2.3	23.2±2.8	39.8±0.6
		IMP	11.1±0.3	15.4±4.0	19.8±2.8	22.7±3.1	25.4±5.1	44.1±3.3
		ALL	16.9±2.5	20.0±1.1	28.0±2.1	26.5±1.2	31.3±1.8	43.8±3.1
	TF	-	19.0±0.5	20.4±1.3	21.4±1.7	26.1±2.5	31.2±1.3	40.3±2.9
		INT	24.5±2.4	26.1±2.7	27.8±2.6	29.9±1.2	30.9±1.3	42.6±2.9
		ASM	19.7±2.0	22.6±2.4	23.0±1.0	26.2±0.9	29.2±1.3	44.6±2.3
		IMP	20.6±1.4	22.8±1.0	24.7±1.2	28.2±1.3	30.2±1.7	47.2±0.4
		ALL	24.8±2.3	25.0±1.1	28.0±1.5	30.0±0.9	30.7±0.7	47.4±0.7

Table 13: Performance of different models on the **IEMOCAP** dataset for the task of ERC for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by “All”). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker’s intention (INT), utterance with the hearer’s assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

Model	Mode	Rationale	5	10	20	50	100	all
bert	ID	-	13.7±7.2	16.1±3.1	24.3±2.7	33.0±1.4	36.1±1.1	40.7±1.5
		INT	11.6±4.6	14.8±0.7	23.2±1.2	35.0±1.5	38.3±1.5	42.6±1.3
		ASM	10.8±5.2	19.6±2.7	22.0±1.4	32.8±1.5	35.8±4.3	41.0±1.8
		IMP	13.3±1.7	14.4±5.6	25.2±1.6	32.2±3.4	36.3±2.7	42.0±1.2
		ALL	12.1±5.4	15.7±2.8	25.0±1.4	35.5±2.6	37.6±1.5	40.4±1.0
	TF	-	23.6±1.9	23.8±3.4	24.0±2.4	27.1±1.0	29.5±0.4	37.1±0.8
		INT	22.8±2.2	23.6±1.8	24.3±1.3	29.0±2.4	30.4±0.9	43.4±1.5
		ASM	23.8±1.0	24.2±0.5	24.4±1.0	26.9±2.5	32.5±4.0	39.4±2.4
		IMP	25.0±1.0	24.6±1.7	25.9±1.3	27.0±1.4	29.9±0.3	42.1±0.9
		ALL	25.4±0.4	25.0±1.8	25.6±0.7	28.3±0.5	30.6±1.3	40.7±5.3
gpt2	ID	-	5.0±4.2	6.0±4.7	10.6±2.2	17.1±2.2	23.4±3.0	35.3±2.4
		INT	5.0±3.3	8.8±2.1	9.1±1.7	16.8±0.3	27.5±1.1	42.5±2.4
		ASM	6.2±1.7	8.5±2.9	9.7±1.9	16.4±1.7	25.1±2.3	39.3±3.2
		IMP	5.4±1.5	7.6±1.4	9.6±0.7	15.3±3.1	24.9±3.4	39.9±0.9
		ALL	5.6±3.2	8.3±2.2	9.0±1.6	15.2±0.5	24.3±1.8	39.7±1.8
	TF	-	17.0±1.1	17.0±0.8	19.6±0.9	23.1±2.0	26.4±0.7	36.0±0.8
		INT	20.3±1.5	20.6±1.0	22.5±1.4	24.8±1.7	28.1±0.1	41.0±3.4
		ASM	19.3±0.0	20.8±1.1	22.5±1.2	25.8±0.5	27.3±1.7	40.0±0.4
		IMP	20.2±1.4	20.5±2.4	21.4±0.3	24.8±1.0	27.5±1.1	40.1±2.8
		ALL	19.9±1.8	22.1±0.7	22.9±1.2	25.5±1.2	27.1±1.6	41.9±1.4
t5-base	ID	-	5.3±4.6	8.0±4.0	10.0±2.1	21.1±2.6	26.9±0.6	42.8±1.7
		INT	7.1±0.2	8.7±4.0	15.1±0.9	18.6±1.8	26.3±3.1	45.0±0.7
		ASM	8.9±2.2	8.1±0.8	10.3±5.6	20.3±1.1	28.6±0.2	43.1±0.6
		IMP	6.3±1.3	13.5±2.1	18.3±2.7	22.8±1.7	28.6±1.8	42.0±0.8
		ALL	6.5±3.8	12.9±3.2	18.4±1.7	22.0±2.0	24.8±0.8	44.2±1.2
	TF	-	19.8±0.7	20.6±0.2	20.6±1.0	24.6±2.1	27.8±1.0	41.7±1.0
		INT	22.3±0.7	22.5±0.9	22.3±0.6	27.6±0.6	31.6±1.7	43.9±0.5
		ASM	21.0±0.6	21.3±1.2	21.6±1.1	25.4±1.2	28.2±1.0	43.5±0.6
		IMP	22.4±1.0	21.9±0.5	22.5±1.3	25.4±1.1	27.9±3.7	40.5±2.6
		ALL	23.1±0.5	23.2±0.3	23.2±0.5	27.1±1.9	29.0±1.0	43.9±1.6

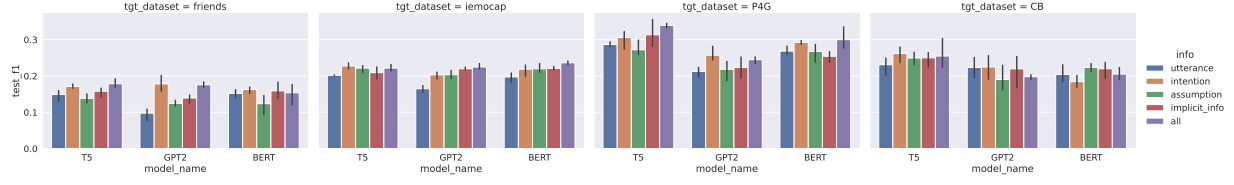


Figure 4: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets in a zero-shot transfer setting, where models trained for the similar task on a given source domain was then applied to the new target domain (e.g. P4G \rightarrow CB and CB \rightarrow P4G for RES and friends \rightarrow iemocap and iemocap \rightarrow friends for ERC.)

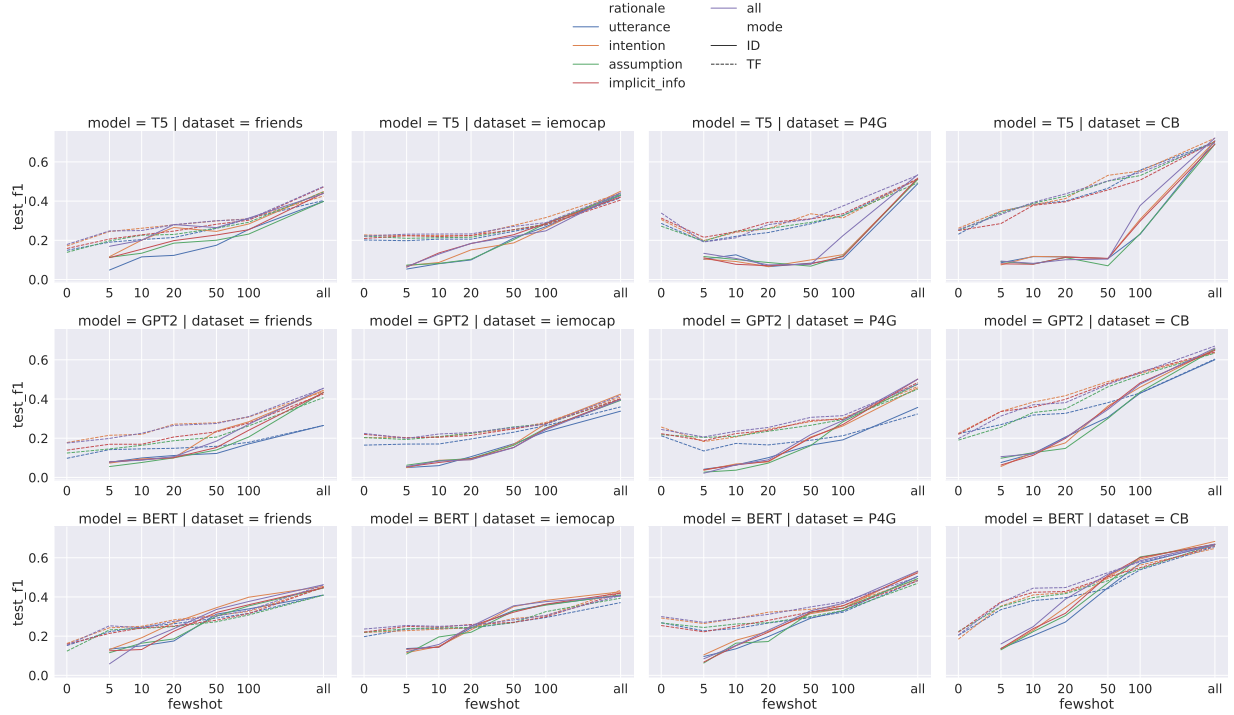
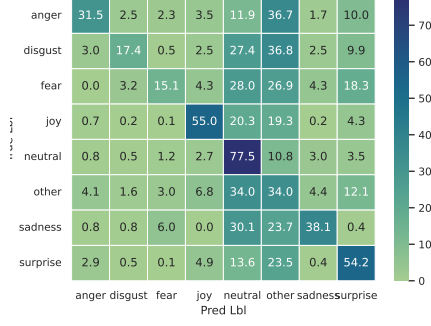
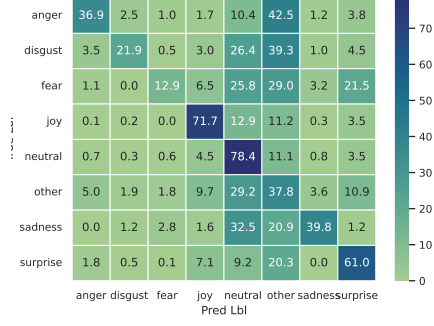


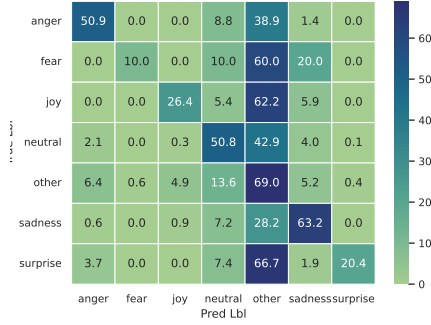
Figure 5: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets for different few-shot examples for all rationales. The solid and dashed lines correspond to the indomain (ID) and transfer (TF) case respectively.



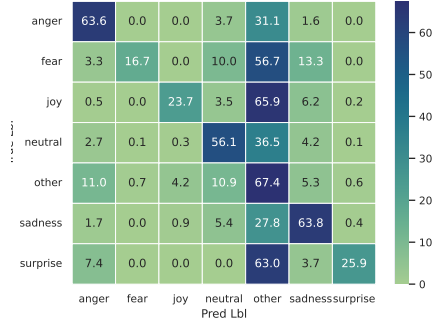
(a) friends with UTT (BERT)



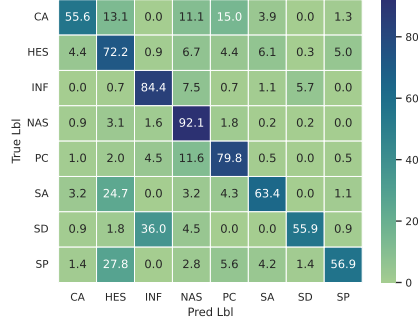
(b) friends with ALL (BERT)



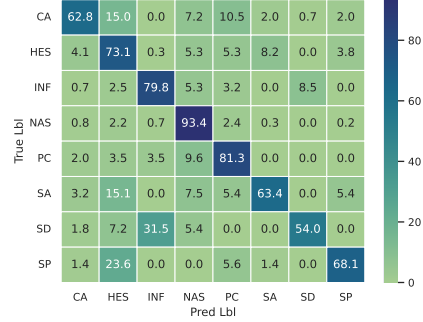
(c) iemocap with UTT (T5)



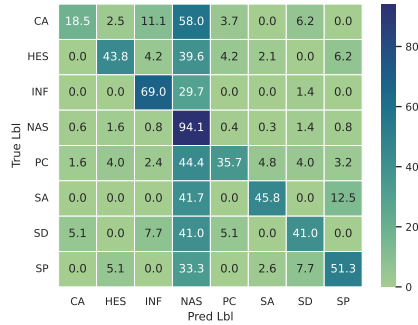
(d) iemocap with INT (T5)



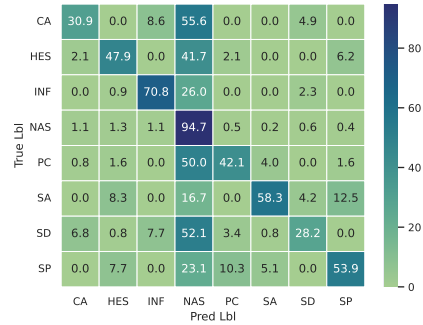
(e) CB with UTT (T5)



(f) CB with ALL (T5)

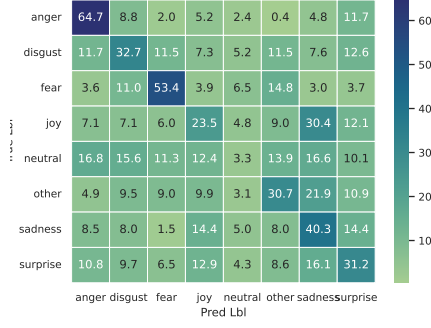


(g) P4G with UTT (T5)

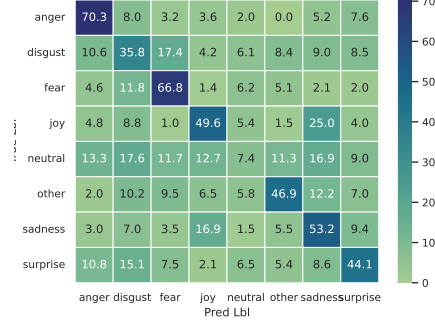


(h) P4G with ALL (T5)

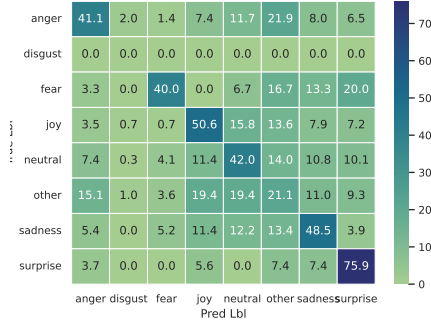
Figure 6: We present here the confusion matrices of the best performing pair of models and rationales in the in-domain setting for the 4 datasets and the corresponding model in absence of any rationale (UTT) in the in-domain setting (ID)



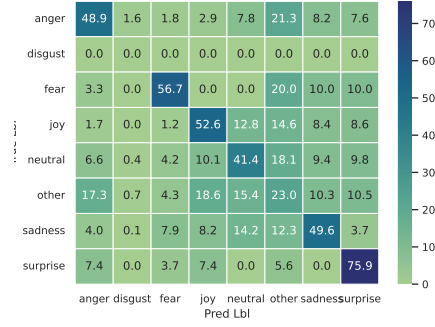
(a) friends with UTT (T5)



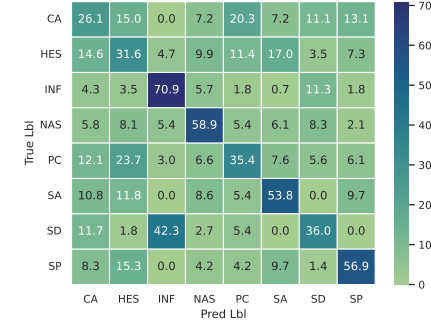
(b) friends with ALL (T5)



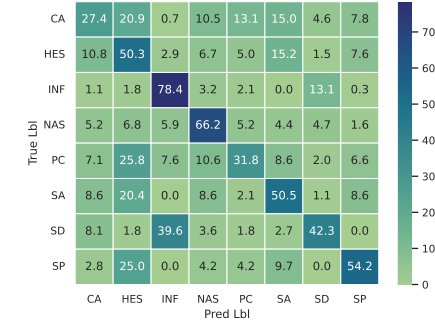
(c) iemocap with UTT (BERT)



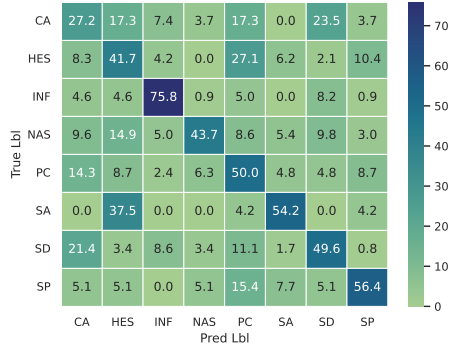
(d) iemocap with INT (BERT)



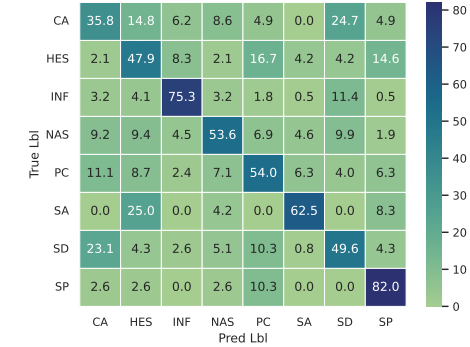
(e) CB with UTT (BERT)



(f) CB with ALL (BERT)



(g) P4G with UTT (BERT)



(h) P4G with INT (BERT)

Figure 7: We present here the confusion matrices of the best performing pair of models and rationales in the transfer setting at k=20-shot case for the 4 datasets and the corresponding model in absence of any rationale (UTT).

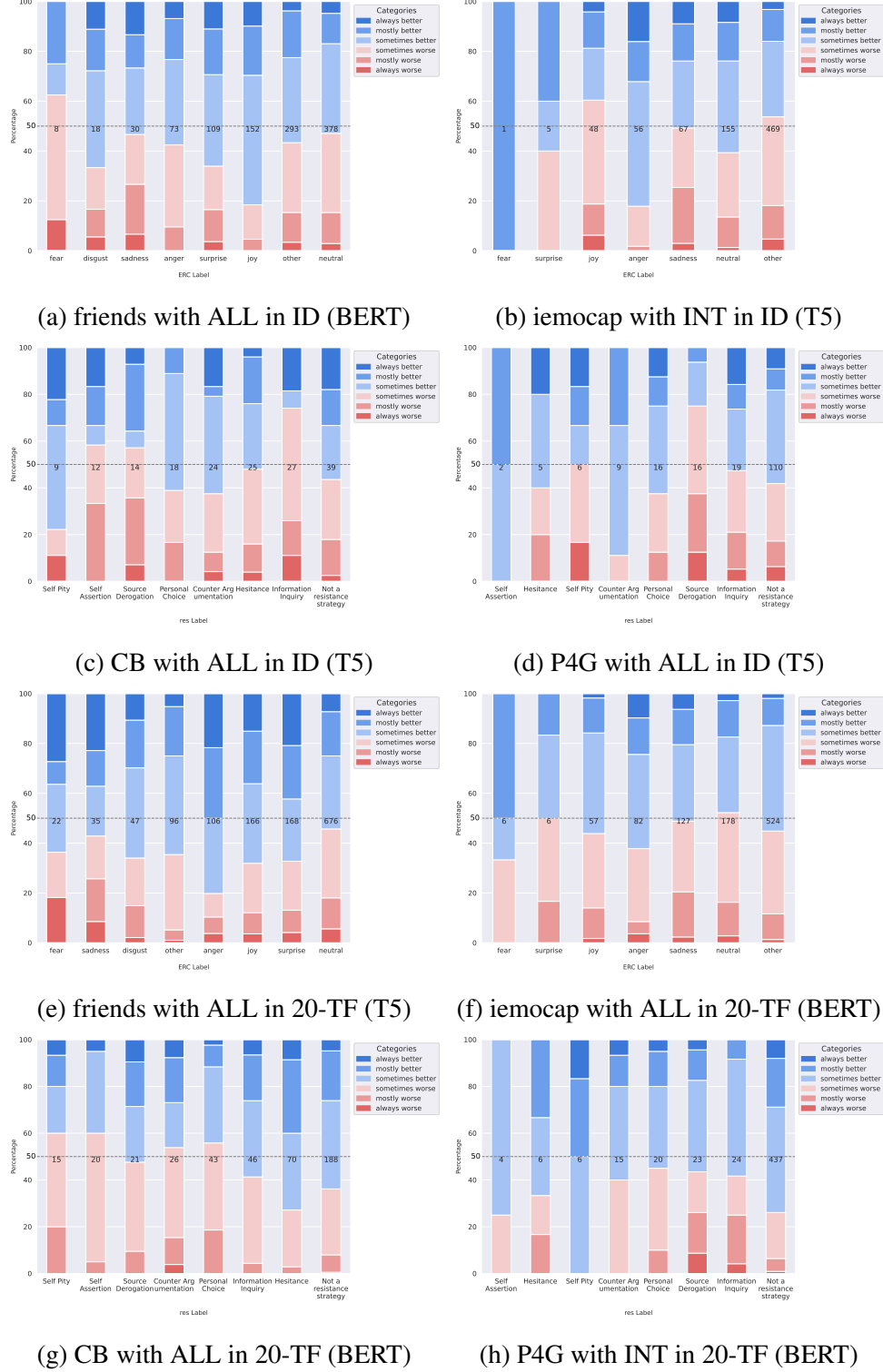


Figure 8: We present here the stacked bar plots that showcases the relative percentage of times a given label was predicted correctly by the best-performing model when augmented with a particular rationale as opposed to the baseline for different datasets. The labels are arranged in increasing order of frequency, with the number inside each bar indicating the frequency of the label.

Table 14: Analysis of dialogue utterances with corresponding contextual information and labels when rationales (RAT) are always better.

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

Table 15: Analysis of dialogue utterances with corresponding contextual information and labels when rationales are always worse

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