University of California
Santa Barbara

Understanding and Improving Language Models Through a Data-Centric Lens

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Alon Albalak

Committee in charge:

Professor William Yang Wang, Co-Chair
Professor Xifeng Yan, Co-Chair
Professor Lise Getoor

June 2024

The Dissertation of Alon Albalak is approved.

Professor Lise Getoor

Professor Xifeng Yan, Co-Chair

Professor William Yang Wang, Co-Chair

March 2024

Understanding and Improving Language Models Through a Data-Centric Lens

by

Alon Albalak

This dissertation is dedicated to my family, who planted the seeds of knowledge and curiosity in my mind, and to my wife who has walked every step of this journey with me.

Acknowledgements

My time in Santa Barbara has been an incredible experience. Going for runs along the bluffs in between studying and research is a once-in-a-lifetime opportunity that I will remember fondly. The beauty of the mountains and ocean are only made more special by the remarkable people that I've gotten to know along the way.

First and foremost, I extend my profound gratitude to my advisors William Yang Wang and Xifeng Yan, who took a chance on me despite my limited research background. William has been an incredible source of ideas, guidance, and support, all while allowing me to forge my own path. Without his open-mindedness and encouragement, my Ph.D. journey would have been much less successful and enjoyable. I also want to express my deep gratitude to Xifeng, who provided invaluable support and infinite patience with me while I was learning, and I am truly grateful for all the wisdom and guidance he has provided me. I would also like to thank Lise Getoor for introducing me to new and exciting areas of research, and for always providing me with thought provoking questions.

A special appreciation is reserved for Ambuj Singh and Tim Robinson, for giving me the opportunity to join UCSB in the IGERT program. It was through the freedom they provided me that I was able to explore exciting research areas in machine learning and natural language processing. I also need to extend a heartfelt thanks to Colin Raffel, who has been an incredible mentor and who's writing style I strive to one day emulate. I am also grateful to my lab mates at UCSB, peers and mentors across academic institutions, as well as in industry: Liangming Pan, Yi-Lin Tuan, Xinyi Wang, Sharon Levy, Wenda Xu, Michael Saxon, Bairu Hou, Shiyu Chang, Connor Pryor, Charles Dickens, Eriq Augustine, Yanai Elazar, Michael Xie, Shayne Longpre, Nathan Lambert, Niklas Muennighoff, Tatsunori Hashimoto, Rohit Jain, Sharon Huffner, Dan Iter, and Mike Ross for the inspirational discussions, support, and guidance.

Curriculum Vitæ

Alon Albalak

Education

2018-2024	Ph.D. in Computer Science **University of California, Santa Barbara**
2016-2018	B.S. in Mathematics **Wayne State University**

Publications and Preprints

[1] **Alon Albalak**, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, William Yang Wang. *"A Survey on Data Selection for Language Models"*. Preprint 2024.

[2] Shayne Longpre, Stella Biderman, **Alon Albalak**, Gabriel Ilharco, Sayash Kapoor, Kevin Klyman, Kyle Lo, Maribeth Rauh, Nay San, Hailey Schoelkopf, Aviya Skowron, Bertie Vidgen, Laura Weidinger, Arvind Narayanan, Victor Sanh, David Adelani, Percy Liang, Rishi Bommasani, Peter Henderson, Sasha Luccioni, Yacine Jernite, Luca Soldaini. *"The Foundation Model Development Cheatsheet"*. Preprint 2024.

[3] **Alon Albalak**, Colin Raffel, William Yang Wang. *"Improving Few-Shot Generalization by Exploring and Exploiting Auxiliary Data"*. Conference on Neural Information Processing Systems, Main Conference (NeurIPS 2023).

[4] **Alon Albalak**, Liangming Pan, Colin Raffel, William Yang Wang. *"Efficient Online Data Mixing For Language Model Pre-Training"*. Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models (Ro-FoMo 2023).

[5] Bo Peng, Eric Alcaide, Quentin Anthony, **Alon Albalak**, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, Rui-Jie Zhu. *"RWKV: Reinventing RNNs for the Transformer Era"*. roceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2023).

[6] Liangming Pan, **Alon Albalak**, Xinyi Wang, William Yang Wang. *"Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning"*. Findings of the Association for Computational Linguistics (EMNLP 2023).

[7] Yi-Lin Tuan, **Alon Albalak**, Wenda Xu, Michael Saxon, Connor Pryor, Lise Getoor, William Yang Wang. *"CausalDialogue: Modeling Utterance-level Causality in Conversations"*. Findings of the Association for Computational Linguistics (ACL 2023).

[8] **Alon Albalak**, Sharon Levy, William Yang Wang. "Addressing Issues of Cross-Linguality in Open-Retrieval Question Answering Systems For Emergent Domains". In Proceedings of the 2023 Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL 2023).

[9] Connor Pryor, Charles Dickens, Eriq Augustine, **Alon Albalak**, William Wang, L. Getoor. *"NeuPSL: Neural Probabilistic Soft Logic"*. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI 2023).

[10] **Alon Albalak**, Yi-Lin Tuan, Pegah Jandaghi, Connor Pryor, Luke Yoffe, Deepak Ramachandran, Lise Getoor, Jay Pujara, William Yang Wang. *"FETA: A Benchmark for Few-Sample Task Transfer in Open-Domain Dialogue"*. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022).

[11] **Alon Albalak**, Akshat Shrivastava, Chinnadhurai Sankar, Adithya Sagar, Mike Ross. *"An Exploration of Methods for Zero-shot Transfer in Small Language Models"*. Efficient Natural Language and Speech Processing Workshop (ENLSP 2022).

[12] Charles Dickens, Connor Pryor, Eriq Augustine, **Alon Albalak**, Lise Getoor. *"Efficient Learning Losses for Deep Hinge-Loss Markov Random Fields"*. Workshop on Tractable Probabilistic Modeling (TPM 2022).

[13] Zekun Li, Hong Wang, **Alon Albalak**, Yingrui Yang, Jing Qian, Shiyang Li, Xifeng Yan. *"Making Something out of Nothing: Building Robust Task-oriented Dialogue Systems from Scratch"*. Alexa Prize Taskbot Challenge 2022.

[14] **Alon Albalak**, Varun Embar, Yi-Lin Tuan, Lise Getoor, William Yang Wang. *"D-REX: Dialogue Relation Extraction with Explanations"*. NLP for Conversational AI Workshop (ConvAI 2022).

[15] Rohit Jain, Devin H. Redmond, Richard B. Sutton, **Alon Albalak**, Sharon Huffner. *"Systems and methods for determining and using semantic relatedness to classify segments of text"*. US Patent 11914963.

[16] Michael Saxon, Sharon Levy, **Alon Albalak**, Xinyi Wang, William Yang Wang. *"Modeling Disclosive Transparency in NLP Application Descriptions"*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021).

Experience

- **Meta**, 06/22 – 09/22.
  Research Science Intern

  - Directed and executed on 2 projects in collaboration with researchers across the company
  - Explored data-efficiency through the use of multi-task learning and various prompting methods for small language models
  - Explored the use of parameter-efficient methods for zero-shot generalization
  - **Resulting Publication**: "An Exploration of Methods for Zero-shot Transfer in Small Language Models"
- **Theta Lake**, 06/19 – 09/20
  Research Associate

  - Built classifiers for automated risk detection in regulated industries through the use of natural language processing and other machine learning techniques
  - Took multiple projects from inception to production, and developed 2 patent pending methods along the way
  - **Resulting Patent**: US Patent 11914963 [15]

Awards and Honors

- NeurIPS Scholar Award, 2023
- Integrative Graduate Education and Research Traineeship (IGERT) Fellow, 2018
- Academic Excellence Fellowship, 2018
- Chia Kuei Tsao Award, For outstanding academic achievement in the undergraduate mathematics program, 2018

Service & Outreach

- Workshop Organizer, NLP for conversational AI, ACL 2023-2024
- Social Organizer, Mindfulness meditation in a time of NLP hyperactivity, ACL 2023
- Workshop Organizer, Transfer learning for NLP: Insights and advances on positive and negative transfer, NeurIPS 2022
- Program Committee, ACL, NAACL, EMNLP, AAAI, 2022-2024

Abstract

Understanding and Improving Language Models Through a Data-Centric Lens

by

Alon Albalak

Training data has played a major role in the rise of large deep learning models. In particular, the scale and diversity of training data has led to incredible new capabilities in large language models. However, despite the success of such models, a notable gap persists in understanding the important role that data plays in their performance, and how to use this understanding to further improve models. In this work, we advocate for, and demonstrate the effectiveness of, data-centric AI.

In the first part of this dissertation, we aim to better understand language models through their data. First, we design a relation extraction system that outputs human-interpretable intermediate outputs, allowing us to better understand why the system makes its predictions. Next, we delve into the intricate relationship between data and models by studying zero-shot and few-shot transfer learning settings, giving us insights into the interactions that training data has on model performance across diverse tasks.

Based on the lessons from the first part of this dissertation, we next aim to improve the data used to train models. We first demonstrate that data selection can be formulated as a multi-armed bandit problem, where the goal is to optimize a model's training data. We apply the multi-armed bandit formulation first to the few-shot fine-tuning setting, and then to language model pretraining, designing algorithms and rewards that are unique for each problem setting. Finally, we show that for cross-lingual question answering, data augmentation is a strong approach to improving the diversity of training data, leading to improved performance.

Overall, this work aims to improve our understanding of how deep learning models work, using data as the viewpoint. Further, we take this understanding and use it to develop data-efficient and performant models. We conclude the dissertation with discussions of future research in data-centric AI and propose avenues for extending these concepts into new research directions.

Contents

Chapter 1

Introduction

1.1 Motivation

1.1.1 A brief history of NLP progress

The field of natural language processing (NLP) has undergone multiple paradigm shifts since it's inception, improving the ability of computers to understand and generate natural language over time. In the early days of NLP, rule-based systems such as SHRDLU [17] and ELIZA [18] operated using a set of predefined linguistic rules and pattern-matching to process inputs and formulate outputs. For example, SHRDLU followed strict rules and has a restricted vocabulary that allowed it to interact with a synthetic block world, and ELIZA simulated a Rogerian psychotherapist with the use of only a 20-word vocabulary. The shift towards statistical NLP marked a significant paradigm shift. Rather than relying on hand-written rules, researchers utilized newly introduced machine learning methods that relied on large quantities of compute to create statistical models that learned patterns and the structure of language from large corpora of text. In particular, IBM developed six "alignment models" [19, 20, 21], with the last version using a hidden markov

model and the large multilingual corpora produced by the European and Canadian parliaments. As computational power continued to increase, and data became more abundant, neural methods emerged as a new paradigm for NLP. Model architectures including Word2Vec [22], LSTM [23] and the Transformer [24] saw success by learning distributed representations of characters, sub-words, words, sentences, and even entire documents by training unsupervised or self-supervised on large corpora of text.

1.1.2 Progress has been driven by scaling

Scaling compute, model sizes, and dataset sizes has led to incredible gains in many areas of NLP and more broadly in machine learning. Each of these paradigm shifts has, in part, been enabled due to improved efficiencies in computation, roughly following Moore's law. However, Gordon Moore and others have suggested that Moore's law either has already ended or will end soon [25, 26, 27], suggesting that we cannot rely on efficiency gains in computation for much longer. In addition to efficiency gains, these paradigm shifts have also been driven by ever-increasing quantities of training data. However, even data cannot be scaled infinitely. More recently, increasing the number of parameters in neural models has been demonstrated to predictably improve performance [28, 29] and sample efficiency [30, 31], as well as leading to unexpected capabilities [32]. However, even model sizes have a limit, and have recently plateaued, with most state-of-the-art large language models in the hundreds of billions of parameters [33], and a very limited number of models reaching beyond 1 trillion parameters [34].

1.1.3 Scaling is not perpetually sustainable

We are arguably in the midst of another, smaller, paradigm shift. At the moment, many researchers are recognizing that scaling model sizes, dataset sizes, and compute

indefinitely isn't a feasible long term approach. Recent works on scaling laws [28, 29] show that performance follows a power-law relationship with each of the model size, dataset size, and total compute. This suggests that the improvements made by scaling parameters, data, and compute are diminishing and adding additional complexity will be a very costly endeavor for marginal gains. In addition, perpetually increasing scale does not benefit many real-world problems, where data collection is difficult or expensive (e.g. medical domain). In these cases, scaling the dataset is simply not an option, and using a small dataset with a very large model can lead to overfitting to the biases present in training data. Finally, as models grow larger and more complex, understanding their internal mechanisms and interpreting their predictions becomes increasingly more challenging. This lack of interpretability has raised concerns about fairness and trust in AI systems [35, 36, 16].

1.1.4 Continued progress with data-centric AI and open-science

Given the concerns raised above, how can we, as a field, continue to make sustainable progress? In this dissertation (and in previous work [1]), I advocate for, and demonstrate, a direction of study orthogonal to scaling which can lead to continued improvement, efficiently. The alternative paradigm that we push for is a deeper understanding of the role that data plays in AI systems, sometimes referred to as data-centric AI. Additionally, an improved understanding of models that is kept behind closed doors will benefit very few and does not encourage progress, so in this dissertation we also advocate for the adoption of open-science principles.

Understanding what makes data more or less helpful for performing a target task and what makes for "high-quality" pretraining data can help us to further improve training datasets by removing detrimental data, or only including data that will be beneficial. Understanding how the characteristics of data (e.g. distribution, diversity,

quantity) impact downstream performance can also help us to optimize datasets for specific purposes, reducing the reliance on massive, general purpose models and datasets. By understanding which aspects of the data are most important, we can reduce the quantity of data required, improving the efficiency of training. Discovering and understanding biases present in a model can assist us to address and mitigate the underlying issue, biases in the training data. Similarly, discovering unwanted behaviors, such as to adversarial examples, can highlight areas of the training data distribution that need to be improved upon to ensure that a model displays robust behavior under all situations.

To further improve our understanding of models, it is also important to design systems that are interpretable to humans. This can allow for a virtuous cycle, where the model is understandable and, because it is understandable, failure cases that arise are interpretable and can be addressed through an improvement to the training data. Another area where an improved understanding of data is particularly impactful is in low-data regimes, where collecting and annotating data is too expensive (e.g requires expert knowledge) or infeasible due to privacy or other concerns. In our proposed paradigm of understanding the relation between data and models, we can continue to improve model performance while reducing our reliance on solely scaling compute.

Research on large scale models and datasets has mostly been developed behind closed doors at well-funded companies, but open-science can propel our progress forward while simultaneously improving our understanding of models. In this dissertation, we also advocate for the open development and dissemination of research in the spirit of open-science, characterized by transparent, collaborative, responsible, and accessible research practices. In this dissertation, we adhere to open-science principles, and promote collaboration, which can enable the pooling of diverse expertise, ensuring the inclusion of many viewpoints. Specifically, we demonstrate that collaboration leads to the development of new benchmarks, which serve as foundational resources for training and evaluating

models. The accessability of openly available resources, including datasets and models, have catalyzed advancements throughout all areas of AI, exemplifying the importance and effect that collaboration has in driving progress. We open-source all models and datasets in this dissertation, further encouraging reproducibility, which can be scrutinized and utilized by future researchers to further improve our understanding of models. Open-science encourages an environment that is conducive to innovation, where ideas and methods can be shared and iteratively refined by a wide variety of scientists, researchers, and practitioners. In recent work [1] we proposed three concrete directions of work that can help advance future open research: (1) metrics that directly evaluate data, (2) data-centric benchmarks and challenges, and (3) open-sourced tools. By adopting and advocating for open-science, we hope to encourage collaborative, innovative, responsible, and reproducible research.

1.2 Overview

In this dissertation, I demonstrate how we can *better understand the relationship between models and data, then use that understanding to further improve the generalization and data efficiency of models.* My research uses natural language as the domain of interest, and the studies performed here analyze large language models (LLM) in particular. Modern LLMs have been scaled up to extreme sizes, making research inefficient, and sometimes impossible, for smaller research groups. To combat this, much of the research contained here is focused on efficiency, with a specific focus on settings that have limited resources (e.g. limited data or low-resource languages).

In the following chapters, we present a series of methods and studies on understanding the relation between models and data, and how to use that understanding to further improve models.

1.2.1 Understanding models through data

Part I consists of three studies that demonstrate methods for improving our understanding of models. In Chapter 2, we start by introducing D-REX, a method for improving the interpretability and performance of relation extraction models by introducing intermediate steps into the system's prediction process. In doing so, we demonstrate one method that not only makes models more understandable to humans, but also improves in performance over black-box methods. Next, in Chapter 3, we explore the interactions between training datasets and model behaviors on unseen tasks in the zero-shot learning setting. To isolate a model's robustness to out-of-distribution data, we study the performance of models trained using multi-task learning on both in-domain and out-of-domain datasets. Then, in Chapter 4, we introduce our benchmark for FEw-sample TAsk transfer (FETA) and provide the first large-scale study of intra-dataset task transfer for NLP. Intra-dataset task transfer is the setting where both the source and target dataset are from the same distribution, meaning that we have isolated task transfer from domain adaptation. To study intra-dataset task transfer with FETA, we compare three task transfer algorithms, three commonly used language models, and both single- and multi-source transfer settings.

1.2.2 Improving models through data

Part II contains three methods for improving model performance, using some of the lessons learned in Part I. First, in Chapter 5, we propose methods for improving the few-shot learning with auxiliary data (FLAD) setting, where the target task has very limited data, but (possibly) related auxiliary datasets are available. We directly connect the FLAD setting to multi-armed bandits and design algorithms that focus on the exploration-exploitation tradeoff. We also design multiple reward functions that are very efficient to compute, leading to performant algorithms. In Chapter 6 we change our focus

to improving the efficiency and performance of language model pretraining. We do so through the problem of data mixing, where the goal is to determine the proportion of data from each of the individual training data domains, formulating the mixing problem as a multi-armed bandit. In this setting, we design a reward function that aims to maximize the information gain of training data, and demonstrate how this leads to significant efficiency gains. Then, in Chapter 7, we discuss and demonstrate a problem setting with very limited data in the real world, question answering for emergent domains. Specifically, we build a system for cross-lingual open-retrieval question answering because the language of a new domain of knowledge is not known ahead of time, requiring systems that can robustly find reliable information across languages. We take COVID-19 as an exemplar of an emergent domain, and demonstrate how to build such a system, even with incredibly limited multilingual and cross-lingual data.

We finish the dissertation by summarizing and providing conclusions to our research. Furthermore, we discuss directions of future research that can further improve our understanding of the relation between models and data as well as promising directions of study to further improve training data for models. Finally, we discuss future directions of research that move beyond siloed data research and consider the entire system.

Part I

Understanding Models Through Data

Chapter 2

Making Relation Extraction Models Understandable

While there have been significant advancements achieved through sophisticated deep learning algorithms, the black-box nature of these methods can pose a significant challenge in high-stakes domains where transparency and interpretability are of high importance. One approach to improving interpretability of machine learning models is to have them to produce intermediate steps in their decision making process, where the intermediate steps can be understood by a human. This approach also facilitates the identification and rectification of incorrect predictions, better enabling practitioners to trace the model's errors and further improve the system. In this chapter, we propose an interpretable relation extraction system that utilizes multiple machine learning models producing intermediate results. We show that not only does the system produce interpretable intermediate results, but actually improves in performance over previous black-box methods.

2.1 Introduction

Traditional relation extraction (RE) approaches discover relations that exist between entities within a single sentence. In recent years, several approaches have been proposed which focus on cross-sentence RE, the task of extracting relations between entities that appear in separate sentences [37, 38, 39, 40] as well as cross-sentence RE in dialogues [41, 42, 43, 44, 45]. A crucial step towards performing cross-sentence RE in multi-entity and multi-relation dialogues is to understand the context surrounding relations and entities (e.g., who said what, and to whom). Figure 2.1 shows an example from the DialogRE dataset where a simple BERT-based model (Initial Predicted Relation in Figure 2.1) gets confused by multiple entities and relations existing in the same dialogue [41]. The model predicts the "girl/boyfriend" relation between Speaker 2 and Chandler, however, it is clear from the context that the "girl/boyfriend" relation is referring to a different pair of entities: Speaker 1 and Chandler.

One approach to encourage a model to learn the context surrounding a relation is by requiring the model to generate an explanation along with the relation [46]. Furthermore, requiring the model to output an explanation also improves the interpretability of the model, allowing the model developer to better understand why the model makes incorrect predictions, and what may be causing the error. In addition to the DialogRE dataset, Yu et al. [41] introduces manually annotated *trigger words* which they show play a critical role in dialogue-based RE. They define trigger words as

Speaker 1: Could you please get the key off the back of the door for me.

Speaker 2: Oh yeah! Yeah!

Speaker 1: You tell your friend Chandler that we're definitely broken up this time.

Speaker 2: Okay!

Subject	Object	Initial Predicted Relation	D-REX Predicted Explanation	D-REX Predicted Relation
Speaker 2	Chandler	girl/boyfriend	your friend	friends

Figure 2.1: A sample dialogue between 2 speakers with actual D-REX predictions. The model initially classifies Speaker 2 and chandler, incorrectly, as girl/boyfriend. After predicting the explanation "yourfriend", D-REX correctly re-ranks the relation as friends.

"the smallest span of contiguous text which clearly indicates the existence of the given relation". In the context of RE, these trigger words can be used as potential explanations of the model's decision.

This chapter demonstrates how to extract explanations that clearly indicate a relation while also benefiting an RE model by providing cross-sentence reasoning. Our proposed approach, D-REX, makes use of multiple learning signals to train an explanation extraction model. First, D-REX utilizes trigger words as a partial supervision signal. Additionally, we propose multiple reward functions used with a policy gradient, allowing the model to explore the explanation space and find explanations that benefit the re-ranking model. Including these reward functions allows D-REX to learn meaningful explanations on data with less than 40% supervised triggers.

In order to predict relation- and entity-specific explanations in D-REX, we pose RE as a relation re-ranking task with explanation extraction as an intermediate step and show that this is not possible for a model trained to perform both tasks jointly.

Our contributions are summarized as follows:

- We propose D-REX, **D**ialogue **R**elation **E**xtraction with e**X**planations, a novel system trained by policy gradient and semi-supervision.
- We show that D-REX outperforms a strong baseline in explanation quality, with human evaluators preferring D-REX explanations over 90% of the time.
- We demonstrate that by conditioning on D-REX extracted explanations, relation extraction models can improve by 1.2-4.7%.

2.2 Problem Formulation

We follow the problem formulation of Yu et al. [41]: let $d = (s_1 : u_1, s_2 : u_2, \ldots, s_n : u_n)$ be a dialogue where s_i and u_i denote the speaker ID and the utterance from the i^{th} turn, respectively. Let $\mathcal{E}, \mathcal{R}$ be the set of all entities in the dialogue and the set of all possible relations between entities, respectively. Each dialogue is associated with m relational triples $<s, r, o>$ where $s, o \in \mathcal{E}$ are subject and object entities in the given dialogue and $r \in \mathcal{R}$ is a relation held between the s and o. Each relational triple may or may not be associated with a trigger t. It is important to note that there is no restriction on the number of relations held between an entity pair; however, there is at most one trigger associated with a relational triple. In this chapter, we consider an explanation to be of high quality if it strongly indicates that a relation holds, and for this purpose we consider triggers to be short explanations, though not always optimal in quality.

2.2.1 Relation Extraction (RE)

Given a dialogue d, subject s, and object o, the goal of RE is to predict the relation(s) that hold between s and o. We also consider RE with additional evidence in the form of a trigger or predicted explanation. Formally, this is the same as relation extraction with an additional explanation, ex.

2.2.2 Explanation Extraction (EE)

We formulate EE as a span prediction problem. Given a dialogue d consisting of n tokens T_1 through T_n, and a relational triple $<s, r, o>$, the goal of EE is to predict start and end positions, i, j in the dialogue, such that the explanation $ex = [T_i, T_{i+1}, \ldots, T_j]$ indicates that r holds between s and o.

2.3 Baseline Models

We first introduce approaches for RE and EE based on state-of-the-art language models. We then propose a multitask approach that performs both tasks jointly. Our approaches use $\text{BERT}_{\text{base}}$ [47] and $\text{RoBERTa}_{\text{base}}$ [48] pre-trained models[1], and follow their respective fine-tuning protocols.

For all models, we maintain a single input format, which follows from Yu et al. [41]. Formally, for a dialogue d, subject s, object o, relation r, and explanation ex, the input sequence to all models is [CLS]$\{r/ex$[SEP]$\}s$[SEP]o[SEP]d, where $\{r/ex$[SEP]$\}$ denotes that the relation or explanation may be included depending on the task setting. For RoBERTa models, we use the <s> and </s> tokens rather than [CLS] and [SEP], respectively.

2.3.1 Relation Extraction (RE)

We follow the fine-tuning protocols of Devlin et al. [47] and Liu et al. [48] for BERT and RoBERTa classification models by using the output corresponding to the first token $C \in \mathbb{R}^H$ ([CLS] and <s>, respectively) as a latent representation of the entire input and train a classification matrix $W \in \mathbb{R}^{K\text{x}H}$, where K is the number of relation types and H is the dimension of the output representations from the language model. For each relation r_i, the probability of r_i holding between s and o in d is calculated as $P_i = \text{sigmoid}(CW_i^T)$. We compute the standard cross-entropy loss for each relation as

$$\mathcal{L}_{RE} = -\frac{1}{\text{K}} \sum_{i=1}^{\text{K}} y_i \cdot \log(P_i) + (1 - y_i) \cdot \log(1 - P_i) \tag{2.1}$$

where y_i denotes whether relation i holds.

[1]Pre-trained models obtained from https://github.com/huggingface/transformers [49]

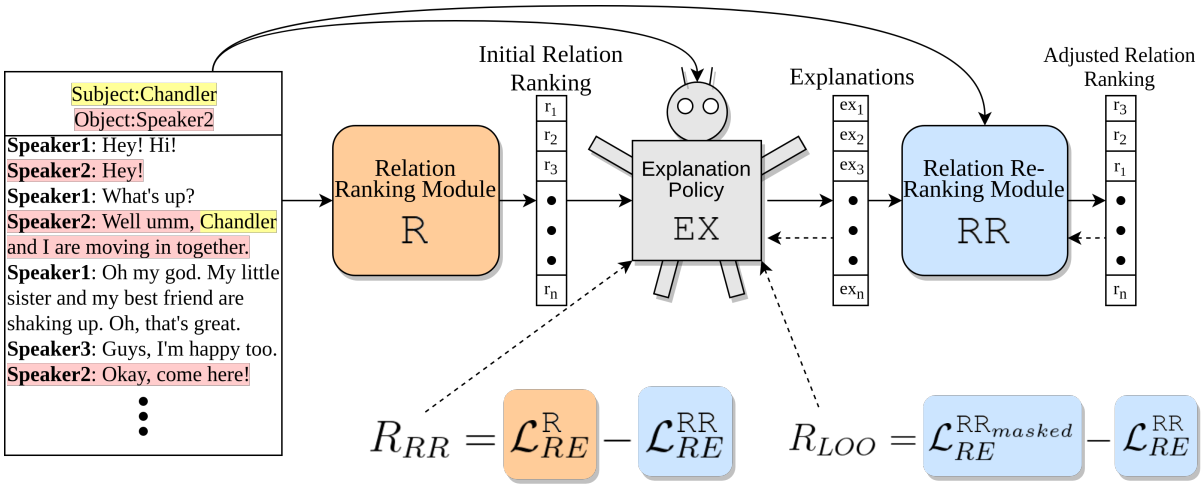

Figure 2.2: Overview of the D-REX system. The relation ***R***anking module ranks relations conditioned only on the subject, object, and the dialogue. The ***EX***planation policy extracts supporting evidence for the ranked relations by conditioning on individual relations in addition to the original input. The relation ***R***e***R***anking module conditions its rankings on supporting evidence from the explanation policy. In this hypothetical example, we see that relation 3 was originally ranked number 3 but had strong supporting evidence and was re-ranked in the number 1 spot. Solid lines represent model inputs/outputs, and dotted lines represent learning signals. Reward functions, $\mathcal{R}_{RR}$ and $\mathcal{R}_{LOO}$, are detailed in equations 2.4 and 2.5, respectively.

2.3.2 Explanation Extraction (EE)

For EE, we use the input described above, with a natural language phrasing of a relation appended to the beginning of the sequence. For example, if r is "per:positive_impression", then we concatenate "person positive impression" to the beginning.

We follow the fine-tuning protocol of Devlin et al. [47] for span prediction. We introduce start and end vectors, $S, E \in \mathbb{R}^H$. If $T_i \in \mathbb{R}^H$ is the final hidden representation of token i, then we compute the probability of token i being the start of the predicted explanation as a dot product with the start vector, followed by a softmax over all words in the dialogue:

$$P^S_{T_i} = \frac{exp(S \cdot T_i)}{\sum_j exp(S \cdot T_j)} \tag{2.2}$$

To predict the end token, we use the same formula and replace the start vector S with the end vector E. To compute the loss, we take the mean of the cross-entropy losses per token for the start and end vectors. Formally, let $|d|$ be the number of tokens in dialogue d, then

$$\begin{aligned}\mathcal{L}_{EX} = -\frac{1}{|d|}\sum_{i}^{|d|} \\ \big(y_i^S \cdot log(P_{T_i}^S) + (1 - y_i^S) \cdot \log(1 - P_{T_i}^S)\big) \\ + \big(y_i^E \cdot log(P_{T_i}^E) + (1 - y_i^E) \cdot \log(1 - P_{T_i}^E)\big)\end{aligned} \tag{2.3}$$

where y_i^S and y_i^E are the start and end labels. Because we want explanations extracted only from the dialogue, if the start or end token with largest log-likelihood occurs within the first l tokens, where l is the length of [CLS]r[SEP]s[SEP]o[SEP], then we consider there to be no predicted explanation.

2.3.3 Joint Relation and Explanation Model

The joint RE and EE model uses the standard input from §2.3. It utilizes a BERT or RoBERTa backbone, and has classification and span prediction layers identical to those in the RE and EE models. Similarly, the loss is computed as the weighted sum of RE and EE losses:

$$\mathcal{L}_{\mathcal{J}} = \alpha\mathcal{L}_{RE} + (1 - \alpha)\mathcal{L}_{EX}$$

where α is an adjustable weight. In practice, we find that $\alpha = 0.5$ works best.

Flaw of the joint model. The disadvantage of the joint model is this: supposing that an entity pair has 2 relations, each explanation should be paired with a single relation. However, by making predictions jointly, there is no guaranteed mapping from predicted

explanations to predicted relations. One method of solving this issue is to predict relations and explanations in separate steps. It is possible to first predict relations and then condition the explanation prediction on each individual relation and conversely. This idea forms the basis for D-REX.

2.4 D-REX System

In this section, we introduce the D-REX system. We begin by introducing the models which make up the system. Next, we present the training and inference algorithms. Finally, we discuss the optimization objectives for each model in the system.

2.4.1 Models

The D-REX framework requires three components: an initial relation ranking model, an explanation model, and a relation re-ranking model, shown in Figure 2.2.

Initial Ranking Model (R). In our algorithm and discussions, we use R to denote the initial ranking model. There are no restrictions on R, it can be any algorithm which ranks relations (e.g., deep neural network, rule-based, etc.) such as [41, 45]. However, if R needs to be trained, it must be done prior to D-REX training; D-REX will not make any updates to R.

In our evaluations, we use the relation extraction model described in §2.3.1. The input to this model is (s,o,d) and the output is a ranking, $R(s, o, d)$.

Explanation Extraction Model (EX). In our algorithm and discussions, we use EX to denote the explanation model. In this chapter we limit our experiments to extractive explanation methods, as opposed to generative explanation methods, however this is not

a limitation of D-REX. The only limitation on the explanation model is that we require it to produce human-interpretable explanations. Thus, it is also possible to use generative models such as GPT-2 [50] or graph-based methods such as [51, 43] with adjustments to the formulation of the reward functions.

In our evaluations, we use the model as described in §2.3.2. The input to *EX* is (r,s,o,d) and the output is an extracted phrase from d, denoted as $EX(r, s, o, d)$.

Relation Re-Ranking Model (*RR*). In our algorithm and discussions, we let *RR* denote the relation re-ranking model. In the D-REX training algorithm, *RR* is updated through gradient-based optimization methods, and must be able to condition its ranking on explanations produced by *EX*. In our experiments, we use the same model architecture as R and include an explanation as additional input to the model. The input to *RR* is (ex,s,o,d) and the output is a relation ranking, denoted as $RR(ex, s, o, d)$.

2.4.2 D-REX Algorithm

The outline of this algorithm is shown in pseudocode in Algorithm 1.

Assuming that we have ranking, explanation, and re-ranking models R, *EX*, *RR*, then given a single datum (s, r, o, t, d), comprised of a subject, relation, object, trigger(may be empty), and dialogue, the D-REX algorithm operates as follows: The ranking model takes as input (s, o, d) and computes the probability of each relation from the predefined relation types. Next, we take the top-k ranked relations, $r_{pred} = R(s, o, d)_{1:k}$, and compute explanations. For $i = 1, ..., k$, explanations are computed as $ex_i = EX(r_{pred_i}, s, o, d)$. Finally, for each predicted explanation, the re-ranking model computes k probabilities for each relation type, using (ex_i, s, o, d) as the input to *RR*. The final probabilities for each relation type are computed as the mean across all k+1 predictions from R and *RR*.

Algorithm 1 The proposed training algorithm for D-REX

Input : Pre-trained ranking, explanation, and re-ranking models: R, EX, RR
k: for number of relations to re-rank
Data: Dataset: $\mathcal{D}$
for *(s, r, o,t,d) in* $\mathcal{D}$ **do**
  Compute ranking loss: $\mathcal{L}^{R}_{RE}(s, o, d)$
  $r_{pred} \leftarrow R(\text{s,o,d})_{1:k}$
  for *i in* r_{pred} **do**
    $ex_i \leftarrow EX(r_{pred_i}, s, o, d)$
    Compute Re-ranking loss: $\mathcal{L}^{RR}_{RE}(ex_i, s, o, d)$; // Equation 2.1
    Compute Re-Ranking Reward: $\mathcal{R}_{RR}$; // Equation 2.4
    Compute Leave-one-out Reward: $\mathcal{R}_{LOO}$; // Equation 2.5
    Compute policy gradient with rewards R_{RR}, R_{LOO} ; // Equation 2.6
  end
  if *t not empty* **then**
    Compute $\mathcal{L}_{EX}$; // Equation 2.3
  end
  Update *EX,RR*parameters with calculated losses
end

2.4.3 Model optimization

We propose multiple optimization objectives to train an *EX* model that extracts explanations meaningful to humans and beneficial to the relation extraction performance while ensuring that *RR* maintains high-quality predictions.

Explanation Model Optimization. We train *EX* with supervision on labeled samples, and a policy gradient for both labeled and unlabeled samples, allowing for semi-supervision. For the policy gradient, we introduce two reward functions: a relation re-ranking reward and a leave-one-out reward.

Re-ranking Reward The purpose of the re-ranking reward is to ensure that *EX* predicts explanations which benefit *RR*. Formally, let $\mathcal{L}^{R}_{RE}(s, o, d)$ be the loss for R, given the subject, object, and dialogue: s, o, d. And let $\mathcal{L}^{RR}_{RE}(ex, s, o, d)$ be the loss of *RR*, given the explanation, subject, object, and dialogue: ex, s, o, d. Then we define the

relation re-ranking reward as:

$$\mathcal{R}_{RR} = \mathcal{L}^{R}_{RE}(s, o, d) - \mathcal{L}^{RR}_{RE}(ex, s, o, d) \tag{2.4}$$

Because R is stationary, EX maximizes this function by minimizing $\mathcal{L}^{RR}_{RE}$. Of course, EX can only minimize $\mathcal{L}^{RR}_{RE}$ through its predicted explanations.

Leave-one-out Reward The purpose of the leave-one-out reward is to direct EX in finding phrases which are essential to correctly classifying the relation between an entity-pair. This reward function is inspired by previous works which make use of the leave-one-out idea for various explanation purposes [52, 53]. We can calculate the leave-one-out reward using either R or RR, and it is calculated by finding the difference between the standard relation extraction loss and the loss when an explanation has been masked. Formally, if d is the original dialogue and ex is the given explanation, let $d_{mask}(ex)$ be the dialogue with ex replaced by mask tokens. Then, the leave-one-out reward is defined as:

$$\mathcal{R}_{LOO} = \mathcal{L}_{RE}(s, o, d_{mask}(ex)) - \mathcal{L}_{RE}(s, o, d) \tag{2.5}$$

Because $\mathcal{L}_{RE}$ is calculated using the same model for both the masked and unmasked loss, EX maximizes this reward function by maximizing the masked loss. Of course, the only interaction that EX has with the masked loss is through the explanation it predicts.

Policy Gradient We view EX as an agent whose action space is the set of all continuous spans from the dialogue. In this view, the agent interacts with the environment by selecting two tokens, a start and end token and receives feedback in the form of the previously discussed reward functions. Let i, j be the start and end indices that the explanation model selects and T_i be the i^{th} token, then $ex = d[i:j] = [T_i, T_{i+1}, \ldots, T_j]$ and the probabilities of i, j being predicted are calculated as $P^{S}_{T_i}$ and $P^{E}_{T_j}$ according to

equation 2.2.

For both reward functions, we use a policy gradient [54] to update the weights of the explanation model and calculate the loss as

$$\mathcal{L}_{EX_{PG}} = -(log(P^S_{T_i}) + log(P^E_{T_j})) * (R_{RR} + R_{LOO}) \tag{2.6}$$

Additionally, while training EX in the D-REX algorithm, we make use of supervision when available. In the case where supervision exists, we calculate an additional loss, $\mathcal{L}_{EX}$, as defined in equation 2.3.

Relation Extraction Re-ranking Model Optimization. While training D-REX we train RR with labeled relations as supervision and use a cross-entropy loss, $\mathcal{L}^{RR}_{RE}$, calculated in the same way as R in Equation 2.1.

2.5 Experimental Evaluation

In this section, we present an evaluation of D-REX in comparison with baselines methods on the relation extraction and explanation extraction tasks.

2.5.1 Experimental settings

For our experiments, we re-implement the BERT_S model from [41] as well as a new version which replaces BERT with RoBERTa. In our work, we refer to these models as R_{BERT} and R_{RoBERTa}. All models are implemented in PyTorch[2] and Transformers[49], trained using the AdamW optimizer [55]. All experiments were repeated five times and we report mean scores along with standard deviations. D-REX models use a top-k of five and

[2]https://pytorch.org/

DialogRE V2			
Dialogues	Relations	Relational Triples (train/dev/test)	Triggers (train/dev/test)
1788	36	6290/1992/1921	2446/830/780

Table 2.1: **Dataset details** for DialogRE. With only 2446 labeled triggers in the training set, D-REX models learn using only a policy gradient and no direct supervision on the remaining 3844 triples.

are initialized from the best performing models with the same backbone. For example, D-REX$_{\text{BERT}}$ uses two copies of R_{BERT} [41] to initialize the ranking and re-ranking models and EX_{BERT} to initialize the explanation model. When training *Joint*, we do not calculate $\mathcal{L}_{EX}$ for relational triples without a labeled trigger.

All models are trained using the AdamW optimizer [56] with a learning rate of 3e-5 and batch sizes of 30. To determine the best learning rate, R and EX models were trained using learning rates in {3e-6, 1e-5, 3e-5, 1e-4}. The best learning rate, 3e-5, was determined by performance on a held out validation dataset. Baseline models (R, EX, and *Joint*) are trained for at most 30 epochs and we use validation-based early stopping to determine which model to test. D-REX models are trained for at most 30 additional epochs with the best model determined based on relation extraction F1 scores computed on validation data. We found the best validation result to always occur within the first 30 epochs. All experiments were repeated five times and we report the mean score along with standard deviation. To train the joint model, we do not calculate $\mathcal{L}_{EX}$ for relational triples which do not have a labeled trigger and we select α from {0.25,0.5,0.75} and set α to 0.5 based on validation performance.

DialogRE Dataset. We evaluate our models on the DialogRE English V2 dataset[3] which contains dialogues from the Friends TV show [41], details of which are in Table 2.1. D-REX models are trained with trigger supervision on less than 40% of the training data, and make no use of dev or test set triggers. The learning signal for the remaining triples comes entirely from our rewards through a policy gradient.

Evaluation Metrics. We adopt separate evaluations for relation and explanation extraction.

First, for relation extraction, we evaluate our models using F1 score, following Yu et al. [41], and additionally calculate the mean reciprocal rank (MRR), which provides further insight into a model's performance. For example, MRR is able to differentiate between a ground truth relation ranked 2nd or 10th, while the F1 score does not. In the dialogRE dataset, multiple relations may hold between a single pair of entities, so we use a variation of MRR which considers all ground truth relations, rather than just the highest-ranked ground truth relation.

For explanation extraction, we focus mainly on manual evaluations, but also propose the Leave-One-Out metric, introduced in section 2.5.4 for an ablation study.

2.5.2 Relation Extraction (RE) Evaluation

In Table 2.2, we compare the baseline RE model R_{BERT} with the methods presented in this study. We also compare with three other methods which use similarly sized language models, but additionally utilize graph neural networks (GNN): GDPNet[43], TUCORE-GCN$_{\text{BERT}}$[45], and SocAoG[44].

First, we see that even though D-REX is designed to introduce human-understandable explanations, it still has modest improvements over R_{BERT}, which focuses on RE, while

[3]Dataset collected from https://dataset.org/dialogre/ for research purposes only

Model	**F1**(σ)	**MRR**(σ)
R_{BERT}	59.2(1.9)	74.8(1.3)
$Joint_{\text{BERT}}$	59.4(1.7)	74.0(0.9)
D-REX$_{\text{BERT}}$	**59.9**(0.5)	**75.4**(0.1)
R_{RoBERTa}	64.2(1.6)	77.9(1.0)
$Joint_{\text{RoBERta}}$	65.2(0.3)	78.3(0.3)
D-REX$_{\text{RoBERTa}}$	**67.2**(0.3)	**79.4**(0.3)
*GDPNet	60.2(1.0)	-
*TUCORE-GCN$_{\text{BERT}}$	65.5(0.4)	-
†SocAoG	**69.1**(0.5)	-

Table 2.2: **Relation extraction results on DialogRE V2**. R models are described in Section 2.3.1, *Joint* models in 2.3.3, and D-REX models in 2.4. R_{BERT} is a replication of BERT$_{\text{S}}$ from Yu et al. [41]. "*" denotes results taken from Lee and Choi [45] and "†" from Qiu et al. [44]

Joint has no significant improvement. Next, we see a five point absolute improvement in F1 from the baseline model when using RoBERTa. The trend from BERT to RoBERTa is similar to results found by Lee and Choi [45], where changing from a BERT$_{\text{base}}$ model to RoBERTa$_{\text{Large}}$(not shown here) improved their model performance significantly. Additionally, we see a 3 point improvement from R to D-REX when using RoBERTa (compared to 0.7 for BERT), which we believe is due to the better performing ranking model, which allows for D-REX to rely more on the input explanations. Finally, we see that by using GNNs, and task-specific dialogue representations, all three GNN-based methods can improve over the general BERT-based methods.

2.5.3 Explanation Extraction (EE) Evaluation

Automatic Evaluation. Although the aim of this study is not trigger prediction, for completeness and reproducibility, we include results on the test set of triggers here. In Table 2.3, we compare our methods for supervised explanation extraction with D-REX. Interestingly, we find that the joint model achieves the lowest F1 score for both the BERT and RoBERTa models. $Joint_{\text{BERT}}$ scores nearly 20 points below its counterpart BERT

model	token F1(σ)	EM(σ)	LOO(σ)
EX_{BERT}	62.1(3.1)	54.1(1.9)	82.2(0.4)
$Joint_{\text{BERT}}$	43(1.3)	38.6(1.4)	89.0(1.0)
D-REX$_{\text{BERT}}$	50.5(1.1)	45.7(1.7)	84.4(1.6)
EX_{RoBERTa}	66.5(2.2)	58.4(2.0)	82.2(0.4)
$Joint_{\text{RoBERTa}}$	49(0.7)	47(0.7)	86.2(0.8)
D-REX$_{\text{RoBERTa}}$	57.2(2.1)	51.6(1.6)	83.9(0.4))

Table 2.3: **Trigger prediction results**. Leave-One-Out metric (LOO) measures how salient a predicted explanation is in determining a relation and is further defined in §2.5.4. Smaller LOO is better.

model, while the $Joint_{\text{RoBERTa}}$ model cuts that difference to just over 15 points below its RoBERTa counterpart. On the other hand, D-REX maintains a token F1 score within 10 points of its counterpart even though it has been trained to generalize beyond the labeled triggers.

Human Evaluation. To better understand how our model performs in extracting explanations and what challenges still exist, we perform two analyses; a comparative and an absolute analysis. We consider two sets of data for evaluation: samples for the DialogRE test set where **N**o **L**abeled trigger exists (***NL***) and samples where the predicted explanation **D**iffers from the **L**abeled trigger (***DL***).

Comparative Analysis

In Table 2.4, we show the results for pairwise comparisons of explanations predicted by D-REX$_{\text{RoBERTa}}$ against 3 baselines: random strings of 1-4 words, predictions from $Joint_{\text{RoBERTa}}$, and labeled triggers. For each comparison, we employ 3 crowd-workers[4], who were given the full dialogue, a natural language statement corresponding to a relational triple, and the two proposed explanations highlighted in the dialogue (see

[4]Amazon Mechanical Turk workers were paid $0.35 per HIT, where a HIT includes 3 comparisons. We estimate an average HIT completion time of ~1.5 minutes, averaging ~$14 per hour. We only accept workers from AUS, CA, and USA.

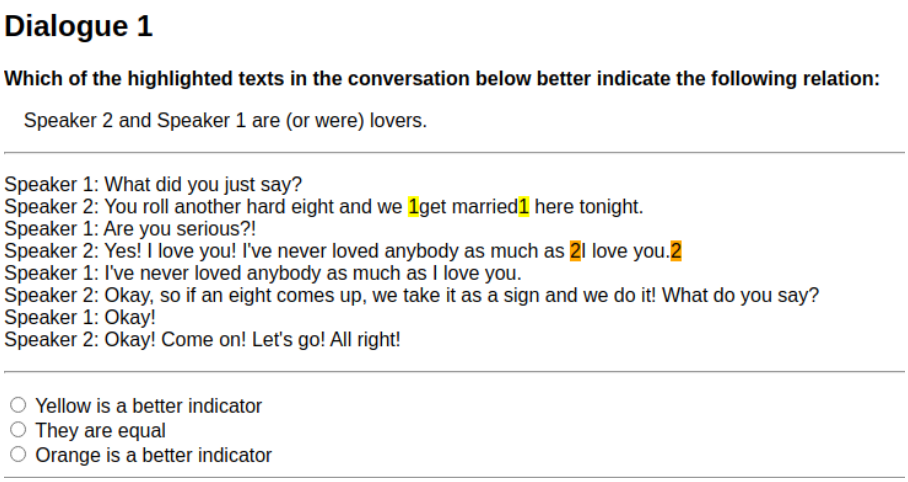

Figure 2.3: A sample HIT that was presented to crowd-workers for the comparative study of explanations.

Figure 2.3 for an example HIT). The crowd-workers were asked to specify which of the highlighted explanations was most indicative of the relation, or they could be equal. For each comparison we use a majority vote, and if there was a three-way tie we consider the explanations to be equal. We compare D-REX with random strings and the joint model on 174 samples from ***NL***, as well as 174 samples from ***DL***.

In Table 2.4 we see that for ***NL***, D-REX produces explanations which were 4.2 times more likely to be outright preferred by crowd-workers than the joint model, suggesting that our reward functions properly guided the explanation policy to learn meaningful explanations on unlabeled data. Surprisingly, we found that on over 12% of samples with labeled triggers, evaluators outright preferred D-REX explanations over the ground truth trigger, suggesting that D-REX indeed finds some explanations which are better than the ground truth trigger.

In section 2.5.5, we include 2 examples comparing explanations from D-REX and *Joint*.

D-REX$_{\text{RoBERTa}}$ vs.	**Win**(%)	**Tie**(%)	**Lose**(%)
Random (*NL*)	79.9	10.4	9.8
Joint$_{\text{RoBERTa}}$ (*NL*)	38.5	52.3	9.2
Ground truth (*DL*)	12.1	44.3	43.7

Table 2.4: **Human evaluator preferences on explanation extraction methods**. *NL* and *DL* are samples where **N**o **L**abeled trigger exists, and where the predicted explanation **D**iffers from the **L**abel, respectively. Results presented are percentages of preference.

	Not Indicative	Incorrect Entity Pair	Incorrect Relation	Indicative
NL	29	19	18	34
DL	19	13	7	61

Table 2.5: **Explanation error analysis** on 100 samples where **N**o **L**abeled trigger exists (*NL*) and 100 where the predicted explanation **D**iffers from the **L**abel (*DL*).

Absolute Analysis

To better understand the quality of D-REX's explanations, we randomly sample 100 explanations from both ***NL*** and ***DL*** for a fine-grained analysis. We classify the explanations into 4 categories: not indicative, incorrect entity-pair, incorrect relation, and indicative. "Indicative" and "Not indicative" have the obvious meanings, "Incorrect entity-pair" means that an explanation actually explains the correct relation, but between the incorrect entity-pair, and "Incorrect relation" means that the explanation indicates a relation different from the desired relation.

Table 2.5 shows the results. Interestingly, we see in the ***NL*** set, that errors were equally likely to come from either an explanation indicating the relation for an incorrect entity-pair as for the incorrect relation altogether. This is in contrast to the ***DL*** set, where D-REX was nearly half as likely to predict an explanation for an incorrect relation as it was for an incorrect entity-pair.

Additionally, in our fine-grained analysis, we also considered whether a relational triple was identifiable from the context alone and found that nearly 20% of the 200 samples had

Model	F1	Leave-one-out(↓)
D-REX$_{\text{RoBERTa}}$ (Full)	67.2	83.9
- reranking reward	66.0	84.9
- LOO reward	67.1	85.4

Table 2.6: **Ablation study** on reward functions. Leave-One-Out metric (LOO) measures how salient a predicted explanation is in determining a relation and is further defined and motivated in §2.5.4. Smaller LOO is better.

ambiguities which could not be resolved without outside knowledge. This suggests that there is likely a maximum achievable relation extraction score on the DialogRE dataset under the current setting.

2.5.4 Ablation Study

To assess the benefit of each proposed reward individually, we perform an ablation study on the reward functions. In order to study explanation quality automatically, we introduce a new metric for explanation quality; the Leave-One-Out metric.

The Leave-One-Out (LOO) metric has a theoretical basis in the works of Li et al. [53] and Ribeiro et al. [57], where Li et al. [53] use word erasure to determine a "word importance score". Here we define LOO formally. For a relation extraction model R, an explanation extraction model EX, and a dataset $\mathcal{D}$, LOO is calculated as

$$LOO(R, EX, \mathcal{D}) = \frac{\text{F1}_R(\mathcal{D}_{\text{MASK}}(EX))}{\text{F1}_R(\mathcal{D})}$$

where $\text{F1}_R(\mathcal{D})$ is the F1 score of R on $\mathcal{D}$ and $\mathcal{D}_{\text{MASK}}(EX)$ is the dataset where explanations predicted by EX are replaced by mask tokens. The LOO metric calculates how essential the predicted explanations are to the ability of the relation extraction model.

To show that LOO is an appropriate measure of explanation quality, we compute the Pearson correlation coefficient between token F1 score and LOO scores for models on labeled triggers, found in Table 2.3. With 6 models trained on 5 random seeds each, we

Dialogue	Subject Object Relation
... Speaker 1: Oh, I'm just so exhausted from dragging around this huge ***engagement ring***! ... Speaker 7: Hey, I'm sorry. I should have given you guys my black book when I got married! Although it wasn't so much a book as a...napkin. With Janice's phone number on it. ...	Janice Speaker 7 girl/boy-friend
Speaker 1: Sir? Speaker 2: What's ***in*** it? Speaker 1: Goat cheese, water chestnuts and panchetta. ... Speaker 3: Joey, it's been three days, okay. You're just a little homesick, okay. Would you just try to relax. Just try to enjoy yourself. Speaker 2: You're different here too. You're mean in England. ...	England Speaker 3 visited_by

Figure 2.4: Two examples comparing predicted explanations from D-REX (underlined) and *Joint* (***bold***).

have 30 data points and a correlation coefficient of -87.4 with $p = 2.4 * 10^{-8}$. Because we calculate the coefficient with respect to human-annotated triggers, this suggests that a low LOO correlates with explanations that humans would determine as indicative of the given relation.

For our experiments, we always calculate LOO using the baseline model, R_{BERT}. From the results in Table 2.6, we see that both reward functions benefit the final results. Compared with R_{RoBERTa}, D-REX$_{\text{RoBERTa}}$ gains 3 F1 points, but without the reranking reward, the model only gains 1.8 F1 score or 60% of the total possible improvement. This performance loss demonstrates that the reranking reward is critical to attaining the best score in relation extraction. Similarly, without the leave-one-out reward, the model's explanation quality, measured in LOO, is 1.5 points, or nearly 10% worse, demonstrating that the leave-one-out reward is beneficial in guiding the model to salient explanations.

2.5.5 Explanation Samples

Figure 2.4 shows two samples comparing explanations from D-REX and *Joint*. In both examples, even though there was no labelled trigger, each model was able to predict an explanation which correlates with the relation. Specifically, "engagement ring" and "got married" are related to the girl/boyfriend relation, and "in" and "mean in" can be associated with the visited_by relation. However, the bottom example shows that *Joint* did not consider the context surrounding it's explanation. The conversation is about food, and the visited_by relation is not relevant. On the other hand, D-REX finds the phrase "you're mean in", where "you're" refers to speaker3, and "in" refers to "England". This is clearly an explanation which indicates the correct relation between the correct entities.

2.5.6 Reduced Labels

All previous results use 100% of labeled triggers in the DialogRE dataset, which covers 40% of all relational triples. To test how few labeled triggers *EX* requires in order to learn meaningful explanations we ran a small scale experiment (1 random seed) using labeled triggers from only 5, 10, and 20% of relational triples. However, in the small tests we ran, we found that at 20% labeled triggers the *EX* model mostly predicts no explanations. Furthermore, at 10% and fewer labeled triggers, the model converges to the trivial solution in the explanation space which is to never predict any tokens.

We believe that this issue is due, in part, to two challenges: the search space over all possible start/end tokens is too large, and the policy gradient has a high variance. Although these results may seem discouraging, we believe this challenge can be overcome in the future by using algorithms which reduce variance in the policy gradient and by initializing *EX* with a model pre-trained in span extraction.

2.6 Related Work

Recently, there have been numerous information extraction tasks proposed which involve dialogues, including character identification [58], visual coreference resolution [59], emotion detection [60, 61].

New settings for relation extraction have also been proposed, such as web text [62] and, in many ways similar to dialogue, document text [40]. There have also been methods developed to include explanations in similar natural language understanding tasks [46, 63, 64, 65]. There have even been methods developed which, similarly to our re-ranking, make use of an explanation as additional information [66].

The work by Shahbazi et al. [52] is aligned with our study. They also focus on relation extraction with explanations; however, their method is based on distant supervision from bags of sentences containing an entity-pair. Due to the cross-sentence nature of relations in dialogue, their method is not applicable here, although we draw inspiration from their work. They explain their model by considering the salience of a sentence to their model's prediction, similarly to our leave-one-out reward.

Also relevant to our study is that by Bronstein et al. [67]. Their work focuses on the task of semi-supervised event trigger labeling, which is very similar to our semi-supervised prediction of relation explanations. In their work, they use only a small seed set of triggers and use a similarity-based classifier to label triggers for unseen event types.

Finally, there have been multiple recent studies in dialogue RE which perform quite well by using graph neural networks [43, 44, 45]. However, they focus only on RE and not on explaining the relations.

Chapter 3

Understanding Zero-Shot Transfer Learning

In the previous chapter, we focused on making models more understandable to humans, and in this chapter we shift our goal to a better understanding of the interplay between training datasets and model behavior on unseen tasks. Understanding zero-shot transfer learning has implications on our knowledge of model generalization, robustness, and adaptation in real-world scenarios. For example, if we are given a brand new task, understanding model generalization to unseen datasets will allow us to select the best model from a set of candidate models. Additionally, most machine learning approaches require large quantities of labeled data, which can be expensive and time-consuming to acquire and may sometimes be explicitly prohibited, but studying zero-shot transfer learning offers a promising direction to improve the data efficiency of machine learning methods. In this chapter, we study the zero-shot transfer learning ability of small language models using recent prompting techniques. We study transfer from models trained using multi-task learning on both in-domain and out-of-domain datasets to better understand how well models can generalize across not just datasets, but also across domains.

3.1 Introduction

Many recent works have demonstrated the benefits of prompting for large language models (see Liu et al. [68] for an extensive survey). The utilization of prompts has further expanded into the use of demonstrations, examples, and task instructions, all of which have been shown to improve the generalization of language models to unseen tasks [30, 69, 70]. Studies on utilizing prompts have also shown that as model sizes scale up, the generalization abilities of a model also increase [71, 72, 73]. However, utilizing models on the hundred-billion parameter scale is not accessible for most researchers and practitioners. Additionally, some use cases for language models, such as conversational agents, may have strict requirements for memory and latency, reducing the possible use cases for advances in prompting methods.

Similar efforts have demonstrated the benefits of task instructions in the dialogue domain [74]. However, some findings have been contradictory across studies. For example, Wei et al. [30] found that models with fewer than 8 billion parameters see decreases in generalization capabilities when training with instructions, whereas Gupta et al. [74] finds consistent gains in models with 3 billion and fewer parameters. To conflate these results further though, Gupta et al. [74] only consider 2 situations: when inputs include prompts and instructions, or if inputs include no prompt and no instruction at all.

Simultaneously with the emergence of prompting, the explicit multi-task learning (MTL) paradigm emerged, with works such as Muppet [75] or T0 [31] and their variants. Explicit MTL has been demonstrated as a means of improving the downstream performance of pre-trained language models in data-constrained settings. However, many prior studies of explicit MTL also do not consider models smaller than the billion parameter scale.

In this chapter we bridge the gap between previous studies by exploring the effects of a variety of factors on the zero-shot generalizability of modestly sized language models

(<500 million parameters). Specifically, we run experiments to find the effects of: (i) model size, (ii) general purpose MTL, (iii) in-domain MTL, and (iv) instruction tuning. Additionally, to better understand the sensitivity of models to instruction phrasing, we analyze variations in performance across task instructions.

In this chapter, we show that

1. In-domain multi-task learning (MTL) gives the largest improvements to generalizability, up to 80% increased performance, and 37.6% on average across all models

2. Increasing model size alone has little effect on generalization, but when combined with in-domain MTL leads to double the (already strong) performance improvement of in-domain MTL

3. General purpose MTL can provide large gains (57% improvement) for downstream tasks which closely resemble the MTL tasks, but still provides modest gains (5%) even for tasks which are more dissimilar

4. Instruction tuning during in-domain MTL provides modest gains of just over 2% performance, regardless of model size.

3.2 Preliminaries

Why should we study small models? Previous studies have shown that trends in large language models (>1 billion parameters) do not hold for smaller language models [32]. For this reason, it is crucial that we must empirically find the trends that occur in smaller models and cannot rely on studies of larger models. Additionally, for situations with latency and memory limitations, small models may be the only option. In particular,

we study zero- and few-shot performance on dialogue tasks, a sample domain in which reducing latency and memory usage are of high importance.

What are prompting methods? In this study, we convert all tasks to a sequence-to-sequence format, allowing for a single generative model to perform all tasks [76]. By treating all tasks as sequence-to-sequence, we can also include textual prompts as part of the text input. In this work, we focus on two types of prompts: answer templates and instructions. First, **answer templates** are a string of text added to the end of the input sequence that specifies the task and which allows the model to solve the task by filling in the template in natural language [68]. This is in contrast to more simple prompts which only specify the task by including an identifier (eg. "cola sentence" for linguistic acceptability, or "topic" for topic classification) [76, 10]. Second, we also consider **instructions**, which are generally added at the beginning of the input sequence and describe the task in natural language. For example, an instruction for document grounded generation is "Read the dialogue and the document text to generate a response."

How is explicit multi-task learning (MTL) used? Explicit MTL has emerged as a strong paradigm for eliciting zero-shot generalization in large language models [31]. In this work we consider 2 types of MTL: general purpose and in-domain. Specifically, general purpose MTL consists of training across a wide variety of tasks and domains, whereas in-domain MTL consists of training across a variety of tasks that all occur within a domain. In this work, we focus on the dialogue domain.

3.3 Experiments

Data. For this study, we utilize 46 annotated tasks from the Instructdial dataset [74]. Each task contains between 3 and 10 instructions, with 4.4 instructions on average across

all tasks. For our zero-shot experiments, we use 3 splits of train/test tasks, where each split contains 40 training tasks and 6 test tasks. Tasks are divided into classification and generation tasks, where classification tasks are evaluated on accuracy and generation tasks are evaluated by Rouge-L scores.

Task list. The full list of tasks is:

Act Classification, Act Generation, Advice Generation, Advice Present, Answer Generation, Answer Selection, Begins-with Controlled Generation, Belief State Generation, Count Response Words, Database-based Generation, Deal Present, Dialfact Classification, Document Grounded Generation, Edit Generation, Emotion Generation, Emotion Tagging, Ends-with Controlled Generation, Evaluation-Binary, Evaluation-Ranking, Fill-in the Missing Utterance, Find the Incoherent Utterance, Graph-based Generation, Intent Classification, Intent Present, Keyword Controlled Generation, Knowledge Grounded Generation, Natural Language Inference, Non-Toxic Feedback Generation, Persona Grounded Generation, Persuasion Generation, Persuasion Present, Persuasion Strategy, Question Generation, Recovery Generation, Relation Classification, Relation Present, Response Generation with n Words, Response Generation, Schema-based Generation, Slot Present, Slot Tagging, Slot-Value Generation, Summarization, Target Controlled Generation, Toxic Response Classification.

Models. In our experiments, we utilize 3 variants of the BART encoder-decoder model [77]: BART-Base, BART-Large and BART0++ [78]. BART0++ is a BART-Large that has been explicitly multi-task trained on PromptSource [79] in the same fashion as T0++ [31].[1]

[1]All pre-trained models were downloaded from the HuggingFace Transformers library.

Experimental Setup. To study the effects of (i) model size, (ii) general purpose MTL, (iii) in-domain MTL, and (iv) instruction tuning, we run a series of experiments. In order to measure the effect of (i) model size, we compare performance between BART-Base (139 million parameters) and BART-Large (406 million parameters). To measure the effect of (ii) general purpose MTL, we compare performance between BART-Large and BART0++. To study the effect of (iii) in-domain MTL, we train and test each model on all 3 of the data splits and compare against an off-the-shelf version of each model that is directly tested on each split without any in-domain MTL. These models utilize only an answer template without access to any instructions. To measure the effect of (iv) in-domain MTL with instructions, we train and test each model on all 3 data splits and include instructions in addition to the answer template in the prompt. All experiments were repeated with 3 random seeds, reported scores are means, and standard deviation is reported where appropriate

Additionally, we train all models for a maximum of 3 epochs, and utilize validation based early stopping. To determine the learning rate, we trained each model on a single seed and validate the best learning rate in {1e-5, 5e-5, 1e-4}, then train for 2 additional seeds using the best learning rate. We found for all models that 5e-5 was the best learning rate. For all experiments we use the AdamW optimizer.

3.4 Findings

Figure 3.1 shows the average model performance divided into classification and generation tasks. Figure 3.2 shows the absolute scores for all models and methods on all 18 zero-shot tasks.

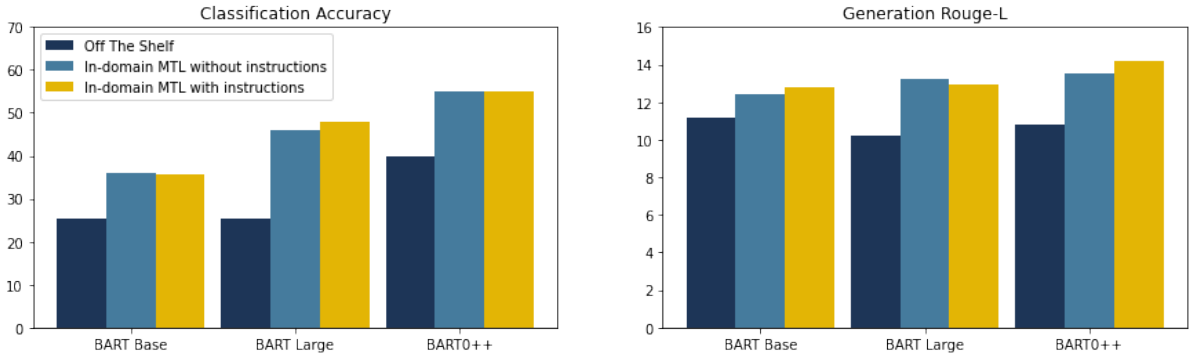

Figure 3.1: Average model performance on 10 zero-shot classification tasks (left) and 8 zero-shot generation tasks (right) comparing pre-trained models (Off the shelf) with models explicitly multi-task trained on in-domain data with and without instructions. BART0++ is a BART-Large model which has been explicitly multitask trained on PromptSource [79] in the same fashion as T0++ [31] and demonstrates the effect of explicit multi-task training prior to in-domain training.

Effects of Model Size. When comparing the average performance of off-the-shelf versions of BART-Base vs. BART-Large, we find nearly identical performance across classification tasks, and slightly better performance for BART-Base (11.2 vs. 10.2 Rouge-L) on generation tasks. However, the benefits of model size are demonstrated once the models have been further trained using in-domain MTL (Figure 3.1). We find that with in-domain MTL the base model improves its average score by 6.5. but the large model doubles that improvement, increasing it's score by 13.3 averaged across all tasks.

Effects of General Purpose Multi-task Learning. When comparing the performance of BART-Large vs. BART0++ we see improvements on 14/18 tasks, and an average absolute improvement of 14.5 accuracy (57.1% improvement) on classification tasks (Figure 3.1 left) and more modest improvement of 0.6 Rouge-L (5%) on generation tasks (Figure 3.1 right). This large discrepancy is likely due to the distribution of tasks in the P3 dataset [79] used to train BART0++, which consists almost entirely of classification tasks with only summarization as a generation task. Figure 3.2 shows that, indeed, an off-the-shelf BART0++ outperforms all other methods on summarization, including

in-domain MTL.

Effects of In-Domain Multi-Task Learning. We find that in-domain MTL (without instructions) contributes the largest portion to the final generalization of each model. As shown in Figure 3.1, BART-Large gets the most benefit with gains of 20.4 points in accuracy (80% improvement) on classification tasks and 3 Rouge-L (29.3% improvement) for generation tasks. Bart-Base gets 41.8% and 11.5% relative improvements on classification and generation tasks, respectively, and BART0++ gets 37.7% and 25.3% relative improvements on classification and generation, respectively. Collectively, this experiment and the previous experiments on general purpose MTL demonstrate the importance of matching both the domain and the task distribution during MTL to the downstream tasks and domain of interest. Additionally, as previously mentioned, in-domain MTL combined with the increased capacity of a larger model shows even greater improvements.

Effects of Instruction Tuning on In-Domain Multi-Task Learning. Finally, we compare the performance of in-domain MTL with and without instructions. The benefits of instruction tuning on small models is less prominent than the three previous variables, but is overall still beneficial. Figure 3.1 shows that BART-Base improves by 3% on generation tasks, but loses 1% accuracy on classification. To the contrary, BART-Large improves by 4% on classification tasks, and loses 2% Rouge-L on generation tasks. Interestingly, BART0++ sees no difference in performance on classification tasks and improves by 5% on generation tasks. These results run counter to those of Wei et al. [30], which found that instruction tuning can degrade performance of models with fewer than 8 billion parameters by about 10%. This is likely partly due to the in-domain nature of the instructions utilized in our experiments (all instructions are related to dialogue), suggesting that future works on instruction tuning for small models should focus on (1)

domain-specific wording used in instructions, and (2) expanding the number of domains included in instruction sets to see more general benefits.

Findings on Sensitivity to Instructions. To better understand the importance of wording and draw insights, we take a closer look at the tasks which had highest variance across instructions. First, we find that Answer selection is the task with highest variance (BART-Base has lowest score of 27.3 and highest score of 63) and find that the three worst performing instructions include variations of "select an option that can substitute <MASK>". The three instructions including this phrase average an accuracy of 39, while the remaining 7 instructions lead to an average accuracy of 60.1. This large discrepancy is likely connected to the unnaturalness of the <mask> token being used in the instruction, and that it is unlikely to have appeared in the BART pre-training corpus, and only appears in 2/46 tasks in our in-domain dataset. The other task which utilizes the <mask> token is the "Fill-in the Missing Utterance" task, which also achieves very poor performance across all models and methods (with and without instructions). This is a strong reminder that to create generalizability in language models, it is crucial to match the downstream task to the pre-training data.

Next, we analyze individual instruction words which most frequently give better than mean performance (see Figure 3.3 for full results). Interestingly, we find that "return" (as used in "return a response to the conversation") almost always leads to better than average performance (7/8 occurances for BART-Base and Large, and 8/8 for BART0++), although it only occurs in 3 tasks, and 8 instructions.

Finally, we look at the standard deviation between instructions, averaged across all tasks and find very little difference between models, with slightly increasing variation as models get larger, and are pretrained (BART-Base: 0.848, BART-Large: 0.867, BART0++: 0.882). At first glance, this seems to suggest that BART-Base is most robust to wording

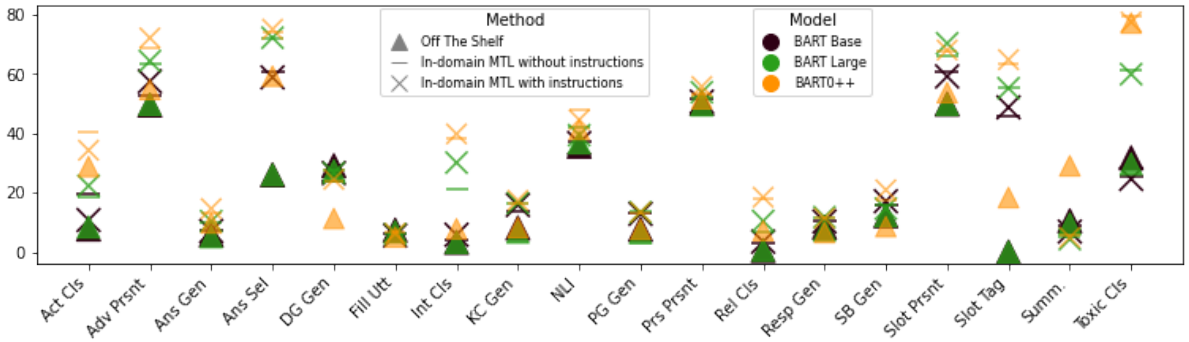

Figure 3.2: **Absolute scores on 18 zero-shot tasks**. Full task names are abbreviated as follows: Act Cls - Act Classification, Adv Prsnt - Advice Present, Ans Gen - Answer Generation, Ans Sel - Answer Selection, DG Gen - Document Grounded Generation, Fill Utt - Fill-in the Missing Utterance, Int Cls - Intent Classification, KC Gen - Keyword Controlled Generation, NLI - Natural Language Inference, PG Gen - Persona Grounded Generation, Prs Prsnt - Persuasion Present, Rel Cls - Relation Classification, Resp Gen - Response Generation, SB Gen - Schema-based Generation, Slot Prsnt - Slot Present, Slot Tag - Slot Tagging, Summ. - Summarization, and Toxic Cls - Toxic Response Classification.

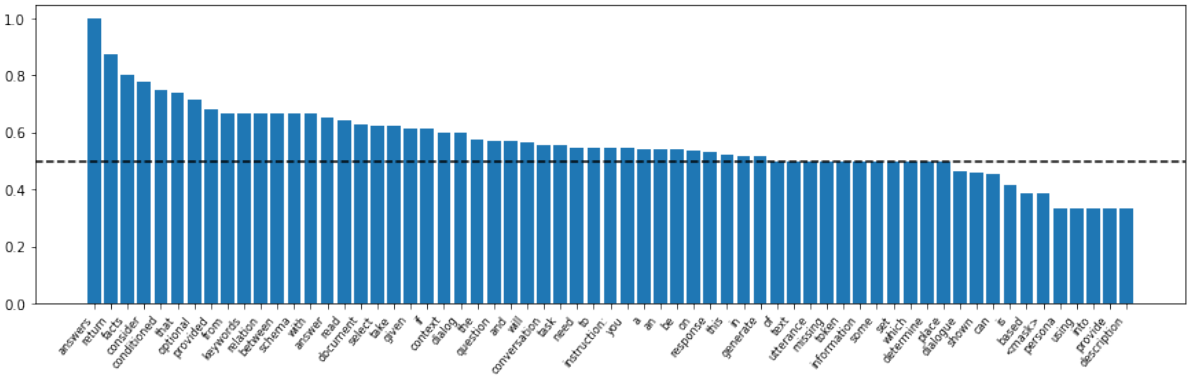

Figure 3.3: Percentage of occurrences of a word that lead to better than average performance for an instruction. Results calculated from BART-Base model and only includes words that occur is more than 5 instructions.

in instructions, but this is actually due to the smaller number of tasks which BART-Base can meaningfully perform, as seen in Figure 3.2.

Chapter 4

Understanding Few-Shot Transfer Learning

Understanding transfer learning can be beneficial not only in zero-shot settings as discussed in the previous chapter, but also in few-shot settings where we have a small amount of data available for our target task. Additionally, while understanding the transfer from multiple tasks is useful, as shown in the previous chapter, it is important to dive deeper into the understanding of transfer from individual source tasks to individual target tasks, as in task transfer. Previous studies of task transfer collect tasks from disjoint datasets without regard for the effects that domain adaptation may have on their results, leading to a gap in our knowledge on transfer learning.

This chapter studies a very narrow and focused problem, intra-dataset task transfer, where both the source and target task are from the same distribution (avoiding domain adaptation). To study intra-dataset task transfer, we first create a large-scale benchmark, FETA, with 132 source-target task pairs, and perform considerable experimentation and analysis comparing different models, learning algorithms, sample sizes, and task types.

4.1 Introduction

Improving sample efficiency through transfer learning has been a long-standing challenge in the machine learning and natural language processing communities [80, 81]. Recently, transfer learning using pre-trained language models as an initial substrate has become integral to performing language tasks [82, 83, 47]. Dialogue data requires multiple cohesive turns with consistent speaker personalities [84, 85], creating a challenge for data collection and motivating the development of techniques that improve sample efficiency in conversational AI [86].

Furthermore, dialogue understanding tasks require a shared knowledge of semantics, pragmatics, human behavior, and commonsense, making dialogue an area of study that can benefit greatly from a deeper understanding of transfer learning.

Two essential transfer learning settings, namely domain adaptation and task transfer, have been studied on language tasks [87]. While domain adaptation has been studied in task-oriented dialogue [88], task transfer has been studied with less rigor in conversational AI. Prior studies of task transfer in dialogue consider only 2-4 tasks, focus on multitask learning, and do not compare learning algorithms [89, 90].

Prior studies have focused on cross-dataset task transfer, gathering tasks annotated on disjoint datasets [91, 92], but this can lead to improvements in domain adaptation being confounded as improvements in task transfer. A precise study of task transfer should be on a single data source in an intra-dataset transfer setting, as in Zamir et al. [93]. Additionally, previous studies focus on learning algorithms and use only a single language model architecture [94, 95, 96], which may lead to a narrow understanding. To the best of our knowledge, this is the first rigorous study on task transfer in dialogue and the most extensive intra-dataset task transfer study in NLP.

In this chapter, we create FETA, a benchmark for **FE**w-sample **TA**sk transfer for

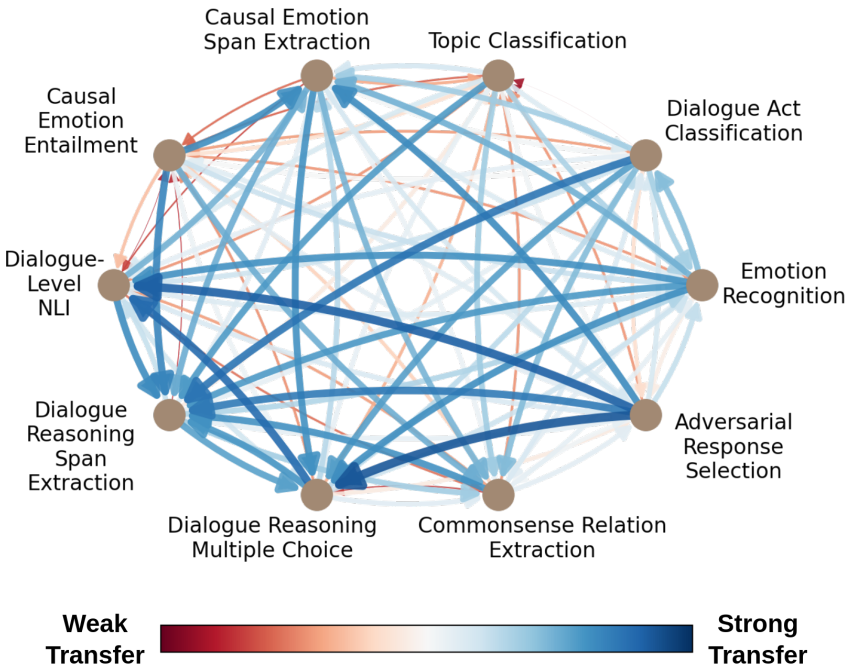

Figure 4.1: **Task Transfer Performance** on FETA-DailyDialog. Computed transfer performance is demonstrated by arrows leaving from source tasks and entering target tasks. Strength of the transfer is denoted by thickness and color of edges.

language understanding in open-domain dialogue with 17 total tasks. FETA datasets cover a variety of properties (dyadic vs. multi-party, anonymized vs. recurring speaker, varying dialogue lengths) and task types (utterance-level classification, dialogue-level classification, span extraction, multiple-choice), and maintain a wide variety of data quantities.

We study task transfer on FETA by comparing three task transfer algorithms and three commonly used language models in single-source and multi-source settings. Figure 4.1 illustrates some results in the single-source setting. For example, we find that Dialogue Reasoning Span Extraction benefits from nearly all source tasks. On the other hand, Adversarial Response Selection and Emotion Recognition improve the performance of many target tasks when utilized as a source task.

In this study, we find that: **(i)** Trends are largely model-dependent, a finding that

previous works have not discussed. **(ii)** Out of all task types, span extraction tasks gain the most as a target, especially with few samples. **(iii)** Adding source tasks does not uniformly improve over a single source task, motivating a better understanding of the complex relationship between source and target tasks.

FETA provides a resource for various future studies, e.g., on the generalizability of model architectures, and pre-training datasets that enable efficient transfer. In addition to task transfer, FETA can also facilitate the study of continual and multitask learning.

In summary, our main contributions are:

- We create the first large-scale benchmark for task transfer in dialogue, FETA, with 132 source-target task pairs.
- Extensive experimentation on FETA in both the single-source and multi-source settings, and an in-depth analysis comparing models, learning algorithms, sample sizes, and task types, finding new and non-intuitive results.
- A readily extensible transfer learning framework[1] that allows for rapid experimentation and an online leaderboard[2] to encourage deeper research into task transfer.

4.2 Related Work

Transfer Learning in NLP. Prior works on transfer learning in NLP have studied a wide variety of topics, including domain adaptation [97], multitask learning [98, 99], and learning representations of words [100, 101, 82, 83]. More recently, DialoGLUE [88] and RADDLE [102] study domain adaptation for language understanding tasks in task-oriented dialogue. Shuster et al. [103] focuses on multitasking in dialogue response

[1] github.com/alon-albalak/TLiDB
[2] alon-albalak.github.io/feta-website/

generation across multiple datasets. Jandaghi et al. [104] develop a measure for predicting transferability across datasets. Albalak et al. [11] studies zero-shot transfer from multitask training to in- and out-of-domain unseen tasks. Similar to this chapter, Pruksachatkun et al. [94] study task transfer, although they study cross-dataset task transfer in general NLP tasks. They perform an analysis of transfer by using probing tasks to discover which source tasks transfer best, but find that the probing task performance doesn't always align well with the target task performance and show that further study is required. Unlike this chapter, they study task transfer across datasets, allowing for domain adaptation as a confounding variable in their experiments. Lourie et al. [95] also study task transfer, but they focus on the T5 model and a suite of commonsenseQA datasets.

Task Transfer in Dialogue. Task transfer has been applied in Task-Oriented Dialogue (TOD) settings but never rigorously studied. For example, Hosseini-Asl et al. [89] and Lin et al. [86] develop multitask models to perform 2-4 TOD tasks but do not aim to analyze the efficiency of models or learning algorithms for task transfer.

Intra-dataset Task Transfer. Intra-dataset task transfer has been studied in computer vision applications [93, 105], but to our best knowledge it has never been studied in NLP.

4.3 Intra-Dataset Task Transfer with FETA

In this section, we briefly define *intra-dataset task transfer*, the problem setting of FETA. Then, we introduce FETA, our benchmark for few-sample task transfer in open-domain dialogue. Finally, we define the metrics we use to evaluate models and learning algorithms on FETA.

	Task Name	Original Samples	FETA Samples Train	Dev	Test	Task Type	Metrics
DailyDialog	Emotion Recognition	102978	7230	1269	15885	Utt Cls	M/m-F1
	Dialogue Act Classification	102978	7230	1269	15885	Utt Cls	M/m-F1
	Topic Classification	13118	958	161	1919	Dial Cls	M/m-F1
	Causal Emotion Span Extraction	36324	2141	169	9133	Span Ex	T-F1,EM
	Causal Emotion Entailment	36324	2141	169	9133	Dial Cls	M-F1,Acc
	Dialogue-Level NLI	5817	569	52	1302	Dial Cls	M-F1,Acc
	Dialogue Reasoning Span Extraction	1098	123	13	244	Span Ex	T-F1,EM
	Dialogue Reasoning Multiple Choice	2165	224	26	496	Mult Ch	Acc
	Commonsense Relation Extraction	4009	350	38	851	Dial Cl.	M-F1,Acc
	Adversarial Response Selection	57145	3400	895	10750	Mult Ch	Acc
Friends	Emotion Recognition (EmoryNLP)	12606	844	207	1912	Utt Cls	m/W-F1
	Reading Comprehension	13865	912	181	2284	Mult Ch	Acc
	Character Identification	50247	3593	638	7803	Utt Cls	M/m-F1
	Question Answering	12257	819	191	1937	Span Ex	T-F1,EM
	Personality Detection	711	54	15	110	Dial Cls	Acc
	Relation Extraction	7636	519	121	1188	Dial Cls	m-F1
	Emotion Recognition (MELD)	9140	616	148	1247	Utt Cls	m/W-F1

Table 4.1: **Overview of FETA tasks**. Task types are abbreviated as follows: Utt Cls for utterance-level classification, Dial Cls for dialogue-level classification, Span Ex for span extraction, and Mult Ch for multiple choice. Metrics are abbreviated as follows: M-F1 for macro-F1, m-F1 for micro-F1, T-F1 for token-F1, W-F1 for weighted-F1, EM for exact match and Acc for accuracy.

4.3.1 Problem Definitions

Let a dataset be composed of the instance set, X, and n task-specific label sets $Y_1, Y_2, \ldots, Y_n$. In FETA, each instance $x \in X$ is a dialogue.

Definition 1 (Domain and Task)**.** *A* domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ *consists of a feature space* $\mathcal{X}$ *and a marginal probability distribution* $P(X)$*. The marginal probabilities are over the instance set* $X = \{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$*.*

A task $\mathcal{T} = \{\mathcal{Y}, f(X)\}$ *is composed of a label space* $\mathcal{Y}$ *and a predictive function,* $f : \mathcal{X} \rightarrow \mathcal{Y}$*.*

Definition 2 (Learning Algorithm)**.** *A* learning algorithm*,* $\mathcal{A}$*, is a protocol that determines the method by which the instance set* X *and task-specific label sets* $Y_1, Y_2, \ldots, Y_n$ *will be used to train a predictive function,* f*.*

Definition 3 (Task Transfer)**.** *Given a source task* $\mathcal{T}_S = \{\mathcal{Y}_S, f_S(X_S)\}$ *and target task* $\mathcal{T}_T = \{\mathcal{Y}_T, f_T(X_T)\}$*,* task transfer *is the use of a learning algorithm,* $\mathcal{A}$*, to improve the*

learning of f_T by using the knowledge in $\mathcal{T}_S$.

In **cross-dataset task transfer**, when $X_S \neq X_T$, we also have $P(X_S) \neq P(X_T)$ and $\mathcal{D}_S \neq \mathcal{D}_T$; domain shift.

In **intra-dataset task transfer**, when $X_S = X_T$, there is no domain shift. This enables the study of the learning algorithm's performance on task transfer, isolated from domain adaptation.

We refer the reader to Pan and Yang [106] and Zhuang et al. [107] for expanded discussions on transfer learning definitions.

Few-Sample. Due to the challenge and cost of collecting and annotating data, many real-world applications of NLP techniques are limited by data quantities. For this reason, we focus on the few-sample setting, defined in FETA as 10% of the original instance set. Out of 10%, 5%, and 1%, 10% was empirically determined to be the smallest percentage that retains labels from all label sets in both the train and development partitions. Given the recent attention focused on NLP applications in low-resource settings [108, 109, 14, 110, 92], we expect research done in such a low-data setting will lead to insights useful for many researchers and practitioners.

4.3.2 FETA Datasets

In this section, we describe the two dialogue sources we use, DailyDialog [111] and Friends [112], and the tasks annotated on each source.

We select these datasets because they complement each other in desirable ways. DailyDialog contains 2-speaker dialogues where speakers are anonymized and averages 88 words per dialogue. In contrast, Friends consists of multiparty dialogues (3.6 speakers mean, 15 max) with recurring characters and averages 283 words per dialogue. These

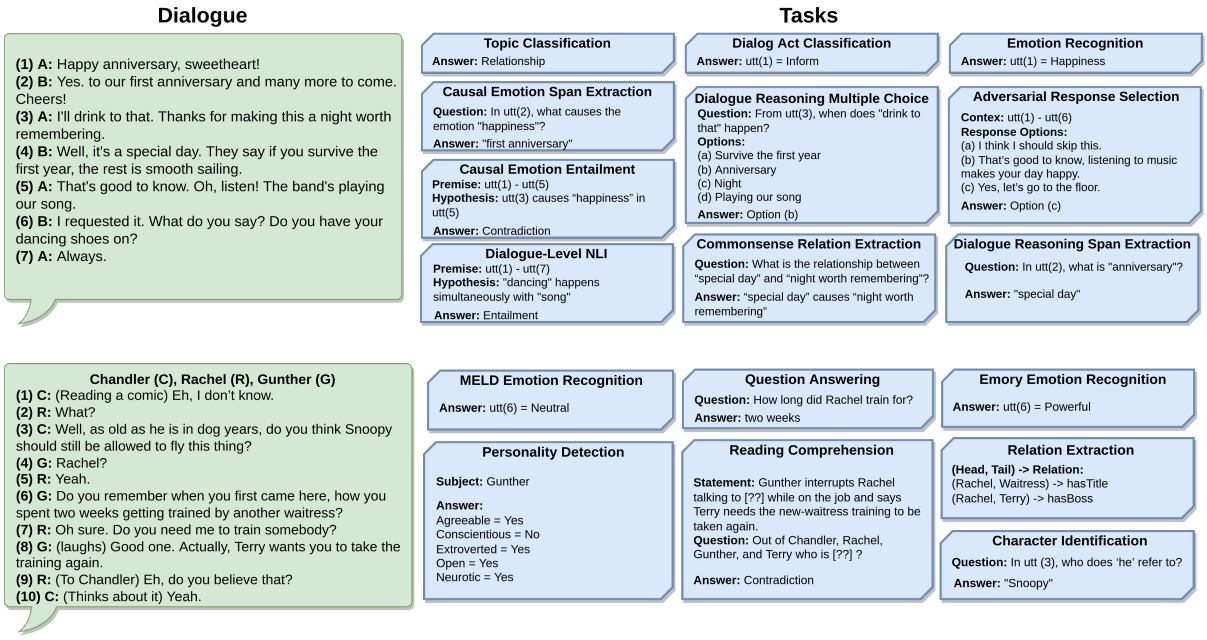

Figure 4.2: **Example dialogues and tasks** for FETA-DailyDialog (top) and FETA-Friends (bottom).

differences lead to each set of dialogue instances having different task annotations, giving FETA a wider variety of tasks. For example, DailyDialog tasks include understanding the causes of emotions and commonsense reasoning, while tasks annotated on Friends revolve more around recognizing entities and understanding personalities.

To create FETA versions of each dataset, we first partition the dialogues into 70/15/15% splits for training, validation, and test sets. After splitting, we randomly down-sample the train and development dialogues to 10% of the original quantities. Thus, FETA splits use 7/1.5/15% of the original dialogues. Not every dialogue is annotated for all tasks, allowing some tasks to have more samples than others. Crucially, the data splits are the same for all tasks, preventing data leakage. Table 4.1 shows an overview of the tasks, samples, and metrics used for each dataset.

FETA-DailyDialog. Li et al. [111] present the DailyDialog dataset, with chit-chat

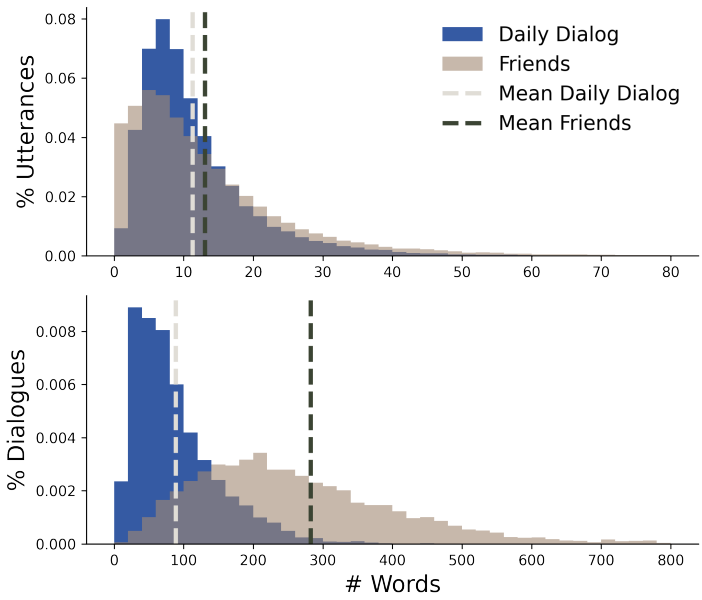

Figure 4.3: **Utterance and dialoguelength distributions in FETA.**

conversations covering 10 various topics including relationships, politics, and work.

Many works add annotations on top of these dialogues and FETA utilizes 10 of them. Figure 4.2 provides an overview of the tasks: *emotion recognition*, *dialogue act classification*, *topic classification* (from DailyDialog [111]), *causal emotion span extraction*, *causal emotion entailment* (from RECCON [113]), *dialogue-level natural language inference*, *dialogue reasoning span extraction*, *dialogue reasoning multiple choice*, *commonsense relation extraction* (from CIDER [114]) *adversarial response selection* (from DailyDialog++ [115]). For further details of these tasks, we refer the reader to their original papers.

FETA-Friends. The Friends dialogues come from transcripts of 10 seasons of the TV show by the same name [112]. In addition to dialogue, the transcripts contain situational information such as behaviors and non-verbal information like scene information.

In total, FETA has 7 task annotations on top of the Friends scripts. As illustrated in Figure 4.2, the incorporated tasks include *Emory emotion recognition* (from [60]),

reading comprehension (from [116]), *character identification* (from [112, 58]), *question answering* (from [117]), *personality detection* (from [118]), and *relation extraction* (from DialogRE [119]) and *MELD emotion recognition* (from MELD [120]). There are two emotion recognition label sets (Emory and MELD), but they have only 22% overlap in instance sets and have different label spaces. For further details of these tasks, we refer the reader to their original papers.

4.3.3 Evaluation Metrics

To define the metrics, we consider 4 variables: source task s, target task t, model f, and learning algorithm $\mathcal{A}$, and we abuse notation slightly to allow for $f_{\mathcal{A}}(s,t)$ to represent a model trained on the source and target tasks using the given learning algorithm. In FETA, we evaluate the performance of a model and learning algorithm with multiple metrics: average and top-1 raw scores, as well as average and top-1 score Δs.

Average and Top-1 Scores. First, we consider the two raw scores: average score and top-1 score. These metrics aim to answer the following questions: How well do a model and algorithm perform across all task pairs, and, how well do a model and algorithm perform supposing that we knew the best source task a priori.

We calculate an average score across all source-target task pairs to understand how each model and algorithm performs in the aggregate. Formally, let the score for a single task be computed as:

$$score(s,t,f,\mathcal{A}) = \frac{1}{|M_t|}\sum_{i=1}^{|M_t|} M_{t,i}(f_{\mathcal{A}}(s,t))$$

where M_t is the set of metrics associated with task t, found in Table 4.1, and $M_{t,i}(f)$ is the ith calculated metric of model f on task t. *All metrics range from 0 to 100.* Then, we

calculate the average score as:

$$\text{Average Score}(f, \mathcal{A}) = \frac{\sum_{t \in \mathcal{T}} \sum_{s \neq t \in \mathcal{T}} score(s, t, f, \mathcal{A})}{|\mathcal{T}| \times (|\mathcal{T}| - 1)}$$

where $\mathcal{T}$ is the set of tasks.

Additionally, we calculate top-1 score to understand how models and algorithms perform if the best source task is known ahead of time. This score is calculated as the maximum score over source tasks averaged over target tasks. The top-1 score does not consider scores less than the baseline, which is a model trained directly on the target task. Denote the baseline algorithm by $\mathcal{A}_{\mathcal{B}}$ and the baseline score as $score(s, t, f, \mathcal{A}_{\mathcal{B}})$. Formally, the top-1 score is calculated as:

$$\text{Top-1}(f, \mathcal{A}) = \frac{\sum_{t \in \mathcal{T}} \max_{s \neq t \in \mathcal{T}} \Big(score(s, t, f, \mathcal{A}_{\mathcal{B}}), score(s, t, f, \mathcal{A}) \Big)}{|\mathcal{T}|}$$

Average and Top-1 Δs. In addition to raw scores, we also calculate score differences to measure how much a source task benefits a target task. The average Δ describes how much benefit the model saw in the aggregate over all source tasks, while the top-1 Δ considers only the best source. Score Δs are calculated with respect to the baseline score as:

$$\Delta(s, t, f, \mathcal{A}) = score(s, t, f, \mathcal{A}) - score(s, t, f, \mathcal{A}_{\mathcal{B}})$$

and the average Δ is calculated as:

$$\text{Average } \Delta(f, \mathcal{A}) = \frac{\sum_{t \in \mathcal{T}} \sum_{s \neq t \in \mathcal{T}} \Delta(s, t, f, \mathcal{A})}{|\mathcal{T}| \times (|\mathcal{T}| - 1)}$$

Additionally, we calculate the top-1 Δ as the maximum positive score difference over

Model	Transfer Algorithm	DailyDialog Average Score (σ)	DailyDialog Average Δ	DailyDialog Top-1 Source Score	DailyDialog Top-1 Source Δ	Friends Average Score (σ)	Friends Average Δ	Friends Top-1 Source Score	Friends Top-1 Source Δ
BERT	Pre-train/Fine-tune	50.61 (0.24)	-0.93	52.22	+0.68	42.39 (0.30)	-0.89	44.36	+1.08
	Multitask	50.95 (0.24)	-0.59	52.40	+0.86	42.88 (0.29)	-0.40	45.14	+1.86
	Multitask/Fine-tune	**51.40** (0.25)	-0.15	52.76	+1.22	**44.69** (0.28)	+**1.41**	**46.00**	+**2.72**
GPT-2	Pre-train/Fine-tune	39.80 (0.25)	-1.28	42.19	+1.11	32.66 (0.18)	-0.64	34.34	+1.04
	Multitask	40.21 (0.24)	-0.86	41.77	+0.69	33.10 (0.16)	-0.20	34.83	+1.53
	Multitask/Fine-tune	41.15 (0.23)	+0.07	42.76	+1.68	34.62 (0.15)	+1.32	35.86	+2.56
T5	Pre-train/Fine-tune	49.92 (0.37)	+0.19	**53.04**	+**3.31**	41.73 (0.19)	-1.10	43.52	+0.69
	Multitask	49.49 (0.42)	-0.24	52.98	+3.25	40.42 (0.20)	-2.40	43.33	+0.51
	Multitask/Fine-tune	50.29 (0.36)	+**0.56**	52.85	+3.12	42.29 (0.17)	-0.53	43.87	+1.05

Table 4.2: **Average and Top-1 Source task transfer scores.** Average scores and Δs aggregate scores over all source tasks, compared with Top-1 scores and Δs which are calculated with scores from the highest performing source task. Δs are the difference from the baseline score without task transfer. Highest values for each model are underlined, highest values across all models are bolded.

source tasks averaged over target tasks:

$$\text{Top-1 } \Delta(f, \mathcal{A}) = \frac{\sum\limits_{t \in \mathcal{T}} \max\limits_{s \neq t \in \mathcal{T}} \Big(0, \Delta(s, t, f, \mathcal{A})\Big)}{|\mathcal{T}|}$$

4.4 Task Transfer Algorithms

In this chapter, we consider three commonly used task transfer methods: Pre-train/Fine-tune, Multitask, Multitask/Fine-tune. We apply these methods with cross-entropy loss to further optimize pretrained language models on FETA.

Pre-train/Fine-tune. Commonly used in NLP today, the pre-train/fine-tune algorithm consists of two stages of training [80]. First, the model is trained on the source task $\mathcal{T}_S$, optimizing Eq 4.1, followed by a separate stage of training on the target task $\mathcal{T}_T$, optimizing Eq 4.2:

$$\mathcal{L}_S = \underset{(x, y_s) \sim \{X, \mathcal{Y}_S\}}{-\mathbb{E}} \big[\log p(y_s|x)\big] \tag{4.1}$$

$$\mathcal{L}_T = \underset{(x, y_t) \sim \{X, \mathcal{Y}_T\}}{-\mathbb{E}} \big[\log p(y_t|x)\big] \tag{4.2}$$

Multitask. In this algorithm, there is only a single stage of multitask training [121]. Formally, the training is conducted on both the source and target task by optimizing Eq 4.3:

$$\mathcal{L}_{S,T} = \underset{(x,y_s,y_t)\sim\{X,\mathcal{Y}_S,\mathcal{Y}_T\}}{-\mathbb{E}} \big[\log p(y_s|x) + \log p(y_t|x)\big] \tag{4.3}$$

Multitask/Fine-tune. This algorithm combines the previous algorithms in two stages. In the first stage, the source and target task are optimized jointly, as in Eq 4.3. Then, the second stage trains using only the target task, as in Eq 4.2.

Even though model selection in multitasking is generally done w.r.t. multiple source and target tasks [121], we modify the setting to validate a model on a single target task at a time. This allows hyperparameter search and early stopping to be controlled by the desired target task.

4.5 Experiment Setup

To study task transfer on FETA, we run extensive experimentation. We utilize three task transfer algorithms: pre-train/fine-tune, multitask, and multitask/fine-tune, as described in Section 4.4. To draw broad conclusions about the performance of each learning algorithm, we utilize pretrained language models with three different architectures: encoder-only (BERT) [47], decoder-only (GPT-2) [50], and encoder-decoder (T5) [122].

Additionally, we use the pretrained model implementations from the HuggingFace Transformers library [49], where the bert-base-uncased model has 110M parameters, GPT-2 has 124M parameters, and T5-base has 223M parameters. We use the Adam optimizer [123] with a batch size of 60 and run a learning rate sweep across $\{3\times10^{-6}$, 1×10^{-5},3×10^{-5},$1\times10^{-4}\}$ during the pre-training phase, finding that 3×10^{-5} worked well across all models. In all experiments we utilize validation-based best model selection, and

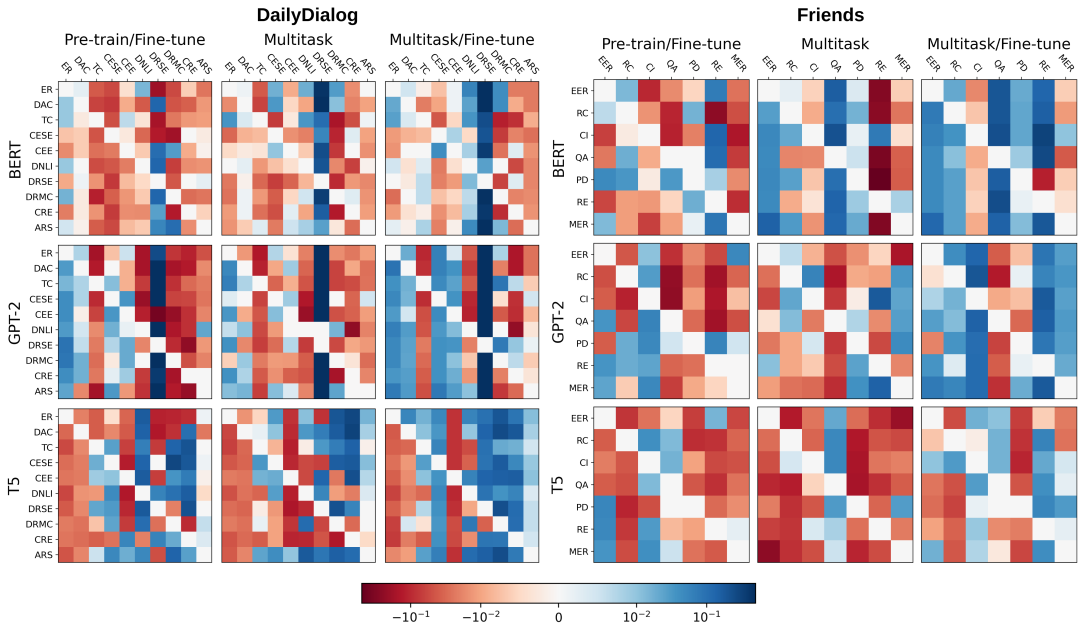

Figure 4.4: **Relative improvement of transfer over fine-tuned baselines**. Rows are source tasks and columns are target tasks. Diagonal cells are baseline scores. Looking at an individual column can demonstrate best source tasks for that target. Looking at rows can determine which source task works well across multiple targets.

train models for 30 epochs on DailyDialog tasks and 20 epochs on Friends tasks.

A complete experiment for a single target task, $\mathcal{T}$, is as follows: First, we directly fine-tune on $\mathcal{T}$ to get the baseline score. Then, for each source task, $\mathcal{S}$, we take the model pre-trained on $\mathcal{S}$ and fine-tune on $\mathcal{T}$. Next, we jointly train on $\mathcal{S}$ and $\mathcal{T}$ together. Finally, we fine-tune the jointly trained model on $\mathcal{T}$.

FETA datasets have 10 and 7 tasks, giving $90 + 42 = 132$ unique source-target task pairs. Our experiments include three learning algorithms, three models, and we run each experiment with 5 random seeds. In total, we run $132 \times 3 \times 3 \times 5 = 5940$ transfer experiments, and $17 \times 3 \times 5 = 255$ baseline experiments leading to 6195 trained models.

In addition to the single-source setting described above, we also consider a subset of tasks to study in the multi-source setting, where multiple tasks are simultaneously used as source tasks to transfer to a single target task (4.6.2). For our experiments, we select

two target tasks from each dataset that benefit the most from task transfer, and we use the three source tasks that transferred best onto those targets.

4.6 Results and Analysis

4.6.1 Single-Source Setting

Table 4.2 shows the results for all three models and algorithms, and we use this table to understand general trends. Figure 4.4 shows the relative improvement of a source task for each target task, demonstrating trends across tasks.

Aggregate Performance. We find that, on average, Friends tasks get scores between 7-8 points less than DailyDialog, likely due to the greater number of speakers and utterance length of Friends. We find that GPT-2 lags behind the raw scores of BERT and T5 by $\sim$10 points. This is expected as autoregressive decoder models are not designed with classification in mind. We find that the largest average Δ is 1.4, leaving room for improvement in task transfer on FETA.

Furthermore, we are interested in knowing: how much is gained by using the best source task vs. a random source task. We calculate the differences between average Δ and top-1 Δ and find the mean difference to be $\sim$1.6 and the largest difference to be $\sim$3.5, motivating a further understanding of which source tasks transfer best to target tasks.

Performance Across Learning Algorithms. We average scores across both datasets and find that pre-train/fine-tune gets an average score of 42.85, multitask 42.84, and multitask/fine-tune 44.07. Table 4.2 shows that multitask/fine-tune achieves the best average score for all models and datasets, and indeed its average score is a 2.8% improvement over the other algorithms. However, aggregate scores obscure some interesting nuances.

Do Trends Vary Across Models? Previous studies on task transfer have focused on a single model [94, 95, 96], but we find that trends vary depending on the model. For example, we find results similar to Lourie et al. [95], namely, that fine-tuning on the target task always benefits the T5 model. However, we discover that this does not hold for BERT and GPT-2, which achieve better scores from multitasking than pre-train/fine-tune.

Furthermore, Figure 4.4 shows that trends on individual tasks also vary depending on the model. For example, T5 positively transferred knowledge to question answering with all learning algorithms and from most source tasks, while GPT-2 had a negative transfer from all algorithms and sources.

For *nearly all* dimensions of analysis (e.g., sample sizes, learning algorithm), we find different trends between models. We *strongly suggest that future research be performed on multiple models* before attempting to draw broad conclusions on transfer learning. In particular, any trends should be tested and verified in existing and future architectures that differ from transformers such as state space models [124, 125] and linear attention models [126, 5].

Multitask/Fine-tune As Regularization. We find that T5's top-1 score and Δ on DailyDialog are highest for pre-train/fine-tune, but the average score and Δ are highest for multitask/fine-tune. To understand why, we find the bottom-1 scores for T5 on DailyDialog: 46.78, 46.69, and 48.26 for pre-train/fine-tune, multitask, and multitask/fine-tune algorithms, confirming that multitask/fine-tune does achieve the best worst-case performance. Moreover, we find that for all datasets and models, multitask/fine-tune does achieve the best worst-case performance. In fact, for GPT-2 on Friends, utilizing the bottom-1 source tasks still leads to a 0.74% improvement over the baseline.

Do All Task Types Benefit Equally? We find that *span extraction tasks gain the most as target tasks*, shown in Figure 4.5 to benefit at all source-to-target sample ratios. Multiple choice tasks also stand to gain from task transfer, but we find that only occurs at a 10:1 ratio of source-target samples. This gain is likely due to the high-level language understanding required by both tasks.

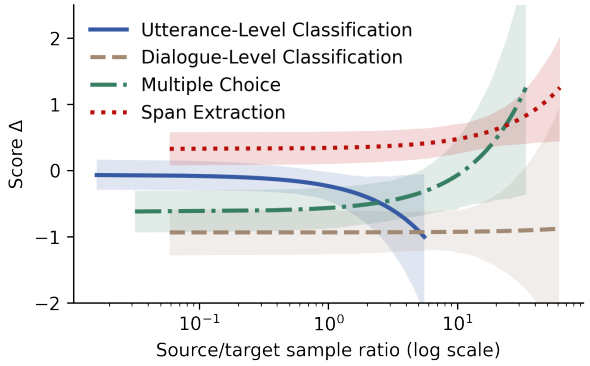

Figure 4.5: **Score Δ by target task type.** Lines show the average score Δ when the target task is of the specified task type, computed as a best-fit linear interpolation of the data with a 95% confidence interval. The number of samples for an individual task are fixed, but source/target ratios vary depending on which task pair is used.

Additionally, we find that utterance-level classification tasks decrease in score Δ at increasing source-to-target sample ratios. This is possibly due to models overfitting to specific tasks and a catastrophic forgetting of general skills learned during their large-scale pre-training.

Do All Task Types Give Equal Benefit? We find that *multiple-choice tasks give the greatest benefit as source tasks*, especially when the ratio of source-to-target samples is low, as shown in Figure 4.6. Additionally, we find that at a ratio of 10:1 source-target samples, dialogue-level classification benefits downstream tasks, but utterance-level classification requires a ratio of 100:1.

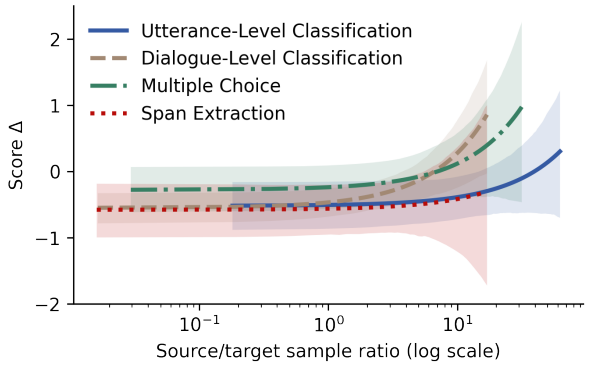

Figure 4.6: **Score Δ by source task type.** The number of samples for an individual task are fixed, but source/target ratios vary depending on which task pair is used.

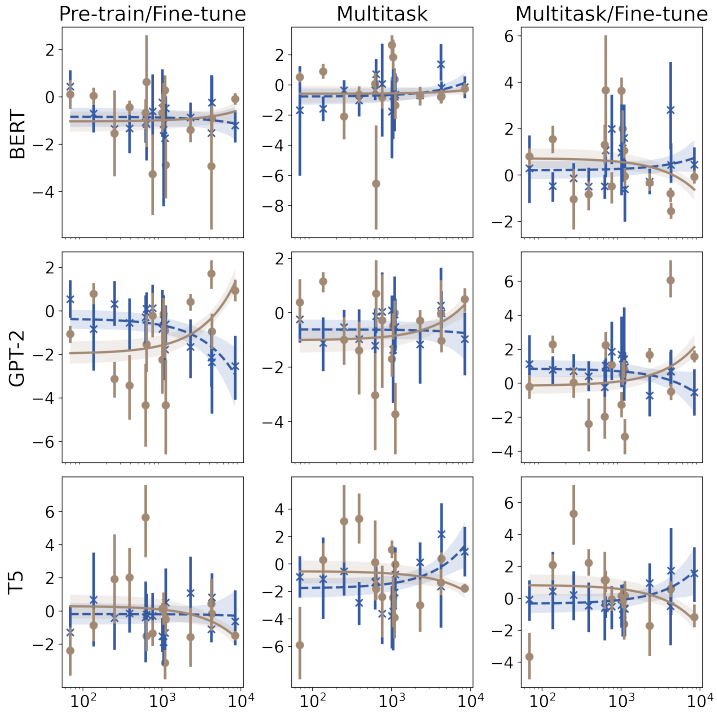

Figure 4.7: **Score Δ by sample count**. Sample count is on the x-axis (log scale) and score Δ is on the y-axis. The blue dotted line represents the average transfer Δ from a source task to all target tasks. The brown line represents the average transfer Δ to a target task from all sources. Trend lines are a linear best-fit on the data with a 95% confidence interval. The number of samples for an individual task are fixed, but source/target ratios vary depending on which task pair is used.

How Do Sample Sizes Affect Transfer? Figure 4.7 shows that, interestingly, GPT-2 and T5 have opposite trends in relation to sample size. We find that Δs for GPT-2 increase with high target samples and decrease with high source samples. This suggests that GPT-2 may be overfitting to the source task and performs better with resource-rich target tasks. We find that T5 Δs decrease as target-task samples increase, *suggesting that T5 is more sample efficient* than both GPT-2 and BERT.

4.6.2 Multi-Source Setting

For multi-source transfer we select the two target tasks from each dataset with the best score differences from the single-source setting, shown in Figures 4.8 and 4.9. We

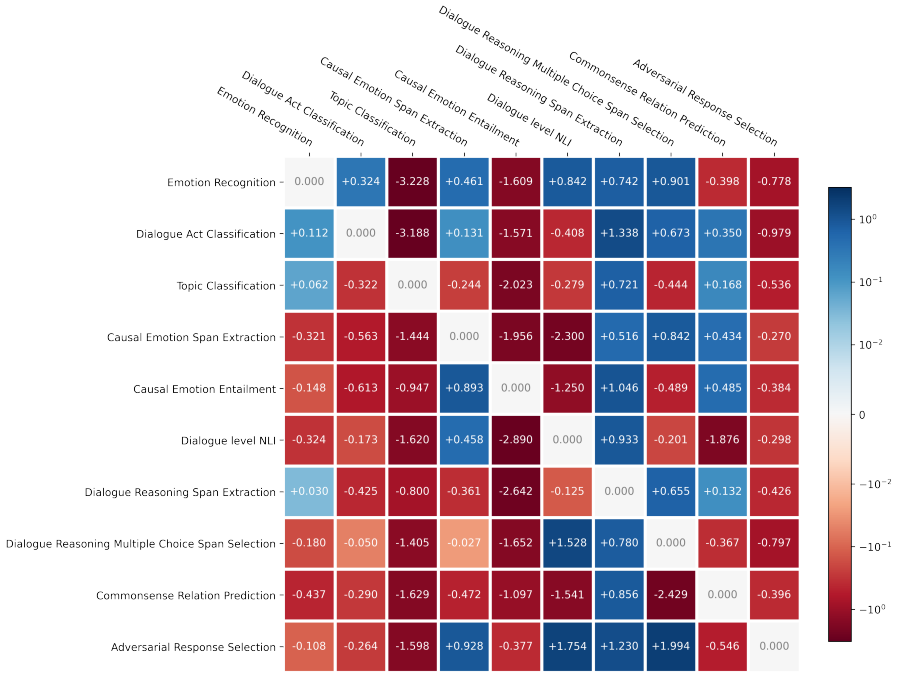

Figure 4.8: **Aggregate task transfer performance on DailyDialog.**

find those four tasks to be Dialogue Reasoning Span Extraction (DRSE), Dialogue-Level NLI (DNLI), Character Identification (CI), and Question Answering (QA). For each of these target tasks, we select the top-3 best source tasks, shown in Table 4.4. Learning in this setting is similar to single-source, except we now simultaneously optimize the loss for multiple source tasks. Table 4.3 shows the multi-source results compared with the average score of the top-3 source tasks from the single-source setting. Full results, including score Δs from the single-source baselines, average top-3 score Δs, and multi-source score Δs are in Table 4.4.

Does Multi-source Improve Over Single-source? We expect that by utilizing the top-3 source tasks from the single-source setting, the multi-source setting will improve performance for all models and algorithms, but find results to the contrary. We find that 6/9 multi-source algorithms outperform their average top-3 single-source counterparts in

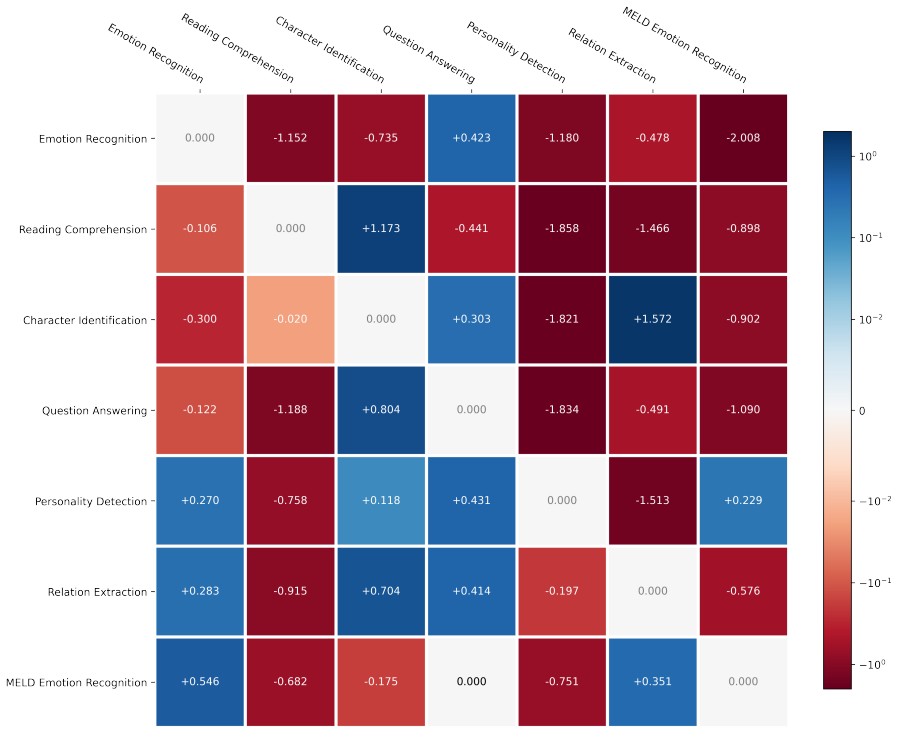

Figure 4.9: **Aggregate task transfer performance on Friends.**

DRSE, 6/9 for DNLI, 3/9 for CI, and only 2/9 for QA, showing that naively combining source tasks is not always beneficial. The impressive result for DRSE follows our original intuition, given that there is an almost unanimous benefit from all source tasks, shown in Figure 4.4. Similarly, we find that *multi-source performance on CI also correlates with the performance of individual source tasks.* We find that in the single-source setting GPT-2 is the only model that improves with any source task, and indeed GPT-2 sees benefits from multi-source training on all algorithms.

Which Models Benefit From Multi-Source? Table 4.4 shows that GPT-2 improves in 8/12 experiments over its average top-3 single-source counterparts, but BERT only 5/12 and T5 in only 4/12 experiments. It is counter-intuitive that T5 should perform the worst as we expect that it has a higher capacity for learning due to twice the model size.

	Target	DRSE	DNLI	CI	QA
BERT	P/F	-1.18	+1.37	-2.11	-0.99
	M	+2.77	+1.57	-0.54	-1.14
	M/F	+1.61	+2.28	-0.34	-0.55
GPT-2	P/F	+0.40	+0.16	+4.25	-3.90
	M	+0.78	+0.98	+1.28	-2.46
	M/F	+0.73	-0.09	+0.00	-0.95
T5	P/F	+0.60	+1.95	-0.79	+0.48
	M	-1.08	-0.96	-1.49	+0.08
	M/F	-1.22	-1.20	-0.24	-0.22

Table 4.3: **Multi-source score Δs from the average score of the top-3 source tasks.** Full results, including score Δs from the fine-tuned baseline are in Table 4.4.

On the other hand, the additional parameters may be causing T5 to overfit on training data in the few-sample setting.

Target		DRSE				DNLI				CI				QA				
		DAC	ARS	CEE	Top-3 Av.	Multi-Source	ARS	DRMC	ER	Top-3 Av.	Multi-Source	RC	QA	RE	Top-3 Av.	Multi-Source	PD	ER	RE	Top-3 Av.	Multi-Source
BERT	P/F	0.46	0.17	0.43	0.35	-0.83	-0.35	-0.39	0.88	0.05	1.48	-1.21	-0.76	-1.15	-0.16	-2.27	0.58	-0.28	-0.08	0.07	-0.92
	M	1.86	0.15	0.86	0.96	3.73	0.53	0.32	1.05	0.63	2.20	-0.77	-1.48	-0.27	-0.84	-1.38	1.89	2.98	2.62	2.50	1.36
	M/F	2.58	2.04	1.40	2.01	3.62	2.66	0.40	3.55	2.20	4.48	-0.58	-0.78	-0.48	-0.61	-0.95	3.04	3.64	4.49	3.72	3.17
GPT-2	P/F	0.93	1.14	-0.3	0.59	0.99	-3.65	0.00	-6.99	-3.55	-3.39	1.29	2.73	1.09	1.70	5.95	0.12	-1.76	-0.66	-0.77	-4.67
	M	1.30	1.59	0.89	1.26	2.04	-0.81	-1.73	-0.94	-1.16	-0.18	2.70	-1.03	-0.26	0.47	1.75	-1.59	-1.00	-1.14	-1.24	-3.70
	M/F	3.43	2.01	1.70	2.38	3.11	0.46	-0.32	-1.92	-0.59	-0.68	8.81	6.69	5.08	6.86	8.81	-1.31	-0.84	-0.83	-0.99	-1.94
T5	P/F	-3.08	-1.08	-1.48	-1.88	-1.28	2.52	5.53	8.60	5.55	7.50	2.22	0.70	1.59	1.50	0.71	0.03	-0.19	-0.31	-0.16	0.32
	M	1.54	1.77	2.93	2.08	1.00	8.83	5.83	0.55	5.07	4.11	-1.84	-0.30	0.22	-0.64	-2.13	1.10	0.82	0.27	0.73	0.81
	M/F	3.00	3.30	2.99	3.10	1.88	5.59	4.10	2.78	4.16	2.96	-0.06	1.46	0.52	0.64	0.40	0.02	0.42	-0.63	-0.06	-0.28

Table 4.4: **Results from the multi-source experiment**, where we use the top-3 source tasks in a multi-source task transfer setting. We include individual scores from all 3 top-3 source tasks and include their average score as a comparison. Multi-source experiments that improve over the top-3 average are underlined.

Part II

Improving Models Through Data

Chapter 5

Improving Few-Shot Generalization

In Part I of this dissertation, we focused on understanding models through the data they were trained on. In this chapter, we shift our focus towards improving the training data for models by applying the lessons learned. Specifically, we focus here on selecting better data for training a model in the few-shot setting. In Chapter 4, we showed that jointly training on the top-3 source tasks together does not always lead to better target task performance over using a single source task, which may be counter-intuitive as we often assume that more data is better. However, the results demonstrate that for transfer learning, some data is more valuable than others.

In this chapter, we develop algorithms that automatically select training data in the aim of improving few-shot generalization. To develop efficient algorithms, we frame each source dataset (henceforth referred to as auxiliary datasets) as the arm of a multi-armed bandit, and design reward functions that appropriately model the desired relation between auxiliary and target data.

5.1 Introduction

Few-shot learning is an attractive learning setting for many reasons: it promises efficiency in cost and time, and in some scenarios data is simply not available due to privacy concerns or the nature of the problem. However, few-shot learning is also a challenging setting that requires a delicate balance between learning the structure of the feature and label spaces while preventing overfitting to the limited training samples [127, 128, 129]. One approach to improving the generalizability of models in the few-shot setting is **F**ew-shot **L**earning with **A**uxiliary **D**ata (FLAD), where additional auxiliary datasets are used to improve generalization on the target few-shot task [130, 131, 132, 133].

However, FLAD methods introduce their own challenges, including increased algorithmic and computational complexity. Specifically, incorporating auxiliary data during training introduces a large space of design choices (e.g. how and when to train on auxiliary data). Manually designing the curriculum for training on large quantities of auxiliary data is not feasible due to the combinatorially large search space, and hand-picking which auxiliary data to use based on heuristics (e.g. from the same domain or task as the target few-shot dataset) can lead to sub-optimal results [10]. Delegating such choices to an algorithm can lead to better solutions, as demonstrated in the transfer learning [134, 135, 94], meta-learning [136, 137], multi-task learning [138, 75, 96, 31], and auxiliary learning literature [130, 139]. However, prior auxiliary learning algorithms often assume that only 1-3 related auxiliary datasets are available and design algorithms whose computational complexity grows linearly (or worse) with the number of auxiliary datasets [140, 10], motivating the search for more efficient methods as the number of auxiliary datasets grows.

To overcome the challenges of prior works, we desire a FLAD algorithm that **(1)** makes no assumptions on available auxiliary data a-priori (in-domain, on-task, quality,

quantity, etc.), **(2)** scales well with the number of auxiliary datasets, and **(3)** adds minimal memory and computational overhead. We design algorithms that satisfy our desiderata by drawing inspiration from the central problem in multi-armed bandit (MAB) settings: the exploration-exploitation trade-off [141, 142]. We relate the set of auxiliary datasets to the arms of a MAB and tailor the classic EXP3 [143] and UCB1 [144] algorithms to fit the FLAD framework by designing three efficient gradient-based reward signals. The combination of our MAB-based algorithms and efficient gradient-based rewards allows us to scale to 100× more auxiliary datasets than previous methods. Figure 5.1 provides a basic illustration of how we formulate FLAD as a MAB problem.

To empirically validate our approaches, we focus on few-shot training of language models and utilize P3 [79], a readily available resource with hundreds of auxiliary language datasets. We evaluate our methods on the same held-out tasks as the T0 language model [31] and show that, when using the same collection of auxiliary datasets, our algorithms outperform a directly fine-tuned T0 by 5.6% (EXP3-FLAD) and 5.7% (UCB1-FLAD) absolute. Furthermore, incorporating all available datasets in P3 (i.e. not just those used to train T0) increases the improvement to 9.1% and 9.2%. Finally, we compare models trained with our methods against state-of-the-art few-shot methods, finding that our methods improve performance by >3%, even though one model utilizes a large collection of unlabeled target dataset samples. Furthermore, to the best of our knowledge, our methods lead to the first 3 billion parameter model that improves over 175B GPT-3 using few-shot in-context learning.

In summary, our main contributions are:

- We connect FLAD to the MAB setting and focus on the exploration-exploitation trade-off by designing two algorithms, EXP3-FLAD and UCB1-FLAD along with three reward functions that are both simple and efficient (in space and computational

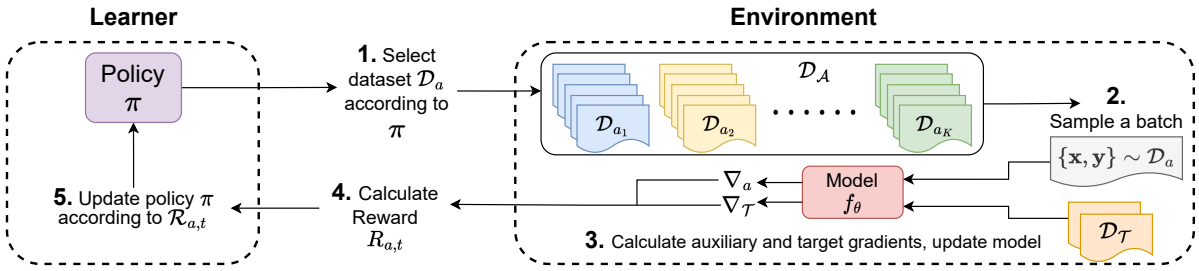

Figure 5.1: **Overview of few-shot learning with auxiliary data (FLAD) as a multi-armed bandit problem.** On the left is the learner which defines a policy π that determines which auxiliary dataset to sample from. On the right is the environment that includes the set of auxiliary datasets $\mathcal{D}_{\mathcal{A}}$, target dataset $\mathcal{D}_{\mathcal{T}}$, and the model f_θ. At each turn t, the following five steps take place, further described in Section 5.3.1: **1.** The learner selects an auxiliary dataset $\mathcal{D}_a$ according to its policy π. **2.** The environment samples a batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$. **3.** The model f_θ calculates gradients for the sampled batch (∇_a) and the target dataset ($\nabla_{\mathcal{T}}$), then updates the parameters θ. **4.** A reward $\mathcal{R}_{a,t}$ is calculated based on ∇_a and $\nabla_{\mathcal{T}}$. **5.** The learner updates π based on $\mathcal{R}_{a,t}$.

complexity).

- We empirically validate that our methods improve few-shot performance of pre-trained language models and show that strategies that employ only exploration *or* exploitation lead to sub-optimal performance.

- We perform case studies to better understand the dynamics of our reward functions and their interaction with the dynamics of large language model training.

5.2 Related Work

A long history of works have found success when combining auxiliary data with target data [130, 145, 132, 146, 147, 131, 140, 133, 78, 148, 134]. Some works have explored the addition of auxiliary learning objectives to aid the learning of the target task [145, 147, 146, 131, 139]. More similar to this study are methods that perform

auxiliary learning by introducing additional data sources beyond the target data [130, 132, 140, 133, 78, 148, 10]. As opposed to the few-shot setting on which this chapter focuses, previous works have studied auxiliary learning in settings with large quantities of target data. For example, Chen et al. [140] and Verboven et al. [133] assume access to 10,000 labeled target samples, Ivison et al. [148] and Lin et al. [78] assume access to 1,000s of unlabeled target samples, and Du et al. [132] and Albalak et al. [10] assume access to 100s of labeled target samples. Additionally, many of the previous works that study auxiliary learning have only considered settings with 1-3 auxiliary datasets [132, 140, 133, 10]. For example, Verboven et al. [133] propose a task-weighting method that requires solving a system of equations that becomes underspecified with multiple auxiliary tasks, limiting their method to only a single auxiliary task. Furthermore, Chen et al. [140] experiment with 3 auxiliary tasks because their method requires learning a target-aware classifier for each source task, so the computation scales as $O(|\mathcal{A}||\mathcal{T}|)$ where $|\mathcal{A}|$ is the number of auxiliary tasks and $|\mathcal{T}|$ is the number of target tasks, making it impractical to scale to large numbers of source and target tasks. In this chapter, we focus on improving auxiliary learning with very few target samples (20-70 samples) by scaling up the number of auxiliary datasets orders of magnitude greater than previous work. In order to scale up the learning process, efficiency is a central concern of this chapter, unlike prior works.

Data selection studies a similar (but distinct) problem where the goal is to selectively utilize a subset of a single large dataset rather than selecting data from auxiliary datasets. Recent research on data selection has found that intelligent data selection can provide significant improvements to model performance [149, 150, 151, 152].

5.3 Multi-armed bandits for few-shot learning with auxiliary data

In this section, we first define the few-shot learning with auxiliary data (**FLAD**) setting. Then, we formulate FLAD as a multi-armed bandits (**MAB**) problem, shown in Figure 5.1. Next, we define reward functions that are efficient to compute and appropriate for FLAD. Finally, we describe our adaptations of two popular MAB algorithms: EXP3-FLAD and UCB1-FLAD.

5.3.1 Setup

FLAD problem setting. Few-shot learning with auxiliary data (FLAD) fits into the following setting: assume access to a large set of auxiliary datasets $\mathcal{D}_{\mathcal{A}}$ where, for all $a \in \mathcal{A}$, $\mathcal{D}_a$ is an individual auxiliary dataset. Given a small quantity of data belonging to a target dataset $\mathcal{D}_{\mathcal{T}}$, the goal of FLAD is to find parameters θ of a model f_θ that achieve high performance on the unknown distribution underlying $\mathcal{D}_{\mathcal{T}}$ while utilizing only the available data, $\mathcal{D}_{\mathcal{T}} \cup \mathcal{D}_{\mathcal{A}}$.

Formulating FLAD as MAB. In this chapter, we adopt the multi-armed bandit (MAB) setting by formulating FLAD as a Markov decision process [153] and defining a learner and environment, illustrated in Figure 5.1. The learner consists of a policy π defining a selection strategy over all $\mathcal{D}_a \in \mathcal{D}_{\mathcal{A}}$. The environment consists of the target dataset $\mathcal{D}_{\mathcal{T}}$, auxiliary datasets $\mathcal{D}_{\mathcal{A}}$, and model f_θ. In this formulation the learner interacts with the environment over N rounds. At each round t the learner selects one of the environment's $|\mathcal{A}|$ datasets $\mathcal{D}_a \in \mathcal{D}_{\mathcal{A}}$. Next, the environment samples a batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$ and calculates the gradient w.r.t. θ using a task-appropriate loss function as $\nabla_a = \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$. Then, the environment computes the target gradient

$\nabla_{\mathcal{T}} = \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}})$, and updates model parameters w.r.t. $\nabla_{\mathcal{T}} + \nabla_a$. Finally, the learner uses a gradient-based reward $\mathcal{R}_{a,t}(\nabla_a, \nabla_{\mathcal{T}})$ to update its policy π. See Lattimore & Szepesvári [154] for further details on multi-armed bandits.

Designing the reward functions. We design the reward function $\mathcal{R}$ with our desiderata in mind. To ensure that our algorithm adds minimal memory and computational overhead we consider rewards that utilize information intrinsic to the model and the losses being optimized, not an external model or metric (e.g. accuracy or BLEU). In this chapter we propose three gradient-based reward functions inspired by previous studies: **gradient alignment** [132, 145, 155], **gradient magnitude similarity** [156, 157], and their aggregation. Formally, at turn t let ∇_a be the gradient of the auxiliary batch and $\nabla_{\mathcal{T}}$ be the target dataset gradient. **Gradient alignment** is defined as $\mathcal{R}^{GA}_{a,t} = \frac{\nabla_a \cdot \nabla_{\mathcal{T}}}{\|\nabla_a\|_2 \|\nabla_{\mathcal{T}}\|_2}$, i.e. the cosine similarity between the gradients of the sampled auxiliary dataset batch and the whole target dataset. **Gradient magnitude similarity** is defined as $\mathcal{R}^{GMS}_{a,t} = \frac{2\|\nabla_a\|_2 \|\nabla_{\mathcal{T}}\|_2}{\|\nabla_a\|_2^2 + \|\nabla_{\mathcal{T}}\|_2^2}$ so that when the two gradients have equal magnitude, this value is equal to 1 and as the magnitudes differ the value goes to zero. In addition to the individual reward functions, we also consider an aggregate reward. To ensure that the aggregate is not dominated by either individual reward, we normalize $\mathcal{R}^{GA} \in [0, 1]$, the same range as $\mathcal{R}^{GMS}$ and define the aggregate to be their sum: $\mathcal{R}^{AGG}_{a,t} = \frac{1+\mathcal{R}^{GA}_{a,t}}{2} + \mathcal{R}^{GMS}_{a,t}$. We provide further discussion on the design of reward functions in Section 5.6.

5.3.2 Adapting the EXP3 algorithm.

EXP3 Background. We base our first algorithm, EXP3-FLAD, on the EXP3 algorithm [143] ("*Exp*onential-weight algorithm for *Exp*loration and *Exp*loitation"). EXP3 targets the adversarial MAB setting, which assumes that the reward-generating process is controlled by an adversary who is given access to the learner's policy π and determines

the sequence of rewards, $(R_{a,t})_{t=1}^{N}$, for each arm prior to play [158]. We consider the adversarial MAB formulation due to the highly non-convex loss landscape of deep neural networks and our use of stochastic gradient descent-based optimization methods. These factors imply that we cannot guarantee our rewards to be stationary, independent, or follow any particular distribution (e.g. Gaussian). Further details on adversarial MAB can be found in [143].

In EXP3-FLAD, the learner selects arms according to a Gibbs distribution based on the empirically determined importance-weighted rewards of arms [159]. To allow for exploration, we mix the Gibbs distribution with a uniform distribution [143]. Formally, let $\mathcal{E}_t$ be the exploration rate at turn t and, recalling that $K = |\mathcal{A}|$ is the number of auxiliary datasets, then π defines the probability of selecting a given arm $a \in \mathcal{A}$ as the linear combination of Gibbs and uniform distributions $\pi_t(a) = (1 - K\mathcal{E}_t)\frac{\exp(\mathcal{E}_{t-1}\hat{R}_a)}{\sum_{a'}\exp(\mathcal{E}_{t-1}\hat{R}_{a'})} + \mathcal{E}_t$ where $\hat{R}_{a,t}$ is the importance weighted reward $\hat{R}_{a,t} = \hat{R}_{a,t-1} + \frac{R_{a,t}}{\pi_{t-1}(a)}$. We want the learner to explore more in early training than in later stages, so we use a decaying exploration rate $\mathcal{E}_t = \min\left\{\frac{1}{K}, \sqrt{\frac{\ln K}{K \cdot t}}\right\}$ as proposed by Seldin et al. [159]. The use of an importance-weighted estimated reward compensates the rewards of actions that are less likely to be chosen, guaranteeing that the expected estimated reward is equal to the actual reward for each action. EXP3-FLAD is designed to be nearly optimal in the worst case, but due to the exploration rate it will select "bad" actions at a rate of $\mathcal{E}_t$. The exploration of EXP3-FLAD combined with importance-weighting allows the policy to handle non-stationary reward-generating processes.

5.3.3 Adapting the UCB1 algorithm.

UCB1 background. While EXP3-FLAD is applicable in unconstrained settings with highly stochastic and non-stationary rewards, it can be outperformed by other algorithms

in settings that *are* constrained. One such algorithm is the upper confidence bound (UCB1) algorithm [144], which was originally designed to be optimal for stationary, normally distributed reward functions. Nevertheless, variants of UCB1 have been demonstrated to be effective in a range of settings, such as those involving non-stationary, sub-Gaussian, or heavy-tailed distributions [160, 161]. The UCB1 algorithm and its variants assign each arm a value called the upper confidence bound based on Hoeffding's inequality [162] and are based on the principle of *optimism in the face of uncertainty*, meaning that with high probability the upper confidence bound assigned to each arm is an overestimate of the unknown mean reward.

In UCB1-FLAD, the learner greedily selects arms according to their upper confidence bound. UCB1 is originally designed for stationary reward-generating processes, so to accommodate non-stationarity we include an exponential moving average when estimating the mean reward for a given arm. Formally, let $R_{a,t}$ be the observed reward for arm a at turn t, then we calculate the estimated mean reward as $\hat{R}_a = (1-\beta)\hat{R}_a + \beta R_{a,t}$ where β is the smoothing factor. Then, we define the upper confidence bound to be $UCB_{a,t} = \hat{R}_a + \sqrt{\frac{2\ln t}{n_a}}$. In the original MAB setting all interactions with the environment occur online, but FLAD is a unique situation where the learner can interact with the auxiliary data prior to training. To take advantage of this, rather than initializing estimated rewards with a single mini-batch, we initialize them with larger data quantities to improve the approximation of the true dataset gradients. This is done for each auxiliary dataset by calculating the gradient $\nabla_a = \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$, where the number of samples in $\{\mathbf{x}, \mathbf{y}\}$ can be significantly larger than a mini-batch, and can be up to the size of the full dataset. In practice, we use 1,000 examples which is computed in ~ 2 minutes on a single GPU.

Algorithms. The EXP3-FLAD and UCB1-FLAD algorithms are visualized in Figure 5.1 and pseudocode is found in Algorithms 2 and 3.

At each turn, both methods will first select an auxiliary dataset $\mathcal{D}_a$. EXP3-FLAD first computes the current exploration rate $\mathcal{E}_t$ and samples $\mathcal{D}_a$ according to the distribution defined by $\pi_t(\mathcal{A})$, while UCB1-FLAD greedily selects $\mathcal{D}_{a^*}$ corresponding to the arm with largest upper confidence bound, $a^* = \arg\max_{a \in \mathcal{A}} UCB_{a,t}$. Next, for both methods, the environment samples a batch from the selected dataset, $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$, and calculates the gradient $\nabla_a = \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$. Let G be the number of rounds between model updates, then the previous steps will repeat G times, at which point the environment calculates the gradient of the target dataset $\nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_\mathcal{T})$ and updates the model w.r.t. $\nabla_\mathcal{T} + \sum_a \nabla_a$. Finally, EXP3-FLAD calculates the importance-weighted reward for each auxiliary batch using the observed rewards, while UCB1-FLAD calculates the smoothed estimated mean reward.

Algorithm 2 EXP3-FLAD

Require: $\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{T}$: Auxiliary and target datasets
Require: f_θ: Parameterized model
Require: G: Gradient accumulation steps
Initialize: $K = |\mathcal{A}|$; $\mathcal{E}_0 = \frac{1}{K}$; $\forall a \in \mathcal{A} : \nabla_a = 0$, $\hat{R}_a = 1$
for $t = 1, 2, \ldots, N$ **do**
  $\mathcal{E}_t = \min\left\{\frac{1}{K}, \sqrt{\frac{\ln K}{K \cdot t}}\right\}$
  $\forall a \in \mathcal{A} : \pi(a) \leftarrow (1 - K\mathcal{E}_t)\frac{\exp(\mathcal{E}_{t-1}\hat{R}_a)}{\sum_{a'} \exp(\mathcal{E}_{t-1} R_{a'})} + \mathcal{E}_t$
  Sample $a \sim \pi(\mathcal{A})$ and batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$
  $\nabla_a \leftarrow \nabla_a + \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$
  if $t \pmod G \equiv 0$ **then**
    $\nabla_\mathcal{T} \leftarrow \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_\mathcal{T})$
    Update model parameters w.r.t.$\nabla_\mathcal{T} + \sum_a \nabla_a$
    for all $\{a \in \mathcal{A} | \nabla_a \neq 0\}$ **do**
      $\hat{R}_a \leftarrow \hat{R}_a + \frac{R_{a,t}}{\pi(a)}$
      $\nabla_a \leftarrow 0$
    end for
  end if
end for

Algorithm 3 UCB1-FLAD

Require: $\mathcal{D}_{\mathcal{A}}, \mathcal{D}_{\mathcal{T}}$: Auxiliary and target datasets
Require: f_θ: Parameterized model
Require: G: Gradient accumulation steps
Require: β: Smoothing factor

Initialize: $\forall a \in \mathcal{A} : n_a = 1, \quad \hat{R}_a = \cos(\nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}}), \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_a))$
for $t = 1, 2, \ldots, N$ **do**
  $a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \ \hat{R}_a + \sqrt{\frac{2 \ln t}{n_a}}$
  Sample batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_{a^*}$
  $\nabla_{a^*} \leftarrow \nabla_{a^*} + \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$
  $n_{a^*} \leftarrow n_{a^*} + 1$
  if $t \pmod G \equiv 0$ **then**
    $\nabla_{\mathcal{T}} \leftarrow \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}})$
    Update model parameters w.r.t. $\nabla_{\mathcal{T}} + \sum_a \nabla_a$
    for all $\{a \in \mathcal{A} | \nabla_a \neq 0\}$ **do**
      $\hat{R}_a \leftarrow (1 - \beta)\hat{R}_a + \beta R_{a,t}$
      $\nabla_a \leftarrow 0$
    end for
  end if
end for

5.4 Experimental setup

Models. For our experiments, we utilize encoder-decoder Transformer models from the T5 family of pre-trained language models [163]. Specifically, we experiment with LM-adapted T5 (T5-LM) and T0. The T5-LM model further trains the T5.1.1 model for 100,000 steps (corresponding to 100B tokens) from the C4 dataset [163] on the prefix language modeling objective [72]. The T0 model was initialized from T5-LM and further trained on a multitask mixture of prompted datasets as described by Sanh et al. [31]. We repeat each experiment with T5-LM XL (hereafter **T5-XL**) and **T0-3B** as our base model. Both models use the same architecture with 2.85 billion parameters, and we used model checkpoints from Hugging Face Transformers [49]).

Target datasets. We obtain all datasets from Hugging Face Datasets[1], and cast them to the text-to-text format by applying prompt templates from the Public Pool of Prompts (P3) [79] that was used to train T0. To evaluate our few-shot methods, we utilize the same held-out datasets as T0, which cover four distinct tasks: **sentence completion** (COPA [164], HellaSwag [165], Story Cloze [166]), **natural language inference** (ANLI [167], CB [168], RTE [169]), **coreference resolution** (WSC [170], Winogrande [171]), and **word sense disambiguation** (WiC [172]). For each dataset, we randomly sample five few-shot splits from their training data, containing the same number of training examples as previous works, between 20 to 70 [108, 173]. We further divide each split into equal training and validation partitions for true few-shot learning [174](e.g. 10 train and 10 validation samples for HellaSwag). Only ANLI datasets have a publicly available test set, so for all other datasets we evaluate models on the original validation set (not utilized for few-shot training or validation).

Auxiliary datasets. We compare the performance of our methods using two sets of auxiliary data and never include any of the target datasets as part of auxiliary data. First, we use the collection of datasets used for multitask training of T0 (henceforth referred to as T0Mix), including 35 unique datasets covering question answering, sentiment analysis, topic classification, summarization, paraphrase detection and structure-to-text. Second, we utilize all datasets in P3 [79] (which forms a superset of T0Mix) and prevent data leakage by filtering out datasets that overlap with any target dataset, leading to 260 available datasets . For each auxiliary dataset, we use at most 10,000 of the dataset's examples.

[1]https://huggingface.co/datasets

Baseline methods. We compare our proposed methods with several FLAD and non-FLAD baselines. **Target-Only** (non-FLAD) directly fine-tunes the base model on the target dataset (i.e. without using auxiliary data). **Explore-Only** [10] is a commonly used FLAD method which simultaneously trains on auxiliary and target data by mixing auxiliary datasets equally. Originally called Multitask in [10], we call this Explore-Only because it is equivalent to continuously exploring auxiliary data and never exploiting knowledge of its relation to the target data. **Exploit-Only** computes gradient alignment prior to training (as in UCB1), and multitask-trains the model by mixing auxiliary datasets according to a Gibbs distribution over the alignments (similar to that in EXP3), resulting in an algorithm that exploits the relations determined prior to training, but never exploring. Both explore- and exploit-only mix target and auxiliary data with a ratio of M times the highest auxiliary sampling probability. For instance, explore-only with $M = 5$ and $\mathcal{D}_{\mathcal{A}} = \text{T0Mix}$ has a $1/35$ probability to sample auxiliary dataset $\mathcal{D}_a \in \mathcal{D}_{\mathcal{A}}$ and a $5/35$ probability for the target dataset. **Loss-Scaling** [132] is a FLAD method similar to EXP3 and UCB1; the main difference being that it scales auxiliary batch losses by their gradient alignment instead of modifying sampling probabilities. Du et al. [132] originally propose to use gradient alignment (**Loss-Scaling (GA)**), but we also propose a version that scales losses by gradient magnitude similarity (**Loss-Scaling (GMS)**).

Training details. For the target-only baseline, we use learning rates in $\{1e\text{-}4, 3e\text{-}4\}$. For all other methods, we always use a learning rate of $1e$-4. For target-, explore-, and exploit-only baselines we use batch sizes in $\{32, 128\}$. For loss-scaling, EXP3-FLAD, and UCB1-FLAD we use mini-batches of 8 samples and let G be in $\{4, 16\}$ to match the batch size of all methods. For explore- and exploit-only, we use a target dataset mixing ratio of $M \in \{1, 5, 10\}$. For all experiments we use the Adafactor optimizer [175] and validation-based early stopping for model checkpoint selection. In preliminary experiments

we consider rewards using gradients from various model partitions: the full model, encoder-only, decoder-only, and the weights of the output vocabulary matrix (language modeling head). We find that using the parameters from the language modeling head provides the best performance and contains only 2.3% of the full model parameters, significantly reducing memory consumption. For the smoothing factor, β, in UCB1-FLAD we ran preliminary experiments using values of $\{0.99, 0.9, 0.75, 0.5\}$ and found 0.9 to work well across datasets. All reported scores use $\beta = 0.9$ and we initialize auxiliary dataset rewards using 1,000 samples from each auxiliary dataset. For all experiments, we use validation-based early stopping, and train for a maximum of 10,000 gradient update steps. In practice, we find that early-stopping leads to significantly fewer than 10,000 updates, usually between 50-150 for direct fine-tuning, and 1-2,000 for other methods.

Experiment procedure. The FLAD experiment process involves training a model that is specialized for each target dataset. For each proposed method and baseline, we train and evaluate a model on each of the 11 target datasets. We repeat training and evaluation on 5 random seeds and include the aggregated results in Table 5.1. Each cell shows the accuracy averaged across all 55 (11 target datasets, 5 random seeds) experiments. This experimental process is performed for each training method on both models and auxiliary datasets.

5.5 Findings and analysis

In Table 5.1 we compare the empirical results of our MAB-based methods (EXP3-FLAD and UCB1-FLAD) and corresponding baselines on 11 target datasets. For each base model and auxiliary data combination (each column) EXP3-FLAD and UCB1-FLAD outperform all the baselines. In fact, we find that *for every single task* our

Training Method \	Base Model	T5-XL		T0-3B	
	Auxiliary Data	*T0Mix*	*P3*	*T0Mix*	*P3*
Target-Only		52.82		56.44	
Loss-Scaling [132] (GA)		53.22	55.19	59.47	60.66
Loss-Scaling [132] (GMS)		55.98	56.40	60.47	60.70
Explore-Only [10]		59.18	60.64	61.17	62.77
Exploit-Only [10]		59.79	60.49	60.87	62.87
EXP3-FLAD ($\mathcal{R}^{GA}$)		61.50	64.07	<u>62.87</u>	65.98
UCB1-FLAD ($\mathcal{R}^{GA}$)		<u>62.01</u>	<u>65.52</u>	62.35	66.29
EXP3-FLAD ($\mathcal{R}^{GMS}$)		<u>61.72</u>	<u>65.57</u>	<u>62.78</u>	65.51
UCB1-FLAD ($\mathcal{R}^{GMS}$)		61.67	<u>65.21</u>	<u>62.85</u>	66.00
EXP3-FLAD ($\mathcal{R}^{AGG}$)		<u>62.05</u>	<u>65.47</u>	<u>62.84</u>	**66.84**
UCB1-FLAD ($\mathcal{R}^{AGG}$)		**62.08**	**65.63**	**62.93**	66.29

Table 5.1: **Main results.** Each cell contains the score of training a base model (top row) with auxiliary data (second row) using the specified training method (left column), averaged across 11 target datasets on 5 random seeds (each cell is the average of 55 experiments). Target-Only does not utilize auxiliary data. **Bolded** scores are those with highest mean for a given base model and auxiliary dataset (column-wise), <u>underlined</u> scores are those where a Wilcoxon rank-sum test fails to find significant difference from the highest score ($p > 0.05$).

methods always perform equal to or better than the baselines. This demonstrates that our MAB-based methods provide a strong improvement in few-shot generalization over previous FLAD methods. For a fair comparison where each method utilizes equal data, we compare the performance of Target-Only using T0 and T0Mix (56.44) against the proposed FLAD methods and baselines using T5 and T0Mix (left column). From this comparison it becomes clear that Loss-Scaling actually does worse than multitask training followed by direct fine-tuning by 0.5-3.2%. However, we do find that the remaining FLAD methods lead to improvements (between 2.7-5.6% absolute improvement). We find small performance differences between EXP3-FLAD and UCB1-FLAD across the three reward functions. In general, $\mathcal{R}^{AGG}$ leads to the best performance, but we perform a two-sided Wilcoxon rank-sum test to check for significance between average scores and find that the other rewards frequently have no significant difference ($p > 0.05$).

The importance of prioritized sampling. Loss-Scaling was originally proposed for use with only a single auxiliary dataset and it was unclear, a priori, how it would cope with larger quantities. Additionally, Du et al. [132] purposefully choose an auxiliary dataset that is related to the target, while in our setting we make no such assumptions. We find that our methods outperform Loss-Scaling methods by 6.3% on average. In Figure 5.3 we show that, over the course of training, the value of gradient alignments and gradient magnitude similarities for most datasets will converge to 0, leading to very small gradient updates for Loss-Scaling. More importantly, *the auxiliary data that is relevant to the target task is seen less frequently for Loss-Scaling* than our MAB-based methods. This can be seen by comparing the difference in performance of Loss-Scaling methods when using less (T0Mix) vs. more (P3) auxiliary data. We find that, at best, Loss-Scaling (GA) improves 2% when using T5 and, at worst, only 0.2% for Loss-Scaling (GMS) with T0. This is compared with the notable improvements of EXP3-FLAD and UCB1-FLAD of 2.6-4% when considering the same data increase from T0Mix to P3.

The importance of exploration *and* exploitation. Interestingly, we expected that Exploit-Only would outperform the Explore-Only method because it utilizes relational information between the target and auxiliary tasks, but find no statistical difference between the methods (two-sided Wilcoxon rank-sum test gives $p > 0.05$). Furthermore, when comparing the ability to leverage additional auxiliary data (i.e. going from T0Mix to all of P3), we find that the improvement for Explore- and Exploit-Only methods is minimal with only 0.7-2% improvement. On the other hand, EXP3-FLAD and UCB1-FLAD show a notable improvement of 2.6-4%, emphasizing the importance of both exploration *and* exploitation, particularly when dealing with large collections of auxiliary data.

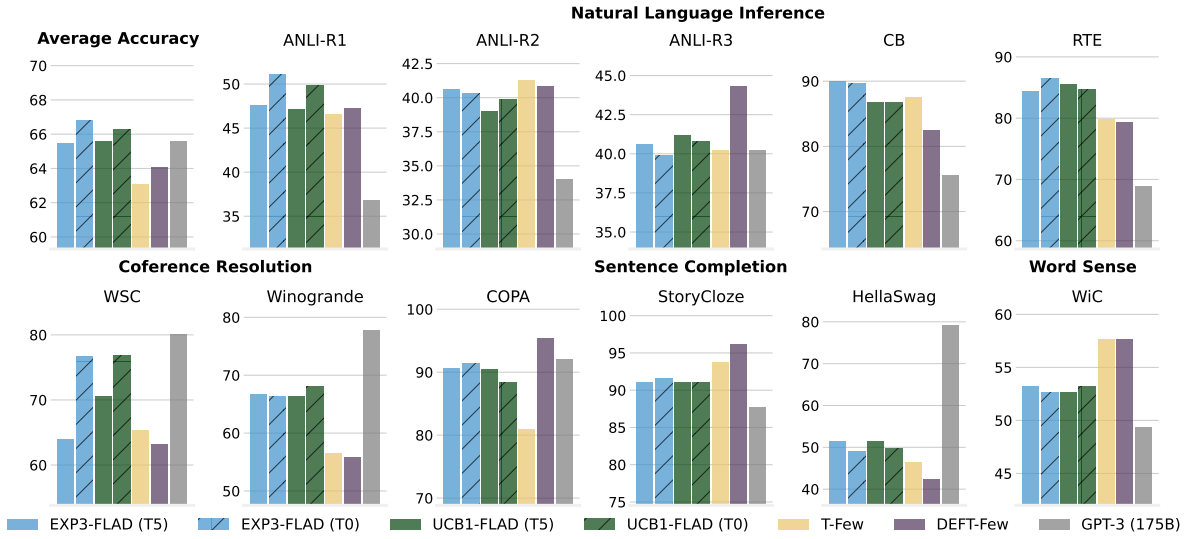

Figure 5.2: **Comparison of state-of-the-art few-shot methods with FLAD methods trained on P3 using $\mathcal{R}^{\mathbf{AGG}}$.** T-Few scores are from [173]. DEFT-Few scores are from [148]. GPT-3 scores are from [108] and utilize few-shot in-context learning. All models utilize the same number of few-shot examples and (other than GPT-3) have 3B parameters.

FLAD provides improved generalization over non-FLAD methods. Next, we compare the performance of our best models trained on P3 using $\mathcal{R}^{AGG}$ with state-of-the-art few-shot methods: T-Few, DEFT-Few, and GPT-3. T-Few [173] is a variant of the T0-3B model that multi-task pre-trains parameter-efficient $(IA)^3$ modules followed by target-only fine-tuning of the $(IA)^3$ modules. DEFT-Few [148] is a variant of the T5-XL model that uses retrieved auxiliary data for multi-task training. It first trains a T5-XL model on the 500 nearest neighbor samples from P3 using 1000 unlabeled target dataset samples, and then performs few-shot target-only fine-tuning with the $(IA)^3$ modules from Liu et al. [173]. Finally, we also compare against the 175 billion parameter variant of GPT-3 [108], which utilizes in-context learning. We find that, on average, models trained using our FLAD-based methods outperform all other methods and, to the best of our knowledge, our methods lead to the first 3 billion parameter model that outperforms

GPT-3 on this dataset mixture (previous smallest models have 11 billion parameters), despite using 62.5 times fewer parameters than GPT-3. Additionally, we find that our FLAD-based methods provide robust performance across datasets, achieving the best or second-best performance on 8/11 datasets, and never performing worst. The use of task-specific modules lead T-Few and DEFT-Few to significant improvements over target-only fine-tuning, preventing the models from ending up in poor local minima. However, these results demonstrate that with the same data, simultaneously fine-tuning with auxiliary and target data leads to improved few-shot generalization, providing a complementary means of improving performance.

Investigating the Reward-Generating Processes. In Section 5.3.2, we mention that due to the highly non-convex loss landscape and the use of stochastic gradient descent-based optimization techniques, we cannot ensure that our reward generating process is stationary, independent across auxiliary datasets, or follows a normal distribution. To gain a deeper understanding of our reward-generating processes, we examine the distribution of each reward using 5,000 samples from all 35 auxiliary datasets of T0Mix and 32 samples from a few-shot target dataset, WSC [170]. The resulting histograms at every 100 steps can be found in Figure 5.3. The left side of Figure 5.3 demonstrates that for $\mathcal{R}^{GA}$, almost every dataset yields a Gaussian reward distribution, with a few multi-modal distributions. Notably, WikiBio [176] (dark orange) exhibits peaks at 0.25 and -0.75. Interestingly, $\mathcal{R}^{GA}$ results in polarized rewards across datasets, with minimal distribution density between -0.75 and 0.25. In contrast, the right side of Figure 5.3 displays more non-Gaussian distributions for $\mathcal{R}^{GMS}$, as well as flatter distributions compared to $\mathcal{R}^{GA}$. Remarkably, we observe that $\mathcal{R}^{GA}$ produces more stationary reward distributions, as the distribution for almost every dataset (30/35) converges rapidly towards 0 after only 100 steps. Although most distributions for $\mathcal{R}^{GMS}$ also converge towards 0, the convergence occurs at a slower

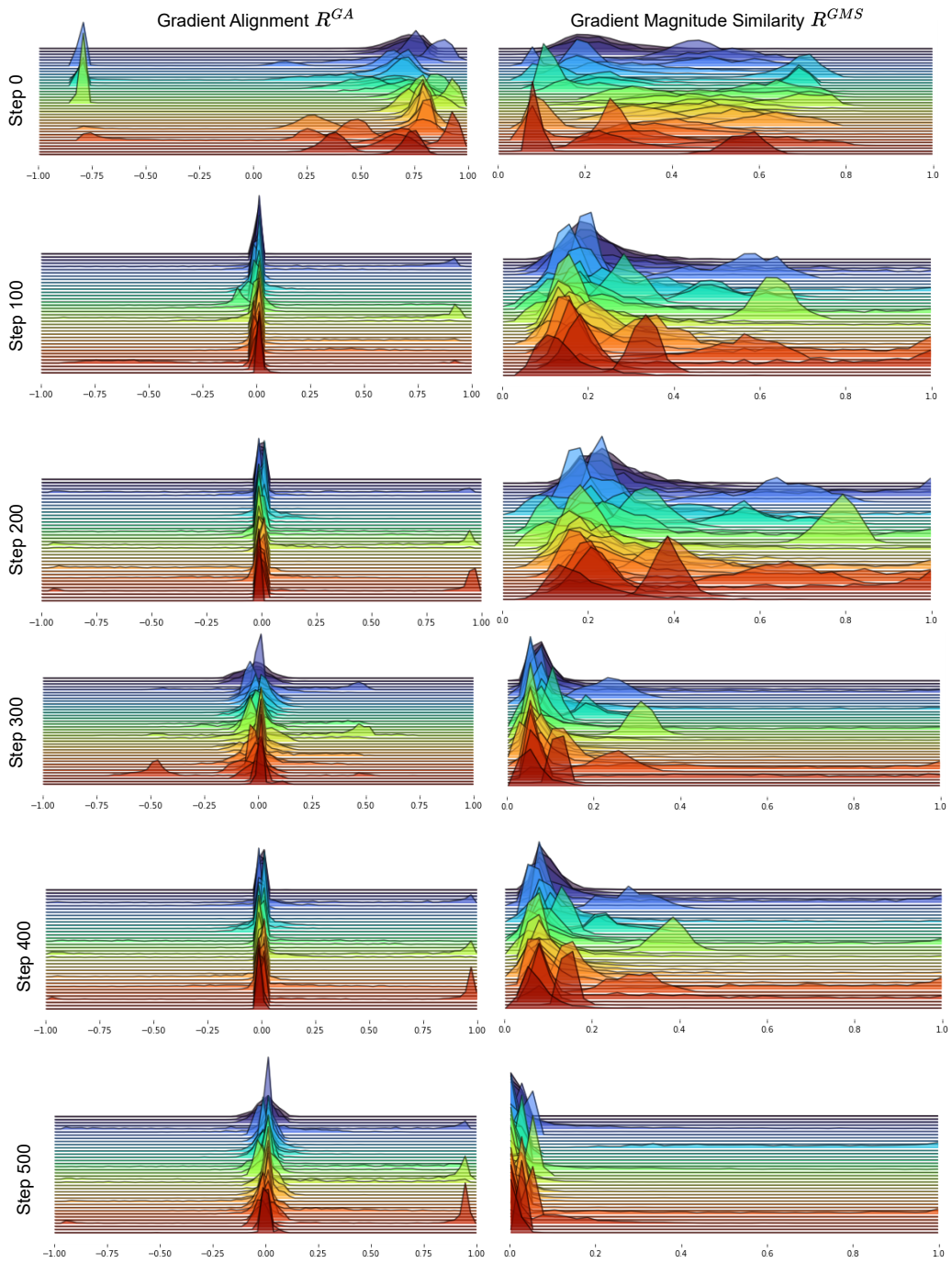

Figure 5.3: **Reward distributions** of R^{GA} and R^{GMS} prior to training (step 0) and every 100 gradient updates thereafter for the T5-XL model with T0Mix as the auxiliary dataset and WSC [170] as the target dataset. Each histogram shows the reward distributions for all 35 auxiliary datasets. By step 300 most auxiliary datasets provide 0 reward, while only the few remaining "beneficial" datasets provide positive rewards.

pace, taking nearly 500 steps.

Probing the training dynamics. To better understand the training dynamics of our proposed methods, we perform a case study on T5-XL with T0Mix and $\mathcal{R}^{GA}$ and find two datasets where either algorithm improves significantly over the other . First, we study RTE, where UCB1-FLAD outperforms EXP3-FLAD. We calculate the empirical distribution of samples seen from each auxiliary dataset and find that EXP3-FLAD samples nearly uniformly from all datasets while UCB1-FLAD forms a bimodal sampling distribution with peaks at 2.5% and 3.25% (30% relative difference). The uniformity of the EXP3-FLAD distribution is counterintuitive, as we do find that it achieves separation between auxiliary tasks in the cumulative estimated reward , but this does not lead to separation in the sampling probability space. Additionally we find that even on COPA, where EXP3-FLAD outperforms UCB1-FLAD, EXP3-FLAD still achieves good separation between cumulative estimated rewards, but has a unimodal sampling distribution, while UCB1-FLAD does not have as clear of a bimodal distribution as in RTE. The difference in empirical sampling distributions is likely due to the difference between the greedy policy of UCB1-FLAD and the stochastic policy of EXP3-FLAD. Empirically, we find that EXP3-FLAD very rarely assigns an auxiliary dataset a probability $< 1\%$, leading to many "bad" batches over the course of thousands of turns. On the other hand, the optimistic policy of UCB1-FLAD spends much less time exploring and will sample "bad" batches much less frequently.

The effect of scaling $|\mathcal{A}|$. To assess the scalability of our proposed methods, we conduct a study by varying the size of $|\mathcal{A}| \in \{35, 75, 125, 175, 225, 260\}$. For each value of $|\mathcal{A}|$, we consistently include the 35 datasets from T0Mix and randomly select additional auxiliary datasets from P3 until we reach the desired $|\mathcal{A}|$. The study is performed on the

same 11 target datasets as the main study, using the T0 base model and $\mathcal{R}^{AGG}$ reward. The experiment is repeated with three random seeds. Figure 5.4 shows the mean across the 11 target datasets, along with the standard deviation between seeds.

We find that both EXP3-FLAD and UCB1-FLAD experience a sharp increase from $|\mathcal{A}| = 35$ to 75. In addition, we observe improvements up to the maximum value of $|\mathcal{A}| = 260$, ultimately improving accuracy by 2.54 for EXP3-FLAD and 3.12 for UCB1-FLAD when transitioning from 35 to 75 datasets, with further increases of 1.54 for EXP3-FLAD and 0.47 for UCB1-FLAD when increasing $|\mathcal{A}|$ from 75 to 260.

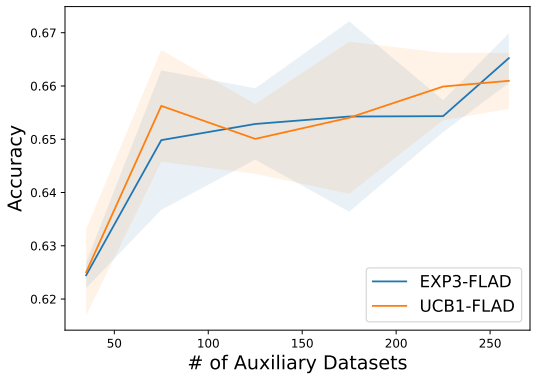

Figure 5.4: **The effect of scaling $|\mathcal{A}|$ on target task performance**. Each line represents mean score across 11 datasets and three random seeds, with shaded regions falling between one standard deviation of the mean.

5.6 Discussion

Discussion on reward functions. In FLAD we want to prioritize training on auxiliary datasets with similar solution spaces as the target task without overfitting to the few-shot target data, and our reward functions are designed to serve this goal. To better understand the reward signal of our aggregate reward, $\mathcal{R}^{AGG}$, we examine four combinations of rewards: low $\mathcal{R}^{GA}$ and $\mathcal{R}^{GMS}$, high $\mathcal{R}^{GA}$ but low $\mathcal{R}^{GMS}$, low $\mathcal{R}^{GA}$ but high $\mathcal{R}^{GMS}$, and high $\mathcal{R}^{GA}$ and $\mathcal{R}^{GMS}$. When both rewards are high, we can assume that the auxiliary gradient is useful. However, when one reward is high and the other is low, it is difficult to draw conclusions as a high $\mathcal{R}^{GA}$ on its own means the auxiliary gradient will update weights in

the right direction, but low $\mathcal{R}^{GMS}$ can mean that we significantly overshoot *or* undershoot the target, where overshooting can be much more detrimental than undershooting. If both $\mathcal{R}^{GA}$ and $\mathcal{R}^{GMS}$ are small, we know the auxiliary gradient leads us away from the target solution space, but we don't know if its magnitude is much larger or smaller than the target. At the beginning of training, we can't know if the target or auxiliary gradient has larger magnitude, but as training progresses, it becomes significantly more likely that the auxiliary gradient is greater than the target. Thus, when both $\mathcal{R}^{GA}$ and $\mathcal{R}^{GMS}$ are low, we are likely to be pulled far from our current solution.

This study uses training set-based rewards, but validation set-based rewards are also possible. One downside of validation-based rewards is they calculate validation score frequently, which increases computational complexity. Additionally, we focus on the few-shot setting and use validation-based early stopping. If we use a validation-based reward, then to prevent overfitting we will need to further split the data into 3 partitions: train, early-stopping validation, and reward-validation.

Choice of baselines. With respect to the number of auxiliary datasets $|\mathcal{A}|$ and target datasets $|\mathcal{T}|$, our methods and the baselines we compare against have a computational complexity of $O(|\mathcal{T}|)$, independent of $|\mathcal{A}|$. For our model and these baselines, the models we train require ~ 6 GPU-hours per target dataset. If we were to consider a baseline whose computation grows linearly w.r.t. $|\mathcal{A}|$, $O(|\mathcal{A}||\mathcal{T}|)$ (e.g. [140]), these experiments would not be feasible without a large amount of hardware: *Training such a model with T0Mix would take over 200 GPU-hours (over 8 GPU-days) for a single target dataset*, and over 1500 GPU-hours (*over 2 GPU-months*) when using all of P3.

How does FLAD relate to few-shot learning and multitask learning? Both few-shot learning and FLAD are concerned with optimizing model performance on a

single target task with a limited number of examples from the target task. In few-shot learning, the model is given only the target task data $\mathcal{D}_{\mathcal{T}}$ and there is no auxiliary data. Effectively, $\mathcal{D}_{\mathcal{A}}$ is the empty set for few-shot learning. In contrast, for the FLAD setting $|\mathcal{D}_{\mathcal{A}}| > 1$. Based on the findings from this study, we highly recommend that practitioners utilize auxiliary data when it is available.

Multitask learning is concerned with optimizing a model for performance on multiple target datasets simultaneously. This is in direct opposition with the FLAD methods presented here, which aim to optimize a model for a single target task. However, it is possible to extend our MAB-based methods to optimize for multiple target tasks simultaneously by aggregating multiple rewards. We believe this would make for an interesting future study.

Limitations. One of the implicit assumptions in the FLAD setting (made in this chapter and all prior works) is that there is at least *some* auxiliary data that will be useful for the target task. However, one of the main distinctions of our methods from prior works in the FLAD setting is that prior works make a strong assumption that all auxiliary data are useful, and thus appropriate auxiliary datasets must be hand-picked by humans. On the other hand, our methods allow for only a small portion of the auxiliary data to be useful – our proposed algorithm explores to find useful auxiliary datasets and then exploits them.

Where else can MAB-Based FLAD methods be applied? The methods proposed in this chapter can be applied in a variety of other settings. Due to the similarities of the multitask setting and the FLAD setting, the proposed methods can be applied in any setting which has a plethora of labeled data for non-target tasks and a limited amount of data for the target task. For instance, our MAB-based FLAD methods can be applied in computer vision [93, 105], multimodal [177, 178], and multilingual [179, 8, 180] settings.

Furthermore, because these methods rely only on gradients, and not on any features specific to language, it should also be possible to extend these methods into vastly different domains, such as robotics [181, 182] or time-series forecasting [183, 184, 185] to further improve their generalization capability to low-resource situations.

Chapter 6

Improving Language Model Pretraining, Efficiently

Modern large language models are trained on data from a variety of domains (e.g. GitHub, Wikipedia, books, web text). Prior works have demonstrated that the exact mixture proportion of each domain in the training mixture can greatly impact the model's performance [186]. Additionally, pretraining large language models is computationally, and fiscally, very expensive. For example, BLOOM [187], a 176-billion parameter model, was trained for 1,082,990 GPU-hours (on 80Gb A100 GPUs).

In this chapter, we focus on improving the data efficiency and performance of pretrained language models by selecting a better training data mixture. Motivated by the success of multi-armed bandits in Chapter 5, we view each data domain as the arm of a multi-armed bandit, and design a reward that aims to maximize the new information content of future training data. We show that not only does this formulation of data mixing lead to improved performance, but can significantly improve data efficiency, potentially reducing costs of training large models in the future.

6.1 Introduction

It is well-known that the training data for machine learning models has a significant influence on their performance. In particular, the data used to pretrain large language models (LLMs) can be a major factor in the performance of a given LLM. For example, the 28 different 7-billion parameter models on the Open LLM Leaderboard[1] have scores varying from 34.92 to 56.26 even though they all use nearly the same model architecture and training process [188]. It is a widely accepted view that *pretraining is performed so that models can absorb large quantities of information* [108, 189, 190], and later training stages such as target task fine-tuning [3], instruction fine-tuning [191], and RLHF [192] primarily refine the model for a specific purpose. This perspective raises the important question of how best to choose pretraining data for training LLMs.

Language models are generally trained on data collected from a variety of domains in hopes that data diversity will lead to a higher-quality model, but the data mixing strategy to use (i.e. how frequently to sample data from each domain) during training is an open question. For example, when introducing The Pile [193] dataset (consisting of data from 22 domains), the authors suggest higher sampling weights on academic texts and those domains that they felt would provide high-quality data, but these weights are determined using intuition and heuristics, raising the question as to whether a more performant set of weights could be found. The recently proposed DoReMi algorithm [186] was specifically designed to automatically determine a data mixing strategy for LLM training. DoReMi optimizes domain weights that maximize the information gained of a "proxy" model over a "reference" model, but requires training multiple models, reducing the method's efficiency. Additionally, we show in this chapter that their sampling weights don't transfer well across models and thus requires training new "reference" and "proxy" models in order to

[1]Open LLM Leaderboard accessed on 10/02/2023, 28 models includes only pretrained models without fine-tuning, instruction-tuning, or RL-tuning.

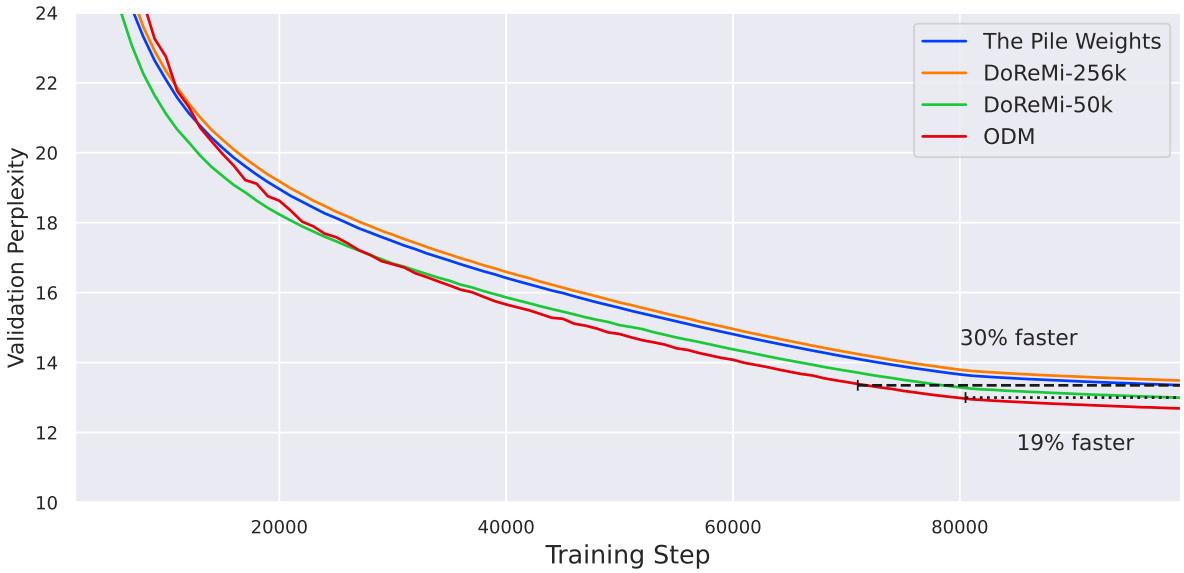

Figure 6.1: **Validation perplexity**, unweighted average over 22 domains from The Pile [193].

determine the best weights for each new model architecture or tokenizer. These additional steps and considerations reduce the effective efficiency of DoReMi and further increase the already expensive cost of training large language models. Furthermore, both DoReMi and The Pile fix weights throughout training and therefore cannot adapt to changing dynamics over the course of pretraining.

In this chapter, we follow the principle that the best data to train on is the data that maximizes information gained and that a data selection method should introduce negligible computational overhead. Motivated by the success of multi-armed bandits (MAB) for auxiliary data selection in few-shot LLM fine-tuning in the previous chapter, we view each data domain as the arm of an MAB and design an algorithm that optimizes the data mixing distribution in an online fashion, thereby adapting to changing training dynamics. Recalling from information theory that perplexity can be thought of as a measure of model uncertainty and the expected information gain from learning the next token, we aim to increase the mixing ratio for domains with the most information to be

learned. We therefore utilize the training loss per domain as a reward for our multi-armed bandit algorithm, which fortuitously requires minimal overhead to compute.

6.2 Online Data Mixing (ODM)

In this section, we first define the setting under which online data mixing for language model pretraining takes place (outlined in Figure 6.2). Then, we formulate the online data mixing problem under the multi-armed bandit (**MAB**) setting, and describe our reward function which measures information gain and is very efficient to compute. Finally, we describe our algorithm for ODM and present pseudo-code in Algorithm 4.

Problem setup. Consider the setting where we are given K groups of data for language model pretraining, where each group $\mathcal{D}_i$ will be sampled according to the probability defined by $\pi(\mathcal{D}_i)$. Each group $\mathcal{D}_i$ could be assigned explicitly according to different domains as in The Pile [193], or they could determined via some automatic method (as e.g. in [194]). In traditional data mixing, each $\pi(\mathcal{D}_i)$ is fixed prior to training, but in online data mixing, we let each $\pi(\mathcal{D}_i)$ be redefined at every training iteration. Given that we want to update $\pi(\mathcal{D}_i)$ at every training iteration, the problem this chapter attempts to solve is how to update $\pi(\mathcal{D}_i)$ so that the information content of the data being trained on is maximized, and how to do so efficiently.

Adapting multi-armed bandits to data mixing. We adopt the multi-armed bandit (**MAB**) framework to attack the online data mixing problem by formulating it as a Markov decision process [153] that is played over N turns. We design our approach based on Exp3 (*Exp*onential-weight algorithm for *Exp*loration and *Exp*loitation) [143]. Exp3 defines the policy as a Gibbs distribution based on the empirically determined importance-weighted reward of dataset proportions [159] and allows for exploration by

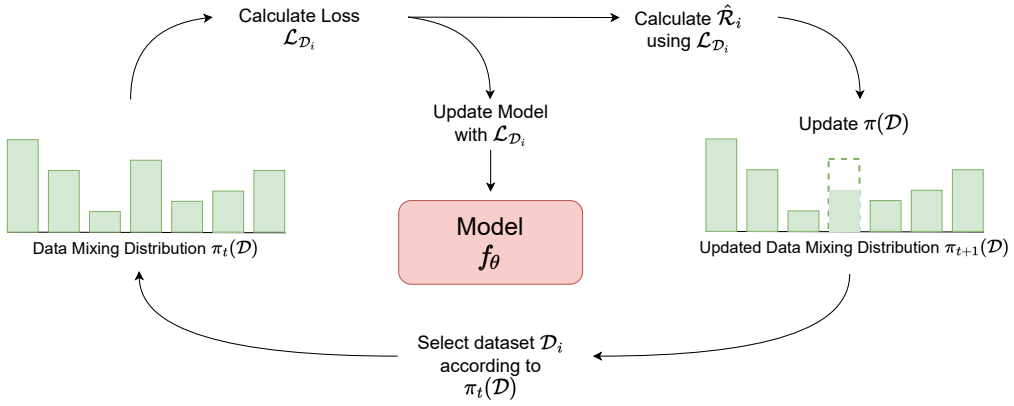

Figure 6.2: **Overview of Online Data Mixing (ODM) as a multi-armed bandit**. At each iteration of training, t, a dataset $\mathcal{D}_i$ is sampled according to the data mixing distribution π. The loss $\mathcal{L}_{\mathcal{D}_i}$ is calculated w.r.t the model f_θ and subsequently used to update the model. Simultaneously, a reward $\hat{\mathcal{R}}_i$ is calculated and used to update π for the next iteration, $i+1$.

mixing the Gibbs distribution with a uniform distribution [143]. Let $\mathcal{E}_t$ represent the exploration rate at time step t, then the probability of selecting dataset $\mathcal{D}_i \in \mathcal{D}$ is defined by π as the linear combination of Gibbs and uniform distributions

$\pi_t(\mathcal{D}_i) = (1 - K\mathcal{E}_t)\frac{\exp(\mathcal{E}_{t-1}\hat{R}_i)}{\sum_j \exp(\mathcal{E}_{t-1}\hat{R}_j)} + \mathcal{E}_t$ where $\hat{R}_{i,t}$ is the moving average of the importance weighted reward $\hat{R}_{i,t} = \alpha\hat{R}_{i,t-1} + (1-\alpha)\frac{R_{i,t}}{\pi_{t-1}(\mathcal{D}_i)}$. We adopt the decaying exploration rate from Seldin et al. [159], defined at turn t as $\mathcal{E}_t = \min\left\{\frac{1}{K}, \sqrt{\frac{\ln K}{K \cdot t}}\right\}$. The main deviation of our method from Exp3 is the use of a moving average estimated reward instead of a cumulative estimated reward. Under normal MAB settings, rewards at each turn are weighted equally, but in our setting we care most about recent rewards. Thus, we still account for past rewards through the use of a moving average, but rewards from the past are weighted less and less moving further into the past.

Designing the reward function. When designing our reward function we have 2 main goals: (1) ensure that the policy favors data with the highest information content, and (2) minimize the computation required. To achieve these goals, we define the reward

to be the current loss for a given dataset group. Formally, at turn t, suppose that dataset $\mathcal{D}_i$ is sampled from $\pi(\mathcal{D})$, and a batch is sampled as $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_i$. Then, the reward is simply $\mathcal{R}_{i,t} = \mathcal{L}(f, \mathbf{x}, \mathbf{y})$. By formulating the reward as the training loss on a dataset, we add no additional forward or backward passes through the model beyond standard training procedures, minimizing the computation required. Additionally, as discussed in section 6.1, perplexity (the exponentiated loss) is a measure of expected information gain from each token in a sequence. Thus, by assigning a high reward to datasets with high perplexity, we favor data with the highest information content.

Algorithm 4 Online Data Mixing (ODM)

Require: $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$: Grouped dataset
Require: f_θ: Parameterized model
Require: $\mathcal{L}$: Loss function
Require: G: Gradient accumulation steps

Initialize: $K = |\mathcal{D}|$; $\mathcal{E}_0 = \frac{1}{K}$; $\forall i \in \{1, \ldots, K\} : \hat{R}_i = 0$
for $t = 1, 2, \ldots, N$ **do**
  $\mathcal{E}_t = \min\left\{\frac{1}{K}, \sqrt{\frac{\ln K}{K \cdot t}}\right\}$ ▷ Update the exploration rate
  $\pi(\mathcal{D}) : \pi(\mathcal{D}_i) \leftarrow (1 - K\mathcal{E}_t)\frac{\exp(\mathcal{E}_{t-1}\hat{R}_i)}{\sum_j \exp(\mathcal{E}_{t-1}\hat{R}_j)} + \mathcal{E}_t$ ▷ Calculate the mixing distribution
  $\forall i = 1, 2, \ldots, K : \mathcal{L}_{\mathcal{D}_i} = 0$ ▷ Reset group losses
  for $g = 1, 2, \ldots, G$ **do**
    Sample $\mathcal{D}_i \sim \pi(\mathcal{D})$ and sample a batch $\{\mathbf{x}, \mathbf{y}\}$ from $\mathcal{D}_i$
    $\mathcal{L}_{\mathcal{D}_i} \leftarrow \mathcal{L}_{\mathcal{D}_i} + \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$ ▷ Record group losses for reward updates
  end for
  Update model parameters w.r.t $\sum_i \nabla_\theta \mathcal{L}_{\mathcal{D}_i}$
  for $i \in \{1, \ldots, K\}$ where $\mathcal{L}_{\mathcal{D}_i} \neq 0$ **do**
    $\hat{R}_i \leftarrow \alpha \hat{R}_i + (1 - \alpha)\mathcal{L}_{\mathcal{D}_i}$ ▷ Update estimated rewards
  end for
end for

Online data mixing algorithm. Our algorithm is shown in pseudocode in Algorithm 4 and runs as follows: At each turn, the exploration rate $\mathcal{E}_t$ is calculated and the policy π defines a sampling strategy over all K datasets $\mathcal{D}_i \in \mathcal{D}$. Since we are dealing with LLM pretraining which typically uses a large batch size, we assume that we will have G gradient accumulation steps. For each accumulation step we sample one of the datasets

$\mathcal{D}_i$, then sample a batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_i$ and calculate the loss $\mathcal{L}_{\mathcal{D}_i}$. After accumulating losses, we calculate the gradient w.r.t. θ and update the model. Finally, for each sampled dataset $\mathcal{D}_i$, we calculate a reward $\mathcal{R}_i$ that is used to update the policy π for the next turn. As a practical method to reduce the very high variance of losses at the beginning of language model training, we include a warmup period during which the model trains, but the policy remains stationary. In practice, we find a warmup period of 1% of total steps to be sufficient.

6.3 Experimental Setup

Training. For our experiments we use The Pile [193], an 825Gb open-sourced language modelling dataset comprising 22 smaller datasets from various domains including Wikipedia, Github, and PubMed Central. We train decoder-only style transformers using an adapted version of the GPT-NeoX library [195]. For all experiments, we train a 1 billion parameter model using the model configuration of Pythia [196]. We explore values of $\alpha \in \{0.25, 0.5, 0.75, 0.9\}$ in preliminary experiments, and let $\alpha = 0.5$ for all the experiments shown here as this was marginally better than the other values. We train using a batch size of 60 sequences per GPU, and accumulate gradients across 8 GPUs in parallel ($G = 8$) to reach a total batch size of 480 samples. We let the sequence length be 1024 and pack sequences together [197]. We train for a total of 100,000 steps, reaching 50 billion tokens. For ODM, we initialize the domain weights using those defined by The Pile.

Our 1-billion parameter model uses a sequence length of 1024, has 16 layers with a hidden size of 2048, 16 attention heads, and rotary positional embeddings [198]. We use FlashAttention [199] to reduce training time. We use the Adam optimizer [123] with a linear warmup over 1000 iterations from a minimum learning rate of 2.5e-5 to a maximum

learning rate of 2.5e-4, and then decay the learning rate with a cosine schedule down to the minimum of 2.5e-5 again. We use the GPT-NeoX-20B tokenizer [200].

Evaluation. To validate the performance of our approach and the baselines, we compute perplexity on held-out validation and test data from each domain of The Pile. Additionally, we evaluate each model on downstream capabilities by performing multiple choice classification on the 57 tasks from MMLU [201]. For each task in MMLU we use 5 in-context examples.

Baselines. We compare the performance of our method against that of the original domain weights suggested by The Pile [193], and refer to it as The Pile Weights (**TPW**). Additionally, we compare with the domain weights proposed by DoReMi [186], but empirically find that the weights do not perform as published. However, after discussion with the authors, we attained weights that were re-calculated on the same tokenizer as ours[2]. The original DoReMi weights are computed with a 256k vocabulary tokenizer while we use a 50k vocabulary tokenizer, so to specify each DoReMi baseline we name them **DoReMi-256k** and **DoReMi-50k**.

6.4 Findings and analysis.

In Figures 6.1 and 6.3 we compare the perplexities of training models using ODM with the baseline data mixing methods. Table 6.1 shows the average 5-shot accuracy on MMLU of ODM and baseline methods.

Main results. Figure 6.1 shows that ODM achieves the final performance of the originally suggested Pile weights (TPW) with 30% fewer iterations, and 19% fewer than

[2]It is hypothesized by the authors of [186] that different tokenizers may lead to different domain weights, but is still an open question why that may be the case.

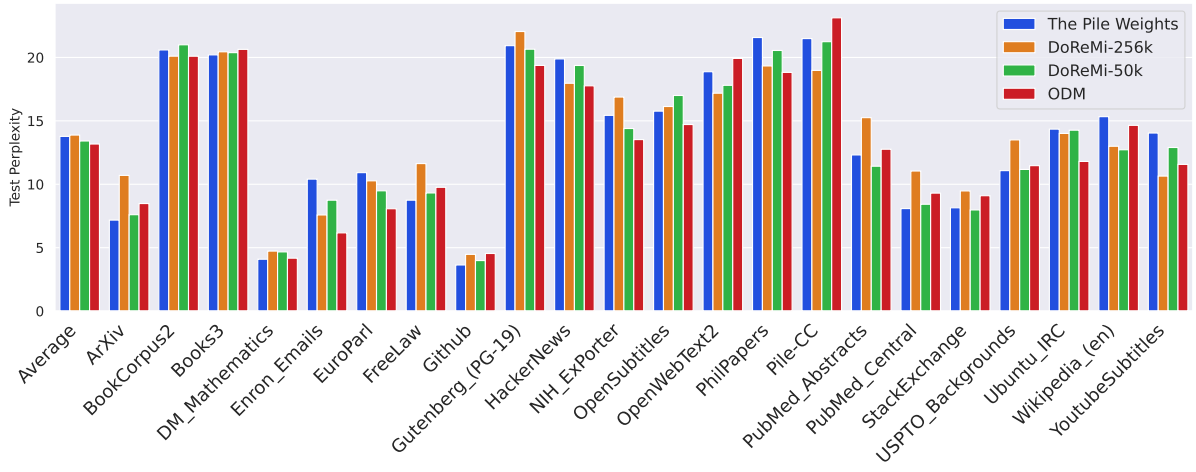

Figure 6.3: **Test perplexity** on average, and on 22 individual domains.

DoReMi-50k. Additionally, Figure 6.1 shows that ODM's final validation perplexity is 4.8% lower than TPW, 2.4% lower than DoReMi-50k, and 4.9% lower than DoReMi-256k, emphasizing how the DoReMi method is not transferrable across models. These results show that ODM improves the training efficiency compared with static data mixing methods. Additionally, Table 6.1 shows that ODM leads to better downstream performance in 5-shot classification tasks, improving over TPW by 3%, and DoReMi-50k by 1.9%.

Figure 6.3 shows the test perplexity of each method on held-out data as well as the average perplexity. Surprisingly, we find that the original domain weights reported for DoReMi [186] (DoReMi-256k) leads to test perplexity that is, on average, 0.7% worse than The Pile Weights, in direct contradiction with their original findings. However, DoReMi-50k does improve over The Pile Weights by 2.6%, demonstrating that the domain weights determined by the

Method	Accuracy
The Pile Weights	0.27469
DoReMi-256k	0.27596
DoReMi-50k	0.27887
ODM	0.28416

Table 6.1: **Average 5-shot accuracy on MMLU**

DoReMi method do not transfer well across models.

The effects of data mixing optimization objectives on individual domain performance. Here we compare the empirical effects of the contrasting optimization of objectives of ODM and DoReMi on individual domains. Recall that the reward function used in ODM favors dataset groups with the greatest information gain (highest loss) at each step, and that DoReMi's objective is to maximize the information gain of a "proxy" model over a "reference" model (i.e. "minimize the worst-case excess loss"). To see these different objectives in effect, we group the performance of each method into one of three buckets: best, worst, or in the middle, where the ideal method would have all 22 domains in the "best" category. Interestingly, we find that The Pile Weights are almost evenly distributed across all 3 buckets, doing worst in 7 domains, best in 7, and in the middle for the remaining 8. As expected from a method that optimizes for the best worst-case scenario, we find that DoReMi-50k's test perplexity is often not the best or the worst, but falls in the middle. In fact, 17/22 domains are in the middle, only performing best on three domains (PubMed_Abstracts, StackExchange, and Wikipedia_(en)), and worst on only two domains (BookCorpus2 and OpenSubtitles). On the other hand, using ODM leads to the best perplexity on 9 domains, with 9 more in the middle, and only performing the worst on 4 domains (Books3, Github, OpenWebText2, and Pile-CC). Notably, two of the domains where ODM performs worst are web text domains but this decreased performance does not seem to have a negative impact on downstream performance.

What does ODM's sampling policy look like? In Figure 6.4 we show the cumulative sampling distribution of each domain over the course of training. Note that ODM is initialized with The Pile Weights, which are the initial values on the left. Figure 6.4 highlights the three datasets whose mixing ratio increased the most (PhilPapers, Hack-

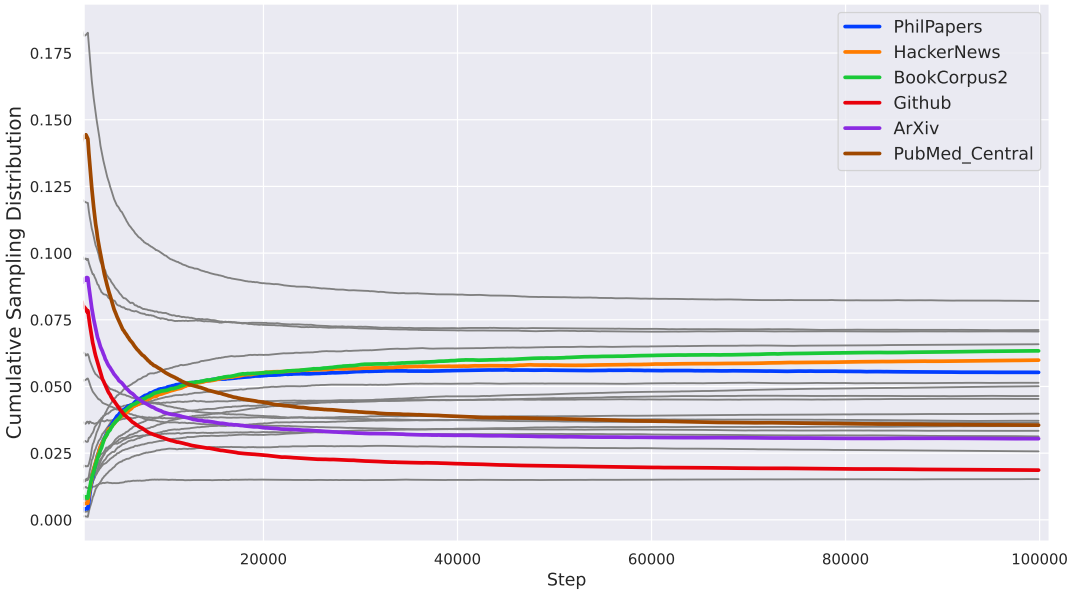

Figure 6.4: **The cumulative sampling distribution of ODM** calculated as the samples per domain out of the total number of samples trained on. Highlighted lines are the six domains whose final sampling distributions have increased/decreased the most from initialization.

erNews, and BookCorpus2), and the three datasets whose mixing ratio decreased the most (Github, ArXiv, and PubMed_Central). It is evident from this figure that ODM finds a sampling distribution which is closer to uniform than The Pile Weights. We also see that the distribution for most domains stabilizes early on in training ($\sim$ 40000 iterations). Beyond the 40000 step, the distribution is still changing, but at a much lower rate. For example, we see that the mixing ratio for Github is still decreasing and the ratio for both BookCorpus2 and HackerNews are increasing all the way until the end of training.

Why does ODM's validation perplexity start off high? Figure 6.1 shows that although our method outperforms the baselines, at the beginning of training ODM actually has higher perplexity than other methods. We believe that this is due to the homogeneity of the micro-batches used in ODM, whereas other methods see a greater mixture of data in each batch. In preliminary experiments we trialed a version of ODM that uses data

from a single domain in all gradient update steps, and found that this exacerbates the phenomena leading to a perplexity that starts even higher. This suggests that one of the weaknesses of our method is the requirement that each batch comes from the same grouped dataset. This problem can be alleviated by decreasing the micro-batch size, but this comes with technical considerations as simply decreasing micro-batch size will reduce GPU utilization, and lead to slower wall clock time. Likely, a better solution would be to mix domains within micro-batches during the warm-up phase, which would lead to validation perplexity exactly the same as The Pile Weights, but gaining the advantages of ODM after the warm-up.

Limitations and Future Directions Some prior studies have found that adding code data to pretraining can lead to improved reasoning within models [202], but we find that ODM heavily downweights the GitHub domain. Why is this, and what can we do about it? Firstly, the low reward found from the GitHub domain is likely due to the limited number of tokens used in code data, leading to an implicitly lower perplexity, rather than code being less informative than other domains. This is one inherent limitation of measuring information gain based on tokens, which our method does not currently overcome. Our method does not directly calculate information gain, but rather the perplexity of each domain. For reference, given a sequence of T tokens, the information gain of the last token t_T is calculated as $\text{IG}(t_T, \{t_1, \ldots, t_{T-1}\}) = H(t_T) - H(t_T|\{t_1, \ldots, t_{T-1}\}$. If we were to estimate the entropy of each domain as $H(D)$, then we can estimate the average information gain on a specific sample, using the estimated entropy and the empirical conditional entropy of a sequence, as $\frac{1}{T}\Sigma_{i=1}^{T}\text{IG}(t_i|\{t_1, \ldots, t_{i-1}\}) = H(D) - \frac{1}{T}\Sigma_{i=1}^{T}H(t_i|\{t_1, \ldots, t_{i-1}\})$. Of course, the additional estimation of $H(D)$ will incur an efficiency loss compared with the current method, so there will need to be a trade-off between adding compute and the potential performance gains.

Next, while our ODM method shows promise, it does not take into account any specific downstream use cases. For example, if it is known ahead of time that the model being trained will be used to generate scientific articles, then it will likely be useful to spend more compute time on ArXiv and PubMed scientific articles. However, our method purposefully downsamples these domains compared with the original pile weights.

Furthermore, ODM does not explicitly take into account the quantity of data in each domain, which could lead to some domains being repeated many times, while others still have not been fully trained. In theory, our reward function implicitly takes this into account by assigning a lower information gain (reward) to domains which are repeated if their data distributions have been learned by the model. However, in the work, we do not explicitly test for this setting. Previous works have found that repeating data up to four times can lead to performance improvements similar to fresh (unseen) tokens [203]. Nonetheless, our method only guarantees that informative data is shown to the model, but not necessarily new data.

In all, this work provides a good stepping stone for future improvements. Future methods can combine some of the points discussed here, where the method use our very efficient online reward, combined with a quantity- and heuristically determined mixing weight. For example, if the goal of the model is to perform high quality reasoning, code data can manually be upweighted according to some heuristically determined weights, in combination with an additional weighting that considers the quantity of data in each sub-domain of code data. Finally, some works have found that mixing supervised instruction-tuning data into the pretraining can significantly improve models performance [96]. How exactly to mix in the supervised data is certainly an open area of research, and all these ideas in combination leave much room for future work.

Chapter 7

Improving Cross-Linguality for Open-Retrieval Question Answering

7.1 Introduction

One challenge of emergent domains is that the originating locality is unknown, leading to the need for reliable information to cross language barriers. However, it is unlikely that domain-specific information will be available across multiple languages for a new domain. Furthermore, information rapidly changes in emerging domains, compounding the challenge of accessing credible data.

An example of a prominent emergent domain is COVID-19, which quickly spread across the globe. To combat the spread of misinformation about COVID-19, researchers have developed open-retrieval question answering [204] systems which use large collections of trusted documents. For example, Lee et al. [205], Levy et al. [206], and Esteva et al. [207] develop open-retrieval QA systems using large corpuses of scientific journal articles. However, because these systems focus on English, they leave a gap for implementation on emergent domains that do not originate in English-speaking locations.

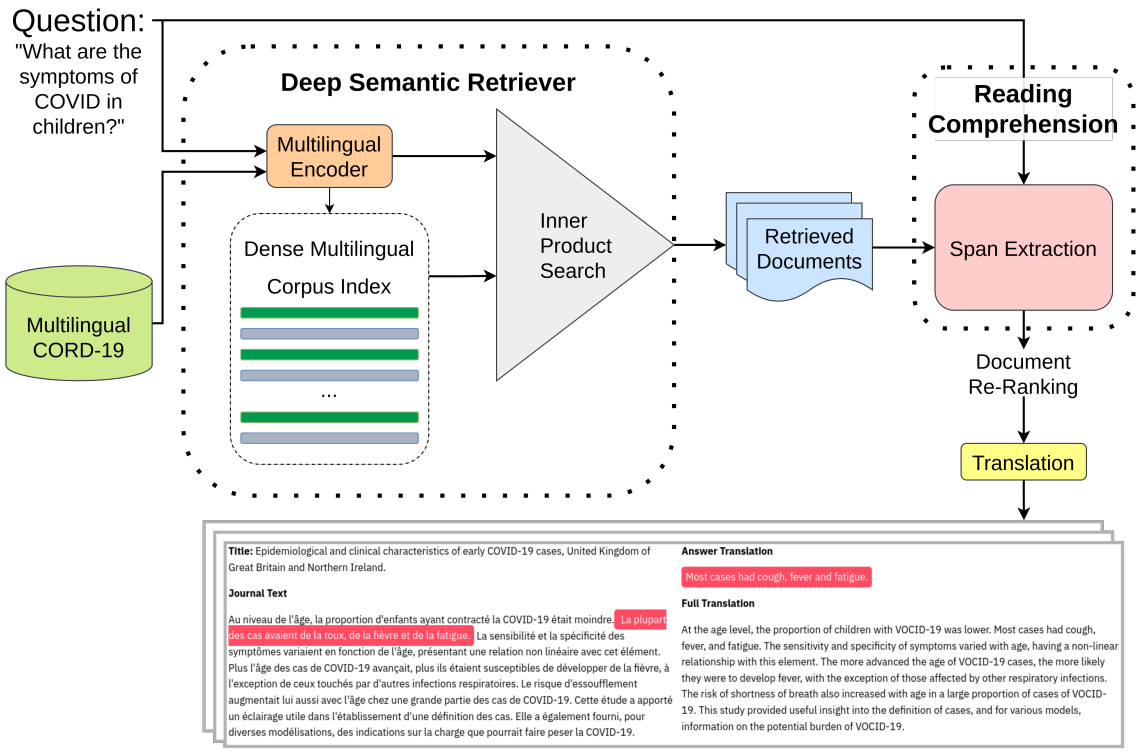

Figure 7.1: An overview of our cross-lingual COVID-19 open-retrieval question-answering system.

To address the limitations of prior systems, we implement a cross-lingual open-retrieval question answering (XOR-QA) system that retrieves answers from a large collection of multilingual documents, where answers may be in a language different from the question [208].

In this chapter we take COVID-19 as an exemplar of an emergent domain and present our system, which addresses two main areas of importance:

- *Cross-linguality*: The locality of an emergent domain is unknown ahead of time, making cross-lingual QA essential. Additionally, because data can rapidly change in emerging domains, new information may develop in multiple languages, motivating the need for systems that function across many languages.

- *Scarcity of training data*: Data scarcity is an expected concern for emergent domains,

but multilingual and cross-lingual data are even more limited. We demonstrate that by employing automatic translation, alignment, and filtering methods, this challenge can be overcome in low-resource open-retrieval QA.

This chapter provides in-depth technical descriptions of the individual components of our cross-lingual open-retrieval question answering (XOR-QA) system: cross-lingual retrieval and cross-lingual reading comprehension modules. Then, we describe how to combine the components along with document re-ranking into the complete system, shown in Figure 7.1, and present several examples taken from our system.

7.2 Cross-Lingual Dense Retrieval

Training a dense retriever is challenging in low-resource settings, such as emergent domains, due to the data-hungry nature of large language models. This challenge is compounded in the cross-lingual setting, where we aim to train a model to encode concepts from multiple languages into a similar location in the embedding space. In this section, we discuss how we overcome these challenges.

7.2.1 Data

Cross-lingual retrieval requires two datasets; a large-scale multilingual corpus of scientific articles from which to retrieve documents and a cross-lingual dataset for training the retriever. However, a very limited number of COVID-19 datasets have been released, few of which are multilingual and none of which are cross-lingual.

CORD-19 [209] is a large-scale corpus of scientific papers on COVID-19, however a known limitation is that it contains only English articles. We draw inspiration from CORD-19 to address the lack of a large scale corpus of multilingual

COVID-19 scientific articles. For our system, we use a manually collected corpus of English abstracts from PubMed, some of which have parallel abstracts in additional languages. The corpus is collected using the same query as described by Lu Wang et al. [209] . We call this corpus multilingual CORD-19 (mCORD-19), and the language distribution can be found in Table 7.1.

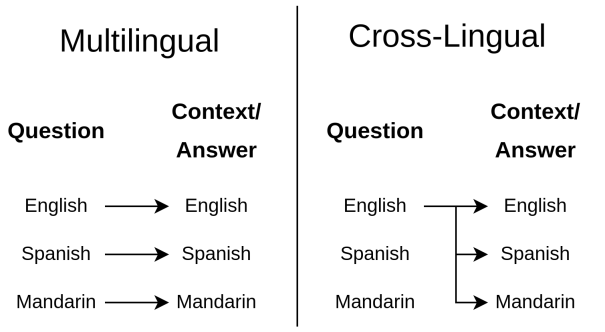

Figure 7.2: **Multilingual vs. cross-lingual question answering**: In the multilingual setting, QA pairs exist for multiple languages in a one-to-one mapping. On the other hand, in cross-lingual QA questions may have answers in any language, creating a one-to-many mapping.

To train our retriever we utilize the COUGH [210] dataset, which is a multilingual FAQ retrieval dataset and consists of COVID-19 QA pairs. Although COUGH is multilingual, containing samples in 9 different languages, COUGH does not contain any cross-lingual QA pairs. The language distribution is shown in Table 7.1.

7.2.2 Cross-lingual Data Generation

To address the lack of cross-lingual data in COUGH we draw inspiration from works in data augmentation [211, 13] and introduce a modification of the dataset which we call English-to-all (En2All), where we convert the dataset from the multilingual to cross-lingual setting, as demonstrated in Figure 7.2. Because we are interested in a system which will find non-English answers to English questions, we create En2All through two translation processes. First, we translate the answer portion of every QA pair from COUGH into eight languages: Arabic, French, German, Italian, Mandarin, Russian, Spanish, and Vietnamese. Secondly, we translate the question portion of all QA pairs from any of the

COUGH	9151 (en)	1077 (es)	778 (zh)	697 (fr)	573 (ja)	531 (ar)
mCORD-19	172977 (en)	1109 (es)	951 (zh)	711 (de)	614 (fr)	328 (pt)

Table 7.1: Top 6 languages by count for COUGH and the multilingual CORD-19 datasets. Language codes are the following: en-English, es-Spanish, zh-Chinese, fr-French, de-German, ja-Japanese, ar-Arabic, pt-Portuguese.

Answer Language	**Spanish**	**Mandarin**	**French**	**Arabic**	**German**	**Russian**	**Vietnamese**	**Italian**
En2All	8695	8441	8372	8231	8226	8156	8072	8003
Filtered En2All	6620	5869	5635	5808	5867	4137	531	6568

Table 7.2: QA pairs in our En2All and Filtered En2All variants of the COUGH dataset, where each question is in English, and the context and answer are in the language specified above.

above languages into English[1].

As machine translation models do not perform perfectly, there may be instances within En2All that contain poor translations. To resolve this problem, we utilize LaBSE [213], an existing BERT-based sentence embedding model that encodes 109 languages into a shared embedding space. The model is utilized to compare the alignment of translations across different languages. We take the following steps to filter out any poor translations in the data:

1. We step through the current En2All and calculate similarity scores between translated answers and their original English answers. To do this, we have eight different comparisons for each translated English QA pair.

2. Once the similarity scores have been calculated, we remove translations that do not meet a threshold and are classified as poor translations.

After going through these steps, roughly one-third of the data samples from En2All are removed for poor translations.

[1]All translations are generated by the MarianNMT system [212] through the Huggingface Transformers [49] library.

7.2.3 Methodology: Deep Semantic Retriever

Our retrieval model is based on the dense passage retriever from Karpukhin et al. [214]. In contrast to their work, we train a unified encoder that encodes both query and corpus into a shared space. For the encoder, we train the multilingual BERT (mBERT) [47] and XLM-RoBERTa (XLM-R) [215] models. Both models have been pre-trained using a tokenizer which shares a vocabulary for over 100 languages, allowing the models to encode all languages into a shared space. We train these models on the FAQ retrieval task by maximizing the inner product of correct QA pairs and minimizing the inner product of within-batch incorrect pairs.

7.2.4 Cross-Lingual Retrieval Evaluation

To evaluate our models in the large-scale open-retrieval setting we utilize the questions from COUGH and En2All as our queries and the mCORD-19 dataset for our retrieval corpus. Because we have no ground truth labels for correct documents, and indeed there may be some unanswerable questions given this corpus, we measure model quality through a fuzzy matching metric, Fuzzy Match at top k documents (FM@k). FM@k utilizes the multilingual Sentence-BERT model from [216][2]. Each of the top k retrieved documents is split into it's component sentences and embedded using the sentence-BERT model. Next, each sentence is compared with the ground truth answer by calculating the cosine similarity with the reference answer embedding from COUGH. If any of the cosine similarities for that documents sentences are above a threshold, the document is evaluated as a positive retrieval.

The results for our models and a BM25 baseline[3] are found in Table 7.3. Since a

[2]We use the 'paraphrase-multilingual-mpnet-base-v2' variant

[3]Implementation details at `https://github.com/alon-albalak/XOR-COVID/tree/master/bm25`

Model	**COUGH** (FM@5/100)	**COUGH +En2All** (FM@5/100)
BM25[4]	18.6/41.4	
$\text{mBERT}_{\text{base}}$	22.8/49.5	26.4/50.7
+ En2All	28.0/54.9	27.7/51.7
$\text{XLM-R}_{\text{base}}$	25.0/51.3	28.1/51.6
+ En2All	30.1/55.4	28.4/52.2
+ Filtered-En2All	**32.9/56.7**	**30.9/53.4**
$\text{XLM-R}_{\text{large}}$	30.5/56.6	29.8/53.2
+ En2All	32.1/56.4	29.6/52.9

Table 7.3: **Retrieval evaluation results**. All models are trained on COUGH and additional training data is denoted by "+". The middle column takes queries from COUGH, the right column from COUGH and En2All. For both columns, the retrieval corpus is mCORD. FM@5 and FM@100 are the fuzzy matching techniques proposed to determine open-retrieval accuracy described in section 7.2.4. Because BM25 is not cross-lingual, we translate it's queries into all languages in order to fairly compare against our cross-lingual models.

multilingual BM25 cannot perform cross-lingual retrieval, in order to fairly compare against cross-lingual models, we translate all queries into every other language in the mCORD corpus and then perform BM25 retrieval.

BM25 drastically underperforms compared to encoder models and demonstrates the need for a dense retrieval model. Although encoder models outperform BM25 when trained on multilingual data (COUGH), they are further improved by training on cross-lingual data (En2All). Additionally, after filtering low quality translations from En2All, we see further improvement in performance.

7.3 Cross-Lingual Reading Comprehension

7.3.1 Data

To train our cross-lingual reading comprehension model, we would ideally use a cross-lingual covid-specific question answering dataset. However, similarly to cross-lingual retrieval no such dataset exists so we augment existing datasets.

Artetxe et al. [217] introduced XQuAD, a multilingual QA dataset composed of 240 paragraphs and 1190 QA pairs from SQuAD v1.1 which have been professionally translated into 10 languages. We utilize XQuAD as a pretraining dataset before performing any training on covid-specific datasets[4]. Möller et al. [218] introduce Covid-QA, a covid-specific QA dataset consisting of 2019 question-answer pairs, however, it contains english-only data. We modify Covid-QA with translations from MarianMT [212] to generate two dataset variants based on the multilingual and cross-lingual settings shown in Figure 7.2: Multilingual Covid-QA (MCQA) and English-to-all (En2All). MCQA is a multilingual version of Covid-QA, created by translating all QA pairs into 9 languages to match those from XQuAD: Arabic, German, Greek, Spanish, Hindi, Mandarin, Romanian, Russian, and Vietnamese. En2All is our cross-lingual variation of Covid-QA, in a similar spirit to the cross-lingual variant of COUGH. Because Covid-QA is english-only, to generate En2All we translate all contexts/answers into the same 9 languages as MCQA.

7.3.2 Methodology: Span Extraction

Similar to our dense semantic retriever, we train mBERT and XLM-RoBERTa models for our reading comprehension task. We formulate reading comprehension as a span extraction task, where each model learns to find start and end tokens which represent the

[4]We open-source our models pretrained on XQuAD at https://huggingface.co/alon-albalak

Model	MCQA (EM/F1)	MCQA+En2All (EM/F1)
$\text{mBERT}_{\text{base}}$	20.0/57.5	19.6/55.4
+ XQuAD	21.2/57.7	20.5/55.6
+ En2All	19.3/56.1	19.2/55.8
$\text{XLM-R}_{\text{base}}$	25.1/60.0	24.4/58.9
+ XQuAD	26.7/61.6	26.1/61.3
+ En2All	24.0/58.8	23.9/58.3
$\text{XLM-R}_{\text{large}}$	26.5/**62.7**	26.4/**62.2**
+ XQuAD	**29.1**/62.1	**29.0**/61.7
+ En2All	26.3/61.1	26.6/60.8

Table 7.4: **Reading comprehension evaluation results**. All models are trained on MCQA, and additional training data is denoted by "+". The left column shows evaluation on a multilingual dataset where questions/contexts are always in the same language. The right column additionally evaluates on a cross-lingual dataset where questions are in english and context paragraphs may be in any language.

answer span in a document.

7.3.3 Cross-Lingual Reading Comprehension Evaluation

To evaluate our models in the reading comprehension task, we utilize the QA datasets described in Section 7.3.1. We evaluate our models based on exact match (EM) and F1 metrics by comparing the predicted answer spans with ground-truth answers.

The results for our models are found in Table 7.4. We train each of our models on MCQA and supplement it with data from XQuAD or En2All. Interestingly, we find that although En2All improved models in the retrieval setting, it only hurt model performance in QA. We also see that pretraining on XQuAD improves performance in all metrics for both base models, but leads to a slight decrease in F1 score for $\text{XLM-R}_{\text{large}}$. In our system, we utilize $\text{XLM-R}_{\text{large}}$ which was pretrained on XQuAD because it has only slightly worse F1 score, but significantly higher exact match compared to the next best model.

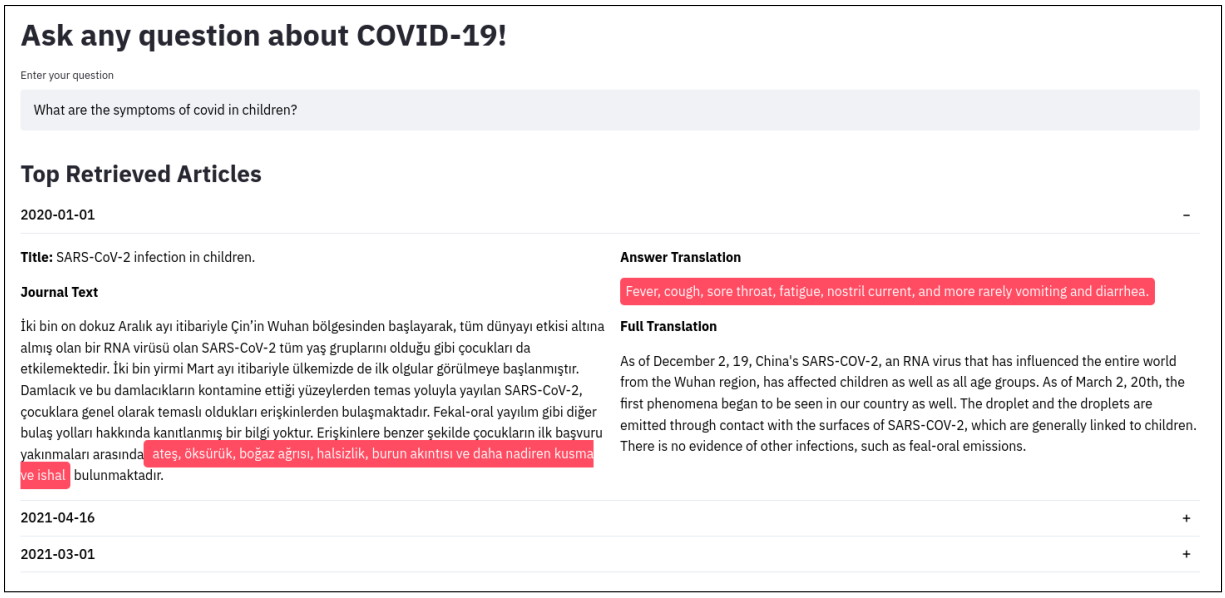

Figure 7.3: **The main interface of our system**. At the top is the search bar, where the current query is "What are the symptoms of covid in children?" Below the search bar are the three retrieved articles, ranked by relevance. In this example, the first retrieved document has been expanded to show the title and original text in Turkish, on the left. And on the right is the translation of the answer and the full document into English.

7.4 Cross-Lingual Open-Retrieval Question Answering

Our system is composed of the retrieval and reading comprehension modules described in sections 7.2 and 7.3. The full end-to-end system is shown in Figure 7.1. After the retriever has been trained, the mCORD-19 corpus is encoded and stored in the dense multilingual corpus index. When a question is posed to the system, the query is encoded, and a maximum inner product search is performed over the index to find documents most similar to the query. Answers are then extracted from the retrieved documents and the documents are re-ranked based on answer confidence from the span extraction model. Finally, the answer spans and full documents are translated into English and presented to the user with highlighted answers.

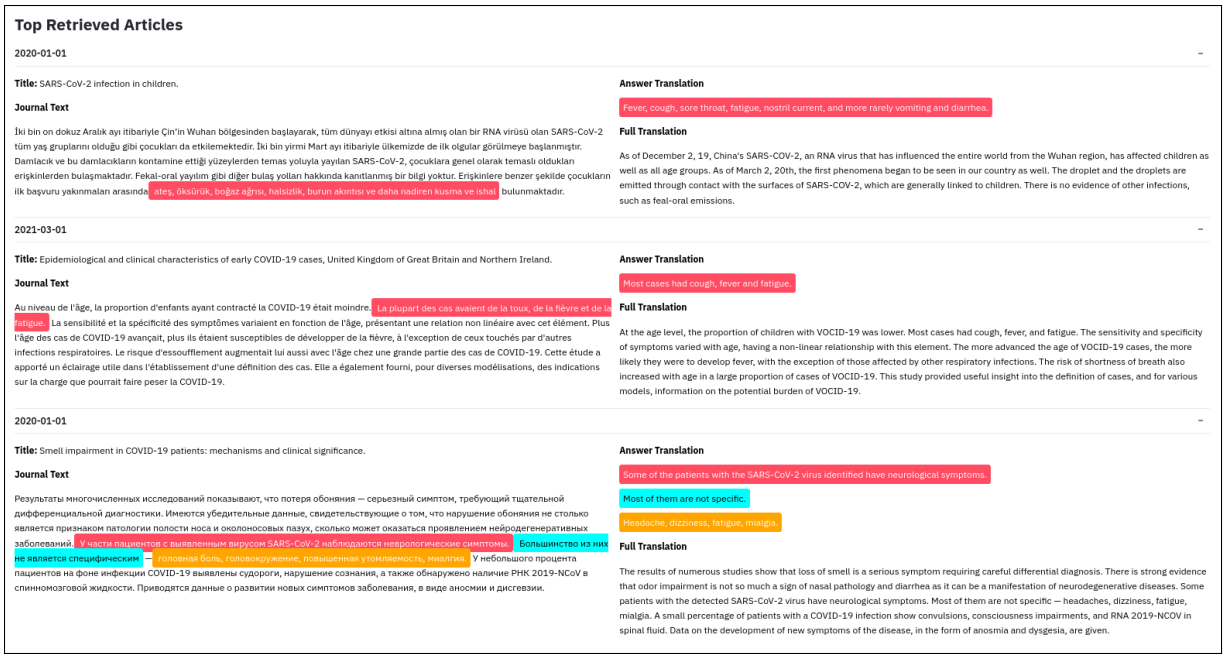

Figure 7.4: The top 3 non-English results for the query "What are the symptoms of covid in children?"

7.5 System Description

The system retrieves documents from our mCORD-19 corpus, which has been encoded by the deep semantic retriever from section 7.2.3. We provide examples from our system in Figures 7.4, 7.5, and 7.6.

7.5.1 Sidebar Interface

Our system has an options sidebar, shown in Figure 7.7, which gives the user several choices before entering a query. The user can determine how many documents they would like to see results from, they can select which languages the retrieved documents should be in, and they can specify a date range for the publications to search over. If there are no relevant documents in the desired date range, then the system will retrieve from any date range and displays a message to inform the user.

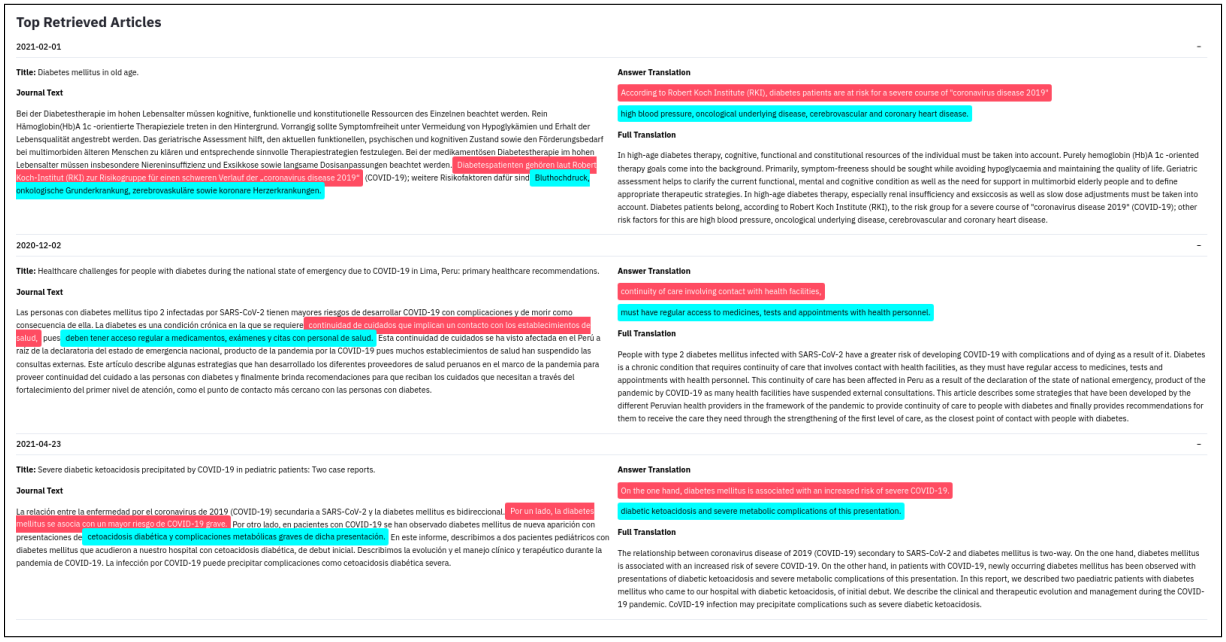

Figure 7.5: The top 3 non-english results for the query "What are the concerns of having covid and diabetes?"

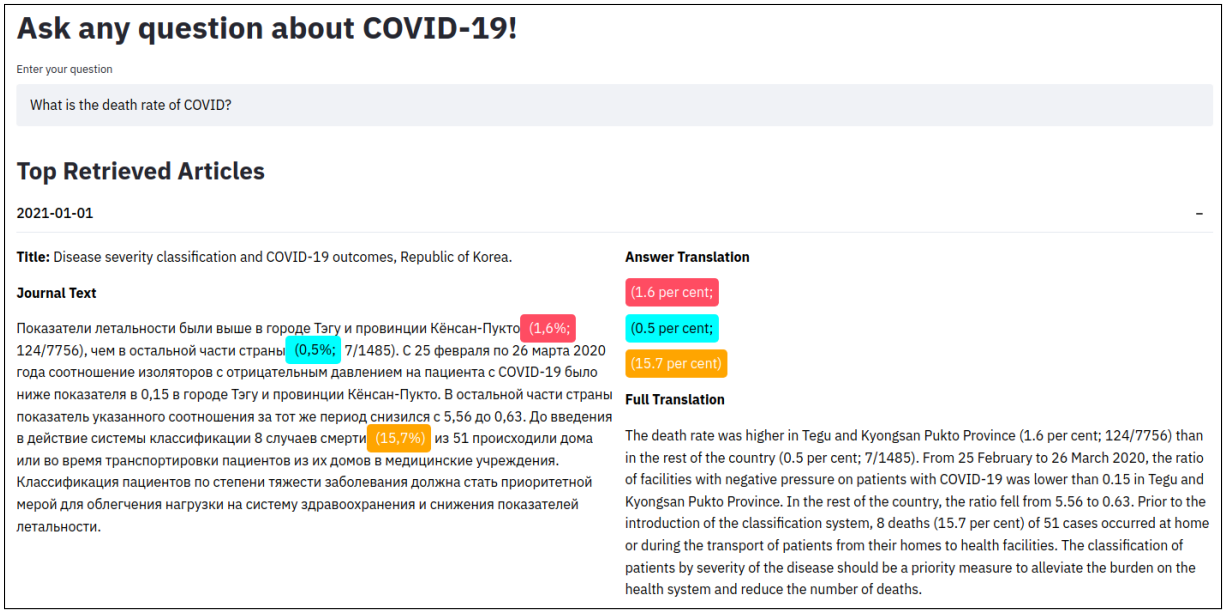

Figure 7.6: A retrieved document for the query "What is the death rate of COVID", which shows multiple correct answers corresponding to different provinces of South Korea.

7.5.2 Main Interface

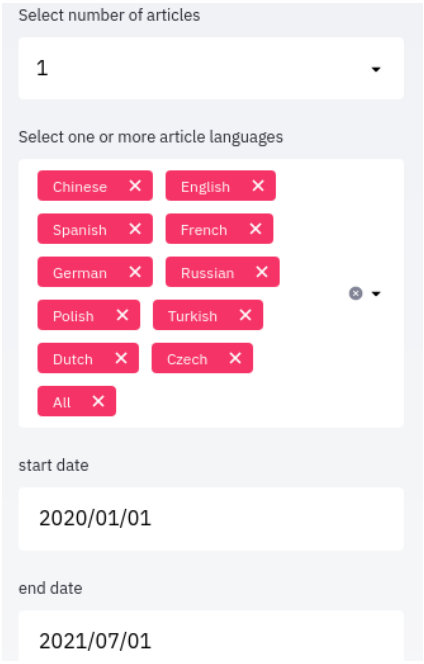

Figure 7.7: The options sidebar for our demonstration system. The options include: number of articles to return, article languages to retrieve from, and publication date range. For visualization purposes we show all language options.

To query the system, a user simply selects the desired options from the sidebar and enters their question into the search bar, as seen in Figure 7.3. After the user enters their question, the system will encode the question using the trained deep semantic retriever and find the most relevant documents within the given language and date range constraints. Then, the reading comprehension model will extract the answer (or answers) most relevant to the query from each retrieved document. Additionally, for any non-English documents, the system translates both the retrieved article and extracted answers into English[5]. Finally, the retrieved documents will be re-ranked based on the confidence scores for the extracted answers.

The desired number of documents will be displayed to the user as a list of publication dates. Each item can be expanded to show the article title, original document with highlighted answers, translated answers, and the full article translation. If an article contains a single answer, it will be highlighted in red. If there are multiple answers, each answer will be highlighted with a different color to allow for easy alignment between original answers and their translations, demonstrated in Figure 7.6.

[5]All translations are generated by MarianNMT [212] from the Huggingface Transformers library [49].

Chapter 8

Conclusions and Future Work

8.1 Summary

In this dissertation we outlined a data-centric paradigm for improving language models which is orthogonal to scaling. In Part I we demonstrated methods for improving our understanding of language model capabilities based on their training data, as well as proposing one method for improving the interpretability of models through the use of data. Furthermore, in Part II we provided methods for improving the data used to train models that have proven to improve data efficiency and performance on both pretraining and downstream tasks.

8.1.1 Understanding models through data

Our research in Chapter 2 first demonstrates how to improve the explainability, and therefore our understanding, of relation extraction methods. We do so by creating a system that extracts explanations for the predicted relation using only partially labeled explanations. To overcome the partial supervision, we use a policy-guided semi-supervised learning algorithm that optimizes for explanation quality and relation extraction perfor-

mance simultaneously. We framed relation extraction as a re-ranking task and included entity-specific explanations as an interpretable intermediate step in the inference process. Our results showed that human annotators were 4.2 times more likely to prefer our systems explanations over an existing baseline. In addition to improving explainability, we also found that our system improves relation extraction performance over strong black-box baselines. One limitation of this method is that we have only validated that the explanations learned are meaningful for a single dataset, and it is not clear if the learned explanation model will transfer to a new dataset. Further studies can explore the idea of a multitask explainer model which could be trained to generate or extract explanations for a variety of tasks including question answering, topic classification, and sentiment analysis. By using an intermediate explanation model, we could further improve both the interpretability and explainability of systems, but also improve their reasoning ability. While we demonstrate the efficacy on relation extraction, the idea of introducing intermediate steps into the inference process can be applied to many more tasks to further improve our understanding of model decisions under many different scenarios.

Next, in Chapter 3 we perform a thorough analysis on whether the benefits of multitask learning (MTL), instruction tuning and prompting seen in large language models translate to smaller models. We explored and isolated the effects of (i) model size, (ii) general purpose MTL, (iii) in-domain MTL, and (iv) instruction tuning. Our results showed that general purpose MTL improved the performance of small models by 31% on average, and further in-domain MTL improved performance by an additional 37.6%, demonstrating the power of multitask learning for zero-shot settings. Contradictory to prior works on large models, our results showed that instruction tuning provided very minimal performance gains, only 2% on average. While our study isolates the contributions from these particular variables, there are still other variables that we do not study. For example, the BART-Base and Large models are both trained on the same dataset, and while this is crucial to

determine the effect of model size isolated from other factors, it means that we have only run experiments with 1 pretraining corpus. Ideally, we would run experiments on models of the same size but pretrained with different data to account for the effect of the pretraining corpus. Additionally, while we focus on small models, the smallest model we investigate is 139 million parameters, but since the conclusion of our study, the OPT and Pythia family of models have been released with many more small model sizes.

Then, in Chapter 4 we study task transfer in conversational AI by introducing FETA, a benchmark for FEw-sample TAsk transfer in open-domain dialogue. FETA contains two underlying sets of conversations upon which there are 7 and 10 tasks annotated, enable the unique study of intra-dataset task transfer; task transfer without domain adaptation. We analyze the intra-dataset task transfer of three popular language models and three transfer learning algorithms. Additionally, we consider both the single-source and multi-source settings to better understand how transfer learning scales with additional source tasks. Through extensive experimentation, we find new and non-intuitive insights on the mechanisms of transfer learning. In particular, our results show that most performance trends are model-specific, and we strongly encourage researchers to consider multiple model architectures before drawing broad conclusions on transfer learning. Additionally, we find that tasks which are deemed more challenging by humans (e.g. span extraction) benefit the most from task transfer. While our experiments do control for domain adaptation, there were aspects we did not control for such as the pretraining corpus of each model. Also, to ensure fair comparisons, we only tested base-sized models, but we would expect better pretraining corpora and larger models to lead to increased raw performance on the individual tasks in FETA. More importantly though, it is unclear whether either of these changes would lead to improved task-transfer performance (average and top-1 δs), and this is an interesting area for further research. In the future, FETA can be a valuable resource for further research into efficiency and generalizability of pretraining datasets and

model architectures, as well as for other learning settings such as continual and multitask learning. Additionally, FETA could be used to test whether a policy-guided algorithm such as D-REX can be used for the transfer learning setting.

8.1.2 Improving models through data

In Chapter 5, we switch topics to improving the data used to train models. We focus on the problem of few-shot learning with auxiliary data (FLAD), and design algorithms that (1) make no assumptions on the available auxiliary data a-priori, (2) scale well with the number of auxiliary datasets, and (3) add minimal memory and computational overhead. To achieve these goals, we formulated FLAD as a multi-armed bandit problem, which leads to computational complexity that is independent of the number of auxiliary datasets, allowing our method to scale to 100x more auxiliary datasets than prior methods. These significant improvements lead to the first 3 billion parameter models that outperform the 175 billion parameter GPT-3 on few-shot learning. This chapter builds on the lessons learned from Chapter 4, where we showed that naively increasing the number of source-tasks in transfer learning is not always beneficial. To improve upon that challenge, our algorithm uses rewards designed to find auxiliary datasets whose solution space is similar to the solution space of the target task.

Next, in Chapter 6 we show just how crucial data mixtures are for language model pretraining. Through the insight that the goal of language model pretraining is performed so that models can absorb large quantities of information, we design a reward function that accurately reflects how much information is gained by the model when seeing data from each of the training domains. We then use this reward with a variation of a multi-armed bandit algorithm that is extremely efficient, adding negligible wall-clock time during pretraining. We find that our method trains a model reaching the final

perplexity of the next best data mixing method with 19% fewer training iterations, and actually improves performance on the 5-shot MMLU benchmark. While this method demonstrated impressive performance gains across all training domains on average, it loses performance on three specific domains. In particular, we hypothesize that our method leads to slightly worse performance on GitHub due to the lower intrinsic entropy of code data (due to its highly structured nature). Some works have recently found that training on higher quantities of code can improve the reasoning capabilities of models [219], which our method does not take into consideration. A plausible next step to improving this method is to combine multiple signals together, including our very efficient information gain-based reward and possibly some slower signals that are gathered during validation. In combination, these signals can lead to further improvements.

In the previous 2 chapters, we approach data selection from a data mixing approach, where we organize many data points together into groups (tasks or domains) and assign the same value to all data points within the group. However, this top-down approach, where we select data for the capabilities that our model will have is only one option. It is also beneficial to work from a bottom-up approach to data selection, choosing individual data points for the value that they bring to the dataset. There are a number of methods to select individual data points [1], aiming to achieve different goals, and our works on data mixing can be a stepping stone to developing new data selection methods that find those individual data points which are most beneficial for a specific target task, as well as for those data points which are most informative for pretraining.

Finally, in Chapter 7 we demonstrate a system for cross-lingual open-retrieval question answering, which is particularly important in low-resource settings such as new and emerging domains, where the language of information is not known ahead of time. In particular, multilingual and cross-lingual resources are scare in emergent domains, leading to few or no such open-retrieval question answering systems. For our system we use

Covid-19 as the example of an emergent domain and address the scarcity of cross-lingual training data issue by utilizing automatic translation, alignment, and filtering to produce an augmented dataset. We show that our system significantly outperforms a BM25 baseline in the cross-lingual setting.

8.2 Future Work

Overall, my research goal is to continue doing open, responsible, and collaborative research. Open, to allow and encourage others to follow in our footsteps. Responsible, to ensure that our work is beneficial and to minimize harms. And collaborative, so that our work may be inclusive and consider many perspectives. With this in mind, I am most interested in pursuing two major directions of future work.

8.2.1 Data-centric research directions

First, I believe that an important direction of research is on making data research more accessible. This can be done by developing methods that directly measure data by expanding on methods of data attribution and valuation [220, 221], and data measurements [222]. Another direction would be to validate whether data research can be done on a smaller scale (model sizes and dataset sizes) and still transfer to larger models and datasets. In this dissertation, we explored methods for improving data mixing for both few-shot learning and for pretraining, but for pretraining in particular, there is no reason to believe that all data within a single domain has the same value. For this reason, it would be very valuable to extend these methods into data selection for individual data points. Additionally, memorization is a known issue in very large models [223], but how exactly to allow models to memorize "good" information (e.g. facts), while reducing "bad" memoriation (e.g. personally identifiable information) is still an open question.

8.2.2 Moving beyond siloed data research

Next, it is important to keep in mind that data is of no specific benefit in isolation, but becomes immensely important when combined with large neural models and advanced optimization procedures. With this in mind, I believe it will be a very important direction of research to consider all three components in the effort of continuing to improve models, efficiently. Furthermore, I believe that by expanding beyond just a single model, and into systems of models, where each model has a separate optimization objective and datasets for different goals, systems will be able to solve more abstract problems. As I've shown in Chapter 2 (as well as in other works [9, 6], systems such as this can become more interpretable and simultaneously more performant, and in the future I believe we should continue down this direction as models have become much more powerful in recent years. Finally, I believe that future research can more closely integrate humans and models together. While algorithms are wonderful for optimizing models for objective functions (immensely better than humans are), they optimize without care for societal impacts (e.g. bias) and side-effects (where humans are much better). The combination of humans + machines, with models as tools augmenting human capabilities, can allow people to spend their effort on defining success and letting machines optimize for that definition.

Bibliography

[1] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

[2] Shayne Longpre, Stella Biderman, Alon Albalak, Gabriel Ilharco, Sayash Kapoor, Kevin Klyman, Kyle Lo, Maribeth Rauh, Nay San, Hailey Schoelkopf, Aviya Skowron, Bertie Vidgen, Laura Weidinger, Arvind Narayanan, Victor Sanh, David Adelani, Percy Liang, Rishi Bommasani, Peter Henderson, Sasha Luccioni, Yacine Jernite, and Luca Soldaini. *The Foundation Model Development Cheatsheet*, 2024. URL `https://github.com/allenai/fm-cheatsheet/blob/main/app/resources/paper.pdf`.

[3] Alon Albalak, Colin Raffel, and William Yang Wang. Improving few-shot generalization by exploring and exploiting auxiliary data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=JDnLXc4NOn`.

[4] Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. *arXiv preprint arXiv:2312.02406*, 2023.

[5] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936. URL `https://aclanthology.org/2023.findings-emnlp.936`.

[6] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248.

[7] Yi-Lin Tuan, Alon Albalak, Wenda Xu, Michael Saxon, Connor Pryor, Lise Getoor, and William Yang Wang. CausalDialogue: Modeling utterance-level causality in conversations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12506–12522, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.792. URL https://aclanthology.org/2023.findings-acl.792.

[8] Alon Albalak, Sharon Levy, and William Yang Wang. Addressing issues of cross-linguality in open-retrieval question answering systems for emergent domains. In Danilo Croce and Luca Soldaini, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 1–10, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-demo.1. URL https://aclanthology.org/2023.eacl-demo.1.

[9] Connor Pryor, Charles Dickens, Eriq Augustine, Alon Albalak, William Yang Wang, and Lise Getoor. Neupsl: Neural probabilistic soft logic. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4145–4153. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/461. URL https://doi.org/10.24963/ijcai.2023/461. Main Track.

[10] 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.

[11] Alon Albalak, Akshat Shrivastava, Chinnadhurai Sankar, Adithya Sagar, and Mike Ross. Data-efficiency with a single gpu: An exploration of transfer methods for small language models. *arXiv preprint arXiv:2210.03871*, 2022.

[12] Charles Andrew Dickens, Connor Pryor, Eriq Augustine, Alon Albalak, and Lise Getoor. Efficient learning losses for deep hinge-loss markov random fields. In *The 5th*

Workshop on Tractable Probabilistic Modeling, 2022. URL https://openreview.net/forum?id=8ZIJa8Z__5L.

[13] Zekun Li, Hong Wang, Alon Albalak, Yingrui Yang, Jing Qian, Shiyang Li, and Xifeng Yan. Making something out of nothing: Building robust task-oriented dialogue systems from scratch. In *Alexa Prize TaskBot Challenge 1 Proceedings*, 2022. URL https://www.amazon.science/alexa-prize/proceedings/making-something-out-of-nothing-building-robust-task-oriented-dialogue-systems-

[14] Alon Albalak, Varun Embar, Yi-Lin Tuan, Lise Getoor, and William Yang Wang. D-REX: Dialogue relation extraction with explanations. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 34–46, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.nlp4convai-1.4. URL https://aclanthology.org/2022.nlp4convai-1.4.

[15] Rohit Jain, Devin H Redmond, Richard B Sutton, Alon Albalak, and Sharon Hüffner. Systems and methods for determining and using semantic relatedness to classify segments of text, February 27 2024. US Patent 11,914,963.

[16] Michael Saxon, Sharon Levy, Xinyi Wang, Alon Albalak, and William Yang Wang. Modeling disclosive transparency in NLP application descriptions. 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 2023–2037, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.153. URL https://aclanthology.org/2021.emnlp-main.153.

[17] Terry Winograd. *Procedures as a representation for data in a computer program for understanding natural language*. MIT. Cent. Space Res., Cambridge, MA, 1971. URL https://cds.cern.ch/record/233416.

[18] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, jan 1966. ISSN 0001-0782. doi: 10.1145/365153.365168. URL https://doi.org/10.1145/365153.365168.

[19] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, Robert L Mercer, and Paul Roossin. A statistical approach to language translation. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.

[20] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, John Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85, 1990.

[21] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[25] Suhas Kumar. Fundamental limits to moore's law. *arXiv preprint arXiv:1511.05956*, 2015.

[26] Tim Cross. After moore's law. *The Economist Technology Quarterly*, 2016. URL `http://www.economist.com/technology-quarterly/2016-03-12/after-moores-law`.

[27] Wallace Witkowski. Moore's law's dead. *MarketWatch*, 2022. URL `https://www.marketwatch.com/story/moores-laws-dead-nvidia-ceo-jensen-says-in-justifying-gaming-card-price-hike-11`

[28] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[29] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[30] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[31] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le

Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=9Vrb9D0WI4`.

[32] 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. *Transactions on Machine Learning Research*, 2022. URL `https://openreview.net/forum?id=yzkSU5zdwD`. Survey Certification.

[33] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[34] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[35] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024.

[36] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.

[37] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017.

[38] Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E17-1110`.

[39] Xiaoyu Han and Lei Wang. A novel document-level relation extraction method based on bert and entity information. *IEEE Access*, 8:96912–96919, 2020.

[40] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, 2019.

[41] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.444. URL https://www.aclweb.org/anthology/2020.acl-main.444.

[42] H. Chen, Pengfei Hong, Wei Han, Navonil Majumder, and Soujanya Poria. Dialogue relation extraction with document-level heterogeneous graph attention networks. *ArXiv*, abs/2009.05092, 2020.

[43] Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. Gdpnet: Refining latent multi-view graph for relation extraction. In *AAAI*, 2021.

[44] Liang Qiu, Yuan Liang, Yizhou Zhao, Pan Lu, Baolin Peng, Zhou Yu, Ying Nian Wu, and Song-Chun Zhu. Socaog: Incremental graph parsing for social relation inference in dialogues. In *ACL/IJCNLP*, 2021.

[45] Bongseok Lee and Yong Suk Choi. Graph based network with contextualized representations of turns in dialogue. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 443–455, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.36.

[46] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/4c7a167bb329bd92580a99ce422d6fa6-Paper.pdf.

[47] 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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[48] 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. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

[49] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison,

Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[50] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[51] Dian Yu and Heng Ji. Unsupervised person slot filling based on graph mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44–53, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1005. URL https://www.aclweb.org/anthology/P16-1005.

[52] Hamed Shahbazi, Xiaoli Fern, Reza Ghaeini, and Prasad Tadepalli. Relation extraction with explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6488–6494, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.579. URL https://www.aclweb.org/anthology/2020.acl-main.579.

[53] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016. URL http://arxiv.org/abs/1612.08220.

[54] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[55] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam, 2018. URL https://openreview.net/forum?id=rk6qdGgCZ.

[56] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. URL http://arxiv.org/abs/1711.05101.

[57] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

[58] Ethan Zhou and Jinho D. Choi. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International*

Conference on Computational Linguistics, pages 24–34, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1003.

[59] Xintong Yu, Hongming Zhang, Yangqiu Song, Yan Song, and Changshui Zhang. What you see is what you get: Visual pronoun coreference resolution in dialogues. 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 5123–5132, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1516. URL https://www.aclweb.org/anthology/D19-1516.

[60] Sayyed M Zahiri and Jinho D Choi. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In *Workshops at the thirty-second aaai conference on artificial intelligence*, 2018.

[61] Eriq Augustine, Pegah Jandaghi, Alon Albalak, Connor Pryor, Charles Dickens, William Wang, and Lise Getoor. Emotion recognition in conversation using probabilistic soft logic. *arXiv preprint arXiv:2207.07238*, 2022.

[62] Róbert Ormándi, Mohammad Saleh, Erin Winter, and Vinay Rao. Webred: Effective pretraining and finetuning for relation extraction on the web. *CoRR*, abs/2102.09681, 2021. URL https://arxiv.org/abs/2102.09681.

[63] Sawan Kumar and Partha Talukdar. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.771. URL https://www.aclweb.org/anthology/2020.acl-main.771.

[64] Hui Liu, Qingyu Yin, and William Yang Wang. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1560. URL https://www.aclweb.org/anthology/P19-1560.

[65] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1011. URL https://www.aclweb.org/anthology/D16-1011.

[66] Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1884–1895, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1175. URL https://www.aclweb.org/anthology/P18-1175.

[67] Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. Seed-based event trigger labeling: How far can event descriptions get us? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 372–376, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2061. URL https://www.aclweb.org/anthology/P15-2061.

[68] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, aug 2022. ISSN 0360-0300. doi: 10.1145/3560815. URL https://doi.org/10.1145/3560815. Just Accepted.

[69] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *ArXiv*, abs/2104.08773, 2021.

[70] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

[71] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[72] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

[73] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. URL https://arxiv.org/abs/2202.12837.

[74] Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey P. Bigham. Improving zero and few-shot generalization in dialogue through instruction tuning, 2022. URL https://arxiv.org/abs/2205.12673.

[75] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.468. URL https://aclanthology.org/2021.emnlp-main.468.

[76] 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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[77] 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 *ACL*, 2020.

[78] Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. Unsupervised cross-task generalization via retrieval augmentation. *ArXiv*, abs/2204.07937, 2022.

[79] Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*, 2022.

[80] Lorien Y. Pratt, Jack Mostow, and Candace A. Kamm. Direct transfer of learned information among neural networks. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*, AAAI'91, page 584–589. AAAI Press, 1991. ISBN 0262510596.

[81] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6 (61):1817–1853, 2005. URL http://jmlr.org/papers/v6/ando05a.html.

[82] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1161. URL https://aclanthology.org/P17-1161.

[83] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

[84] Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. Learning to speak and act in a fantasy text adventure game. 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 673–683, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1062. URL https://aclanthology.org/D19-1062.

[85] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38:1 – 32, 2020.

[86] Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.273. URL https://aclanthology.org/2020.emnlp-main.273.

[87] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-5004. URL https://aclanthology.org/N19-5004.

[88] S. Mehri, M. Eric, and D. Hakkani-Tur. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *ArXiv*, abs/2009.13570, 2020.

[89] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e946209592563be0f01c844ab2170f0c-Paper.pdf.

[90] Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. Soloist: Building Task Bots at Scale with Transfer Learning and Machine Teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824, 08 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00399. URL https://doi.org/10.1162/tacl_a_00399.

[91] Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across nlp tasks. In *EMNLP*, 2020.

[92] Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. In *EMNLP*, 2021.

[93] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[94] 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 *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.

[95] Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. *AAAI*, 2021.

[96] Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Vzh1BFUCiIX.

[97] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010. URL http://www.springerlink.com/content/q6qk230685577n52/.

[98] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

[99] Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2026.

[100] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–480, 1992. URL https://aclanthology.org/J92-4003.

[101] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

[102] Baolin Peng, Chengkun Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. Raddle: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. *ArXiv*, abs/2012.14666, 2021.

[103] Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *ACL*, 2020.

[104] Pegah Jandaghi, Pei Zhou, Alon Albalak, and Jay Pujara. T-measure: A measure for model transferabilty. 2023.

[105] Arghya Pal and Vineeth N Balasubramanian. Zero-shot task transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2189–2198, 2019.

[106] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[107] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109:43–76, 2021.

[108] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[109] Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. Self-supervised meta-learning for few-shot natural language classification tasks. In *EMNLP*, 2020.

[110] Subhabrata (Subho) Mukherjee, Xiaodong Liu, Guoqing Zheng, Saghar Hosseini, Hao Cheng, Greg Yang, Chris Meek, Ahmed H. Awadallah, and Jianfeng Gao. Clues: Few-shot learning evaluation in natural language understanding. In *NeurIPS 2021*, December 2021. URL https://www.microsoft.com/en-us/research/publication/clues-few-shot-learning-evaluation-in-natural-language-understanding/.

[111] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-1099.

[112] Yu-Hsin Chen and Jinho D. Choi. Character identification on multiparty conversation: Identifying mentions of characters in TV shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles, September 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-3612. URL https://aclanthology.org/W16-3612.

[113] Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, Alexander Gelbukh, and Rada Mihalcea. Recognizing emotion cause in conversations. *Cognitive Computation*, 2021.

[114] Deepanway Ghosal, Pengfei Hong, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. CIDER: Commonsense inference for dialogue explanation and reasoning. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 301–313, Singapore and Online, July 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.sigdial-1.33.

[115] Ananya B. Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M. Khapra. Improving Dialog Evaluation with a Multi-reference Adversarial Dataset and Large Scale Pretraining. *Transactions of the Association for Computational Linguistics*, 8:810–827, 12 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00347. URL https://doi.org/10.1162/tacl_a_00347.

[116] Kaixin Ma, Tomasz Jurczyk, and Jinho D. Choi. Challenging reading comprehension on daily conversation: Passage completion on multiparty dialog. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2039–2048, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1185. URL https://aclanthology.org/N18-1185.

[117] Zhengzhe Yang and Jinho D. Choi. FriendsQA: Open-domain question answering on TV show transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 188–197, Stockholm, Sweden, September 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5923. URL https://aclanthology.org/W19-5923.

[118] Hang Jiang, Xianzhe Zhang, and Jinho D Choi. Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13821–13822, 2020.

[119] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://arxiv.org/abs/2004.08056v1.

[120] 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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, 2019.

[121] Rich Caruana. Learning many related tasks at the same time with backpropagation. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL https://proceedings.neurips.cc/paper/1994/file/0f840be9b8db4d3fbd5ba2ce59211f55-Paper.pdf.

[122] 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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

[123] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

[124] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=uYLFoz1vlAC`.

[125] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pages 28043–28078. PMLR, 2023.

[126] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.

[127] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=rJY0-Kcll`.

[128] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), jun 2020. ISSN 0360-0300. doi: 10.1145/3386252. URL `https://doi.org/10.1145/3386252`.

[129] Archit Parnami and Minwoo Lee. Learning from few examples: A summary of approaches to few-shot learning, 2022. URL `https://arxiv.org/abs/2203.04291`.

[130] Pengcheng Wu and Thomas G. Dietterich. Improving svm accuracy by training on auxiliary data sources. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 110, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015436. URL `https://doi.org/10.1145/1015330.1015436`.

[131] Reza Esfandiarpoor, Amy Pu, Mohsen Hajabdollahi, and Stephen H. Bach. Extended few-shot learning: Exploiting existing resources for novel tasks, 2020. URL `https://arxiv.org/abs/2012.07176`.

[132] Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2020. URL https://arxiv.org/abs/1812.02224.

[133] Sam Verboven, Muhammad Hafeez Chaudhary, Jeroen Berrevoets, Vincent Ginis, and Wouter Verbeke. Hydalearn. *Applied Intelligence*, Jul 2022. ISSN 1573-7497. doi: 10.1007/s10489-022-03695-x. URL https://doi.org/10.1007/s10489-022-03695-x.

[134] 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. *arXiv preprint arXiv:2205.06262*, 2022.

[135] Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across nlp tasks. *arXiv preprint arXiv:2005.00770*, 2020.

[136] Sebastian Thrun and Lorien Pratt, editors. *Learning to Learn*. Kluwer Academic Publishers, USA, 1998. ISBN 0792380479.

[137] Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. Self-supervised meta-learning for few-shot natural language classification tasks. *arXiv preprint arXiv:2009.08445*, 2020.

[138] Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylzhkBtDB.

[139] Lucio M. Dery, Paul Michel, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. AANG : Automating auxiliary learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vtVDI3w_BLL.

[140] Shuxiao Chen, Koby Crammer, Hangfeng He, Dan Roth, and Weijie J Su. Weighted training for cross-task learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ltM1RMZntpu.

[141] William G Macready and David H Wolpert. Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on evolutionary computation*, 2(1): 2–22, 1998.

[142] Alex Simpkins, Raymond De Callafon, and Emanuel Todorov. Optimal trade-off between exploration and exploitation. In *2008 American Control Conference*, pages 33–38. IEEE, 2008.

[143] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1): 48–77, 2002.

[144] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[145] Xingyu Lin, Harjatin Baweja, George Kantor, and David Held. Adaptive auxiliary task weighting for reinforcement learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/0e900ad84f63618452210ab8baae0218-Paper.pdf.

[146] Lucio M. Dery, Paul Michel, Ameet S. Talwalkar, and Graham Neubig. Should we be pre-training? an argument for end-task aware training as an alternative. *ArXiv*, abs/2109.07437, 2021.

[147] Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary learning by implicit differentiation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=n7wIfYPdVet.

[148] Hamish Ivison, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. Data-efficient finetuning using cross-task nearest neighbors, 2022. URL https://arxiv.org/abs/2212.00196.

[149] Sören Mindermann, Muhammed Razzak, Winnie Xu, Andreas Kirsch, Mrinank Sharma, Adrien Morisot, Aidan N. Gomez, Sebastian Farquhar, Janina Brauner, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learned. In *International Conference on Machine Learning*, 2021.

[150] Shoaib Ahmed Siddiqui, Nitarshan Rajkumar, Tegan Maharaj, David Krueger, and Sara Hooker. Metadata archaeology: Unearthing data subsets by leveraging training dynamics. *ArXiv*, abs/2209.10015, 2022.

[151] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=UmvSlP-PyV.

[152] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023.

[153] RICHARD BELLMAN. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957. ISSN 00959057, 19435274. URL http://www.jstor.org/stable/24900506.

[154] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. doi: 10.1017/9781108571401.

[155] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=F1vEjWK-lH_.

[156] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C. Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Transactions on Image Processing*, 23(2):684–695, 2014. doi: 10.1109/TIP.2013.2293423.

[157] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[158] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th annual foundations of computer science*, pages 322–331. IEEE, 1995.

[159] Yevgeny Seldin, Csaba Szepesvári, Peter Auer, and Yasin Abbasi-Yadkori. Evaluation and analysis of the performance of the exp3 algorithm in stochastic environments. In Marc Peter Deisenroth, Csaba Szepesvári, and Jan Peters, editors, *Proceedings of the Tenth European Workshop on Reinforcement Learning*, volume 24 of *Proceedings of Machine Learning Research*, pages 103–116, Edinburgh, Scotland, 30 Jun–01 Jul 2013. PMLR. URL https://proceedings.mlr.press/v24/seldin12a.html.

[160] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, pages 174–188, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-24412-4.

[161] Lai Wei and Vaibhav Srivastava. Nonstationary stochastic multiarmed bandits: Ucb policies and minimax regret, 2021.

[162] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459. URL http://www.jstor.org/stable/2282952.

[163] 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. *J. Mach. Learn. Res.*, 21(1), jun 2020. ISSN 1532-4435.

[164] Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, 2012.

[165] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[166] Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, 2018.

[167] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[168] Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *Sinn und Bedeutung*, 2019.

[169] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6.

[170] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

[171] Keisuke Sakaguchi, Ronan Bras, Chandra Bhagavatula, and Choi Yejin. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8732–8740, 04 2020. doi: 10.1609/aaai.v34i05.6399.

[172] Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*, 2018.

[173] Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=rBCvMG-JsPd.

[174] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=ShnM-rRh4T.

[175] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/shazeer18a.html.

[176] Rémi Lebret, David Grangier, and Michael Auli. Generating text from structured data with application to the biography domain. *CoRR*, abs/1603.07771, 2016. URL http://arxiv.org/abs/1603.07771.

[177] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10437–10446, 2020.

[178] Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E Gonzalez, Kurt Keutzer, and Trevor Darrell. Multitask vision-language prompt tuning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5656–5667, 2024.

[179] Jinlan Fu, See-Kiong Ng, and Pengfei Liu. Polyglot prompt: Multilingual multitask prompt training. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9919–9935, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.674. URL https://aclanthology.org/2022.emnlp-main.674.

[180] Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura

OMahony, et al. Aya dataset: An open-access collection for multilingual instruction tuning. *arXiv preprint arXiv:2402.06619*, 2024.

[181] Dmitry Kalashnikov, Jake Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Scaling up multi-task robotic reinforcement learning. In *Conference on Robot Learning*, pages 557–575. PMLR, 2022.

[182] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023.

[183] Razvan-Gabriel Cirstea, Darius-Valer Micu, Gabriel-Marcel Muresan, Chenjuan Guo, and Bin Yang. Correlated time series forecasting using multi-task deep neural networks. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 1527–1530, 2018.

[184] Zekai Chen, E Jiaze, Xiao Zhang, Hao Sheng, and Xiuzheng Cheng. Multi-task time series forecasting with shared attention. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 917–925. IEEE, 2020.

[185] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[186] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=lXuByUeHhd`.

[187] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina Mcmillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco de Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin,

Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro von Werra, Leon Weber, Long Phan, Loubna Ben Allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-Shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh Hajihosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne

Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael Mckenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel de Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-Aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. working paper or preprint, November 2023. URL `https://inria.hal.science/hal-03850124`.

[188] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

[189] Yuval Kirstain, Patrick Lewis, Sebastian Riedel, and Omer Levy. A few more examples may be worth billions of parameters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1017–1029, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.72. URL `https://aclanthology.org/2022.findings-emnlp.72`.

[190] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-*

seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

[191] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

[192] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020.

[193] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

[194] Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Scaling expert language models with unsupervised domain discovery, 2023.

[195] Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Shivanshu Purohit, Tri Songz, Wang Phil, and Samuel Weinbach. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch, 8 2021. URL https://www.github.com/eleutherai/gpt-neox.

[196] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[197] Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. Scaling up models and data with t5x and seqio, 2022.

[198] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2022.

[199] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

[200] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022. URL https://arxiv.org/abs/2204.06745.

[201] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

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

[203] Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

[204] Danqi Chen and Wen-tau Yih. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-tutorials.8. URL https://aclanthology.org/2020.acl-tutorials.8.

[205] Jinhyuk Lee, Sean S. Yi, Minbyul Jeong, Mujeen Sung, WonJin Yoon, Yonghwa Choi, Miyoung Ko, and Jaewoo Kang. Answering questions on COVID-19 in real-time. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online, December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlpcovid19-2.1. URL https://aclanthology.org/2020.nlpcovid19-2.1.

[206] Sharon Levy, Kevin Mo, Wenhan Xiong, and William Yang Wang. Open-Domain question-Answering for COVID-19 and other emergent domains. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 259–266, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-demo.30. URL https://aclanthology.org/2021.emnlp-demo.30.

[207] Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. *npj Digital Medicine*, 4(1):68, Apr 2021. ISSN 2398-6352. doi: 10.1038/s41746-021-00437-0. URL https://doi.org/10.1038/s41746-021-00437-0.

[208] Akari Asai, Jungo Kasai, Jonathan H. Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. XOR QA: Cross-lingual open-retrieval question answering. In *NAACL-HLT*, 2021.

[209] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset. *ArXiv*, page arXiv:2004.10706v2, Apr 2020. ISSN 2331-8422. URL https://pubmed.ncbi.nlm.nih.gov/32510522.

[210] Xinliang Frederick Zhang, Heming Sun, Xiang Yue, Simon Lin, and Huan Sun. COUGH: A challenge dataset and models for COVID-19 FAQ retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 3759–3769, 2021.

[211] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[212] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-4020.

[213] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

[214] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://www.aclweb.org/anthology/2020.emnlp-main.550.

[215] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

[216] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[217] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *ACL*, 2020.

[218] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, July 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.nlpcovid19-acl.18.

[219] Hao Fu, Yao; Peng and Tushar Khot. How does gpt obtain its ability? tracing emergent abilities of language models to their sources. *Yao Fu's Notion*, Dec 2022. URL https://yaofu.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-t

[220] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

[221] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. 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 9525–9587. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

[222] Margaret Mitchell, Alexandra Sasha Luccioni, Nathan Lambert, Marissa Gerchick, Angelina McMillan-Major, Ezinwanne Ozoani, Nazneen Rajani, Tristan Thrush, Yacine Jernite, and Douwe Kiela. Measuring data. *arXiv preprint arXiv:2212.05129*, 2022.

[223] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2020. URL https://api.semanticscholar.org/CorpusID:229156229.