Eliciting and Understanding Cross-Task Skills with Task-Level Mixture-of-Experts

Qinyuan Ye Juan Zha Xiang Ren
University of Southern California
{qinyuany, juanzha, xiangren}@usc.edu

Abstract

Recent works suggest that transformer models are capable of multi-tasking on diverse NLP tasks and adapting to new tasks efficiently. However, the potential of these multi-task models may be limited as they use the *same* set of parameters for *all* tasks. In contrast, humans tackle tasks in a more flexible way, by making proper presumptions on what skills and knowledge are relevant and executing only the necessary computations. Inspired by this, we propose to use task-level mixture-of-expert models, which has a collection of transformer layers (*i.e.*, experts) and a router component that chooses from these experts dynamically and flexibly. We find that these models help improve the average performance gain (ARG) metric by 2.6% when adapting to unseen tasks in the few-shot setting and by 5.6% in the zero-shot generalization setting. Further, we show that the learned routing decisions partly rediscover human categorization of NLP tasks – certain experts are strongly associated with extractive tasks, some with classification tasks, and some with tasks requiring world knowledge.[1]

1 Introduction

Pre-trained transformer models (Devlin et al., 2019; Liu et al., 2019b) have demonstrated remarkable capabilities in natural language processing (NLP) in recent years. Moreover, generative transformers can be viewed as a universal model that can be optimized for any language task primed into text-to-text format (Raffel et al., 2020). Recently, researchers found that training these transformer models to multi-task on a diverse collection of NLP tasks is beneficial – not only are they better at handling seen tasks (Aghajanyan et al., 2021; Aribandi et al., 2022), but also at generalizing and adapting to unseen tasks (Wei et al., 2021; Sanh et al., 2022).

[1]Our code will be released at https://github.com/INK-USC/CrossTaskMoE.

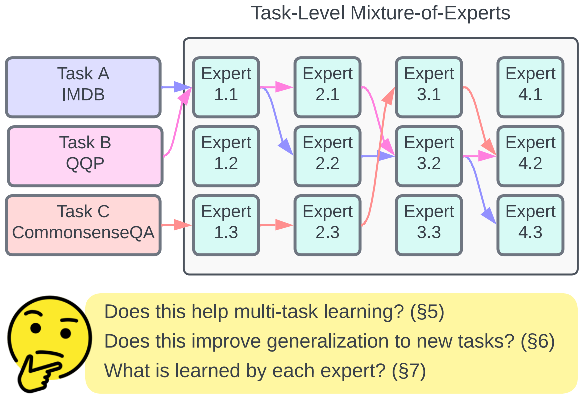

Figure 1: **Illustration of Task-level Mixture-of-Expert Models.** In this work, we train such models to multi-task on diverse NLP tasks, aiming at modeling skill sharing explicitly and understanding the learned patterns.

However, little is known about how multi-tasking capabilities and cross-task generalization is achieved, especially that the exact *same* set of weights is applied, and the *same* computation is executed, for very *different* tasks. Humans, on the other hand, do not exhaust their brain capacity for every task at hand. Humans develop skill sets and accumulate knowledge during learning, and can reuse and recompose them when facing a task. Inspired by this, we hypothesize that a model that explicitly emulate skill and knowledge sharing may help improve multi-task performance and generalization to new tasks. A natural fit for this goal would be task-level mixture-of-expert models (Jacobs et al., 1991; Kudugunta et al., 2021), where the model computation is conditioned on the task at hand. More specifically, the model contains a collection of experts and a router that chooses from the experts and composes the final model (Fig. 1-2).

In this paper, we first empirically investigate several key design choices for effectively training task-level mixture-of-experts models (§5). We further test the model's task-level generalization capabilities by testing it on unseen tasks (§6). Compared

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2567–2592
December 7-11, 2022 ©2022 Association for Computational Linguistics

to a multi-task BART-Base (Lewis et al., 2020) baseline, our final method leads to an 2.6% improvement in the average performance gain (ARG) metric when adapting to 18 unseen tasks (Ye et al., 2021) in the few-shot learning setting. Further, a gain of 5.6% in ARG is obtained in the zero-shot setting with P3 dataset (Sanh et al., 2022). Lastly, we conduct a detailed analysis quantifying the correlations between the learned routes and the characteristics of tasks (§7). We find that the routing decisions, though learned purely from multi-tasking *without* prior knowledge, strongly correlate with human understanding of task characteristics, such as the task being a classification task, the task being extractive, or the task requiring world knowledge.

2 Related Work

Massive Multi-task Learning. Multi-task learning (Caruana, 1997) has been continuously explored in NLP and is shown to be beneficial (McCann et al., 2018; Liu et al., 2019a). Recently, multi-task learning in NLP is brought to a new scale by using a significantly larger collection of tasks and examples (Aghajanyan et al., 2021; Aribandi et al., 2022; Khashabi et al., 2020; Hendrycks et al., 2021). These work demonstrate that multi-task learning improves the learning of text representation and thus boost the performance of seen tasks. Moreover, these models also exhibit strong adaptability to unseen tasks, in both few-shot (Ye et al., 2021) and zero-shot settings (Wei et al., 2021; Sanh et al., 2022; Mishra et al., 2021). Despite their effectiveness in terms of performance, how a model learns and spontaneously develops language skills during multi-task learning is a relatively underexplored topic. In our work, we try to investigate this question by training task-level MoE models and interpreting them. We additionally discuss contemporary works (Ponti et al., 2022; Gupta et al., 2022; Asai et al., 2022) in Appendix D.

Mixture-of-Experts in NLP. Mixture-of-experts models (Jacobs et al., 1991) divide the problem space into several sub-spaces and allow experts to be specialized in each subspace. Recently this concept is successfully applied to NLP (Shazeer et al., 2017), enabling models of billion or even trillion parameter scale (Fedus et al., 2021; Du et al., 2021; Artetxe et al., 2021; Zoph et al., 2022). However these applications mainly focus on the *scaling* aspects. Besides, most of them select experts on a per-example or per-token basis. In this work we are interested in multi-task learning with per-task gating decisions (Rosenbaum et al., 2018; Kudugunta et al., 2021), and mainly focus on understanding and interpreting task transferability.

Task Transferability in NLP. Phang et al. (2018) explored supplementary training on intermediate tasks (STILT), *i.e.*, training on a data-rich intermediate task before fine-tuning on the target task. STILT improves performance on the target task and stabilizes the fine-tuning process. Pruksachatkun et al. (2020) and Vu et al. (2020) further investigated when and why intermediate task transfer works. These studies mainly focus on transferability between specific *source-target pairs*, while we consider a more general setting of transferring within and beyond a *group* of NLP tasks.

3 Problem Setting

Our goal is to better understand multi-task learning with mixture-of-experts models with an explicit routing mechanism. We also hypothesize that such models help improve the model's capability to generalize/adapt to new tasks. Our problem setting closely resembles CrossFit (Ye et al., 2021). In the following, we introduce data usage (§3.1), training procedure (§3.2), and evaluation protocol (§3.3).

3.1 Data Usage

Assume that we have a collection of diverse NLP tasks $\mathcal{T}$, partitioned into two non-overlapping sets $(\mathcal{T}_{train}, \mathcal{T}_{test})$. These sets are also referred to as (Meta-Train, Meta-Test). $\mathcal{T}_{train}$ is mainly used for multi-task learning; $\mathcal{T}_{test}$ is used to quantify the model's adaptability to new tasks. Each task $T \in \mathcal{T}$ has three subsets, i.e., $T = (D_{train}, D_{dev}, D_{test})$. Additionally, we assume that all tasks are cast to a unified text-to-text format, *i.e.*, $D = \{(x, y)\}$, where x is the input text sequence, and y is the output text sequence.

3.2 Training Procedure

The training procedure has two stages: (1) an **upstream learning stage** for multi-task learning on T_{train}, to develop the skills that are needed to solve different tasks; and (2) a **downstream fine-tuning stage** on T_{test}, for evaluating the model's ability to adapt to new tasks. During the upstream learning stage, the model is expected to be trained for multi-task learning with the D_{train} from tasks in $\mathcal{T}_{train}$. D_{dev} for tasks in $\mathcal{T}_{train}$ will be used for hyperparameter tuning and model selection. During the downstream fine-tuning stage, the model will be fine-tuned on each task in $\mathcal{T}_{test}$ respectively. D_{train} will be used for fine-tuning, D_{dev} for validation, and D_{test} for reporting the final performance.

3.3 Evaluation Protocol

Each task in $\mathcal{T}$ has a pre-defined evaluation metric. For example, F1 score for classification tasks, and accuracy for multi-choice QA tasks. During the upstream learning stage, for simplicity, the model is validated on the *average* D_{dev} performance on all tasks in $\mathcal{T}_{train}$, and we report *average* D_{dev} performance and D_{test} performance. During the downstream fine-tuning stage, we compare the model's performance to the baseline of fine-tuning a vanilla transformer (without upstream learning), and compute the average relative performance gain (ARG) as our evaluation metric. More details about the baselines and ARG are deferred to §6.

4 Task-level MoE Transformers

Recall that our goal is to better elicit transferable skills during multi-task learning, and understand how those skills contribute the model performance. For this purpose we develop a mixture-of-experts variant of text-to-text transformer models, conditioning on task representations. The model contains two major components: (1) **a router** that selects and decides which experts to use for each task in each layer, based on its task representation; (2) **a collection of experts** that are dynamically composed into a final model based on the router selection. See Fig. 2 for a detailed illustration.

In the following, we introduce the router and the experts in more details. Note that we provide a general description in this section, and leave specific design choices in §5.3 for empirical comparison.

Collection of Experts. In an *original* implementation of text-to-text models (Raffel et al., 2020; Lewis et al., 2020), there are n transformer layers stacked and executed sequentially. The first $n/2$ layers are encoder layers and the last $n/2$ layers are decoder layers. In *our variant* of transformer models, we copied each layer for m times, resulting in $m * n$ experts in total. We refer to the j-th expert in the i-th layer as $E^{(i,j)}$. Note that we assume that each transformer block is an expert, which is different from Kudugunta et al. (2021). This is to make whole model dynamic and conpositional.

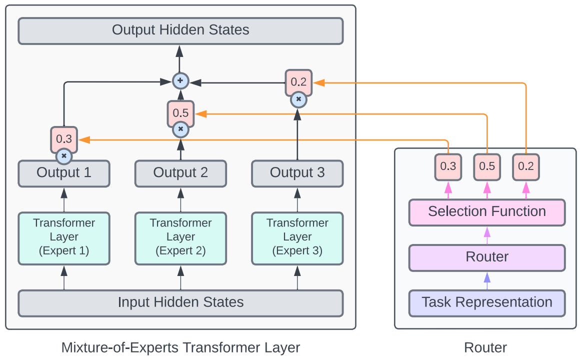

Figure 2: **Task-level Mixture-of-experts Transformer models used in this study. Right:** A router takes in a task representation and make decisions on expert selection. **Left:** the weighted sum of the outputs from each expert are considered the final output for this layer.

Router. For a given task $T_k \in \mathcal{T}$, with k as its task index, the router first takes the task representation ($\mathbf{T}_k$) from a look-up embedding table ($\mathbf{T}$). The router network outputs a matrix $\mathbf{L} \in \mathbb{R}^{m\times n}$, where $\mathbf{L}_{i,j}$ represents the logits of using expert $E^{(i,j)}$ in layer i. $\mathbf{L}$ goes through a selection function f to normalize the routing decisions in each layer, resulting in a final decision matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$.

Task-level MoE Transformers. We use the decision matrix $\mathbf{D}$ from the router to control the computation conducted by the experts. More specifically, in layer i, given input hidden states $\mathbf{h}_{in}^{(i)}$, the output $\mathbf{h}_{out}^{(i)}$ would be the weighted sum of all experts in the layer, and the weights are specified in $\mathbf{D}_{i,\cdot}$, *i.e.*,

$$\mathbf{h}_{out}^{(i)} = \sum_{j=1}^{m} \mathbf{D}_{i,j} E^{(i,j)}(\mathbf{h}_{in}^{(i)}) \qquad (1)$$

5 Applying Task-level MoE Models to Multi-task Learning

In our pilot studies, we found it is non-trivial to train these mixture-of-experts models properly and effectively. In this section, we present a detailed empirical study on baselines and design choices. We first introduce experiment details in §5.1. We then start with investigating simple baselines such as random or average routing (§5.2), which will help navigate our experiments on *learning* task-level MoE models. In §5.3 we introduce different variants we experiment with for learning task-level MoEs, and we summarize our findings in §5.4.

5.1 Experiment Details

Data. We previously discussed that a collection of diverse NLP tasks is required for the purpose of our study (§3.1). In our experiments, we use the task collection in CrossFit (Ye et al., 2021), which contains NLP tasks covering a wide range of task formats, goals and domains. We use its random task partition, with 120 tasks in T_{train} and 18 tasks in T_{test}. All tasks are converted to a unified text-to-text format and sub-sampled to be few-shot[2]. Details about the tasks are listed in Appendix E-F.

Model and Its Initialization. We previously introduced the model architecture of task-level MoEs in §4. In our experiments, the model is instantiated with the pre-trained BART-Base model (Lewis et al., 2020), a 12-layer encoder-decoder transformer model ($n = 12$). All m experts in layer i are initialized from the i-th layer of the BART-Base model. Additionally we add a Gaussian noise with variance of 1e-8 to the weights of each expert to avoid symmetry. We manually set the number of experts per layer $m = 3$ to allow sufficient flexibility while maintain a tractable model size.

Training Details. Deferred in Appendix B.1.

5.2 Investigation on Baselines

Before we experiment with learning routers, we first launch a series of baseline experiments related to the task-level MoE architecture. The goal is to get insights to help us better design our final model. We experiment with (1) Vanilla transformer, where mixture-of-experts are not involved; (2) Instance-level random routing, where the routes are randomly sampled for *each instance* during the forward pass; (3) Task-level random routing, where routes are sampled for *each task* once before training; (4) Average routing, where each experts were assigned the same weight in Eq. (1), *i.e.*, $\mathbf{D}_{i,j} = 1/3$. For (2) and (3), we try random selecting either one or two out of the three experts in each layer (denoted as "1/3" and "2/3"). In the case of "2/3", the output is the average of the outputs produced by the activated experts.

Findings. Performance of these baseline models are in top 2 sections in Table 1. We also plot the dev loss and performance curves during vanilla baseline training in Fig. 6 in Appendix C.1. We have the following findings.

(1) In Fig. 6, we found that dev losses dip in the early phase of training, then gradually rise. Meanwhile, the dev performance continue to increase. This is an important lesson learned for comparing different design choices: the simple and faster heuristic of model selection based on dev loss may be sub-optimal. We hypothesize this is because the text generation loss may not align well with the final evaluation metric[3].

(2) All random routing methods (except for "Random Task 2/3") leads to worsened performance compared to vanilla transformer baselines. This suggests that introducing sparsity and routing mechanism into transformer models naively can in fact hurt performance. This may be due to underfitting (the number of examples routed to each expert is reduced) or asynchronism in optimization (a different collection of experts is activated and updated at each optimization step).

(3) The observation that Random Task Routing (2/3) is better than Vanilla and Average Routing suggests that task interference exists in multi-task models with *fully* shared parameters, and allowing task-specific computations (as in Random Task 2/3) can be helpful. The observation that Random Task 2/3 is better than 1/3 suggests that performance is highly sensitive to the portion of shared vs. task-specific parameters. There is a fine line between MoE mechanism being helpful or being intrusive, adding difficulty to *training* MoE models.

5.3 Investigation on Design Choices

In the following we describe the key design choices we compared in training task-level MoEs.

Expert Selection. The selection function f is responsible for normalizing and discretizing (if necessary) the logit output of router network into final decisions. We consider three variants: (a) Softmax, the default design in most MoE models. (b) Gumbel-Softmax (Jang et al., 2016), which add gumbel-distributed noise to the logits and promote discrete decisions. (c) Gumbel-Softmax ST, where ST stands for straight-through estimator. For (b) and (c), we apply the temperature annealing mechanism to encourage exploration in the beginning of training.

Router Architecture. Router is a key component for our MoE model which computes the logits

[2]For classification tasks, there are 16 examples per task in D_{train}; for non-classification tasks, D_{train} has 32 examples.

[3]This finding is relevant to Csordás et al. (2021) which advocates proper validation protocol.

of selecting experts based on input task representations (see §4). We consider three router architecture with different complexities: (d) MLP, which contains two dense layers separated by GELU activation. (e) Bi-LSTM, which takes the sum of the task representation and a positional embedding as input at each time step (*i.e.*, layer). One linear layer is used to project the LSTM states to routing decisions. (f) Transformer (Vaswani et al., 2017), which takes the same input as Bi-LSTM and applies one single transformer encoder layer.

Task Representations. Vu et al. (2020) suggest that pre-computed task representations contain rich information for predicting task transferability. Here we consider incorporating these task representations as the initialization for the look-up embedding table $\mathbf{T}$ in our model (§4). In particular, we consider: (g) Random, which initialized every task representation with a randomly initialized 768d vector. (h) TextEmb, which is produced by encoding the input text with a pre-trained BART-Base model and taking the representations of the last encoder layer. We tried both the average representation of all tokens in the sequence (AVG) and BOS token representation. (i) FT-TextEmb, which is mostly identical to (h), despite that the BART-Base model is first fine-tuned on the D_{train} of the current task. (j) Fisher-TaskEmb (Vu et al., 2020), which is the diagonal of fisher information of the trainable parameters in a model. We use adapter (Houlsby et al., 2019) fine-tuning on D_{train} and compute the fisher information on these adapter parameters to avoid expensive computations.

Freezing Task Representations. Since adaptability to unseen task will be considered in later parts of this study, we further consider between (k) not freezing and (l) freezing the task representations during multi-task learning. We conjecture that the structure of seen task representations may be changed after multi-task learning, while the unseen task representations may not reflect the change; hence the freezing variant.

Two-stage Training. In §5.2, we find that introducing routing mechanism naively may lead to worsened performance. Also, average routing is stable and achieves competitive performance. Based on these observations, we design a two-stage training strategy to combine the benefits of both methods. In the first stage, the model jointly learns the router and the experts. In the second stage, the experts are re-initialized from BART's pre-trained weights, and the routes gradually transforms from average routing to the learned routes by controlling the temperature used in the softmax function. As a result, in the beginning of the training, the temperature is set to be high, so the router is functioning like average routing; during the training process, the temperature decreases gradually, and the router will give more discrete routing decisions.

Model	Compute	Dev (%)	Test (%)
Vanilla Transformers			
(1) BART-Base	1x	54.47±0.05	48.93±0.23
(1) BART-Large	-	58.10±0.20	54.06±0.22
Baselines			
(2) Random Inst. Routing (1/3)	1x	47.50±0.20	41.87±0.76
(2) Random Inst. Routing (2/3)	2x	44.81±1.76	38.48±1.00
(3) Random Task Routing (1/3)	1x	52.89±0.57	47.27±0.35
(3) Random Task Routing (2/3)	2x	**55.35±0.23**	**50.44±0.29**
(4) Average Routing (3/3)	3x	54.61±0.11	50.02±0.19
Task-level Mixture-of-Experts			
(c) + (d) + (g) + (k) + (n)	1x	**55.28±0.12**	**50.52±0.38**
(c) + (d) + (j) + (k) + (m)	1x	53.07±0.45	48.16±0.34
(c) + (d) + (j) + (l) + (m)	1x	53.06±0.19	47.64±0.79
(c) + (d) + (j) + (l) + (n)	1x	**55.40±0.08**	**50.39±0.68**

Table 1: **Performance on baselines and selected models.** Average performance on D_{dev}/D_{test} over tasks in $\mathcal{T}_{train}$ are reported. Average and standard dev. are computed based on runs with three different random seeds.

Model	Dev (%)	Model	Dev (%)
Expert Selection		Task Repr. (cont.)	
(a) Softmax	40.93	(i) FT-TextEmb-BOS	52.93
(b) Gumbel-Softmax	52.02	(i) FT-TextEmb-AVG	53.29
(c) Gumbel-Softmax ST	53.14	(j) Fisher-TaskEmb	53.51
Router Architecture		Freeze Task Repr.	
(d) MLP	53.14	(k) Not Freezing	53.51
(e) LSTM	53.55	(l) Freezing	53.37
(f) Transformer	53.13	-	-
Task Repr.		Two-stage Training	
(g) Random	53.14	(m) Use one stage	53.51
(h) TextEmb-BOS	52.51	(n) Use two stages	55.36
(h) TextEmb-AVG	53.30	-	-

Table 2: **Investigation on Design Choices.** By default the model uses (c) + (d) + (g) + (k) + (m) when comparing different choices in each colored section.

5.4 Results and Findings

We first present the performance of variants mentioned above in Table 2. For the best-performing model variants, we run three times with different random seeds to reduce variance in performance (Table 1, Bottom). We have the following observations. **(1) What helps?** We found that the choice of selection function and the two-stage learning procedure are important for training task-level MoEs. Gumbel-Softmax with straight-through estimator achieves the best performance among the three choices.[4] Two-stage training helps improve performance by 1.8%.[5] **(2) What doesn't help?** We did not observe significant difference with choices in router architecture or task representation initialization. Supposedly, LSTMs and transformers are able to capture relations more complicated than MLPs, and precomputed task representations carry richer information about the task than random initialization. This unexpected observation suggests that the router struggle to leverage task-level information with the current training methods and supervision signals. **(3) Comparing with the baselines.** Our best task-level MoE using random initialized task representations ((c) + (d) + (g) + (k) + (n)) can rival the best baselines in §5.2 (Random Task Routing 2/3), while using half of its computation in a forward pass. With careful design, task-level MoEs are beneficial for multi-task learning.

6 Generalizing to Unseen Tasks

We hypothesize that task-level MoE models can recombine the learned skills effectively when they encounter new tasks. In §6.1 we evaluate the models obtained in §5 on adapting to new tasks in a few-shot learning setting. In §6.2 we further extend our method to a zero-shot learning setting and test it on the P3 dataset (Sanh et al., 2022).

6.1 Few-shot Adaptation

Compared Methods. We use the following models as initialization for few-shot fine-tuning on unseen tasks (T_{test}). **(1) Direct Fine-tuning.** For each unseen task, we fine-tune the off-the-shelf BART-Base model with its D_{train}. **(2) Multi-task BART.** We take the multi-task BART-Base from §5 as initialization and fine-tune the model on D_{train}. **(3) Baseline Routing BART.** We re-use the models using random task routing (1/3, 2/3) and average routing in §5. **(4) Learned Routing BART.** We take the (c) + (d) + (j) + (l) + (n) model from §5. This models uses fisher information as the task representation (j) and the representations for seen tasks are frozen (l) during multi-task learning. For the unseen task, we first compute its fisher information based on D_{train} and feed it to the learned router to select experts. We then fine-tune the selected experts on D_{train}.

[4] See Appendix C.3 for further investigation.

[5] We also use heterogeneous batching (Aghajanyan et al., 2021) and two-speed learning rate (Ponti et al., 2022) in our model as recommended by these works.

Data and Evaluation. We use the 18 unseen tasks specified in CrossFit random partition in Ye et al. (2021)[6]. We first obtain the performance of fine-tuning the pre-trained BART-Base model as the baseline. Then we compute and report the average relative gain (ARG) over pre-trained BART for the multi-task BART and routing BART methods. For example, if fine-tuning pre-trained BART achieves 50% accuracy on task A and 80% F1 on task B, and fine-tuning multi-task BART achieves 80% accuracy on task A and 60% F1 on task B, the ARG would be the average of $(80\% - 50\%)/50\%$ and $(60\% - 80\%)/80\%$, which equals to 17.5%.

Results. We present the performance gains on individual tasks and their average in Fig. 3. Multi-task BART remains a strong baseline, achieving an ARG of 9.74%. Random task routing (2/3) and average routing baselines achieves 10.21% and 8.06% respectively. Our task-level MoE model (c) + (d) + (j) + (l) + (n) achieves the best average performance gain (12.30%), which is 2.6% higher than the multi-task BART. We observe that negative transfers are alleviated and few-shot performance are improved compared to the baselines for many tasks. This suggest that our task-level MoE model is learning reusable experts and meaningful routes.

6.2 Zero-shot Generalization

In this section, we modify our proposed method to zero-shot learning settings where each unseen task has no labeled data. We use Public Pool of Prompts (P3) dataset as our testbed (Sanh et al., 2022).

Data. Following Sanh et al. (2022); Bach et al. (2022), we use the prompt templates to change texts from various NLP tasks into a unified text-to-text formats. Specifically, we have 36 upstream tasks for $\mathcal{T}_{train}$, and 10 tasks for $\mathcal{T}_{test}$. We use accuracy as the evaluation metric. We report both the average performance on $\mathcal{T}_{test}$ (AVG), and the average performance gain (ARG) described in §6.1.

Compared Methods. For all the models, we train on the D_{train} for all tasks in $\mathcal{T}_{train}$, and directly test the model on D_{test} for each task in $\mathcal{T}_{test}$. We mainly compare four methods: **(1)** Multi-task BART-Base. **(2)** Random Task Routing (2/3). **(3)** We train a new (c) + (d) + (h) + (l) + (m) model on P3 data. **(4)** Similar to (3), we train a model with the configuration of (c) + (d) + (h) + (l) + (n).

[6] We exclude Free-base QA and Yelp Polarity from the evaluation as performance is unusually unstable on these tasks.

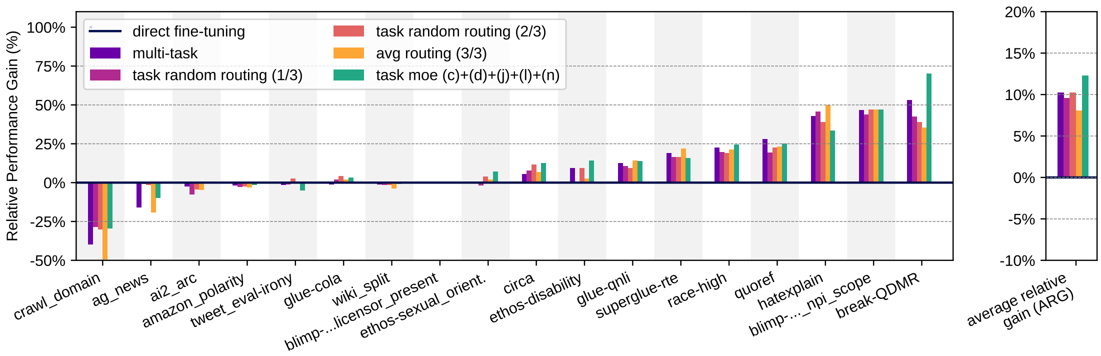

Figure 3: **Few-shot Performance on Unseen Tasks.** Bar heights represent relative performance gain over directly fine-tuning a pre-trained BART-Base model. The right-most bars are the average performance gain.

Main models	anli_r3	HellaSwag	cb	wic	wsc	winogrande	arc-chan.	obqa	piqa	SQuADv2	AVG	ARG
Multi-task BART-Base	27.6	22.0	44.6	43.1	57.5	52.7	23.1	26.2	26.4	14.6	33.7	-
Random Task Routing (2/3)	23.7	14.5	19.3	37.8	45.2	49.0	14.5	18.5	3.5	9.1	23.5	-33.6
(c) + (d) + (h) + (l) + (m)	33.7	20.7	43.6	40.4	50.2	46.8	11.8	18.1	22.4	18.6	32.2	-8.3
(c) + (d) + (h) + (l) + (n)	32.0	23.7	44.3	43.4	56.5	52.2	21.1	28.5	30.2	17.6	34.9	5.6

Table 3: **Zero-shot Performance on Unseen Tasks**. Accuracy (%) on the test set of 10 unseen tasks. We compare the AVG and calculate the ARG of routing model (c) + (d) + (h) + (l) + (m) and (c) + (d) + (h) + (l) + (n) over multi-task BART-Base. The former routing model uses one-stage training while the latter uses two-stage straining.

Note that in the zero-shot setting, we cannot use pre-computed task representations for unseen tasks based on labeled examples (as described in §5.3). Therefore for (h) TextEmb used in **(3)** and **(4)**, we encode prompt templates as the auxiliary task information. More details are in Appendix B.3.

Results. We present the results in Table 3. Our findings are: **(1)** Compared to the multi-task BART-base baseline with an AVG of 33.7%, our routing model (4) achieves a higher AVG (34.9%) and a positive ARG (5.6%). This demonstrates the model's improved generalization ability to novel tasks in the zero-shot setting. **(2)** The gap between model (3) and model (4) shows that the two-stage training strategy is essential in the zero-shot setting as well. **(3)** Different from the findings in the few-shot setting, Random Task Routing (2/3) has a negative ARG (-33.6%). Without labeled data in unseen tasks, random routing cannot actively select relevant experts or update model parameters, resulting in worsened performance. In contrast, task-level MoE has the flexibility to select relevant experts and achieves better performance.

7 Interpreting the Routes and Experts

7.1 Learning Dynamics of the Routes

We visualized the learned routing decisions of the (c) + (d) + (g) + (k) + (m) model trained on CrossFit data in Fig. 4. Note that (g) represents that the task representations are randomly initialized and learned spontaneously during multi-task learning. We observe that distinct patterns for classification and generation tasks emerge in the early stage of the training (step 3000). These patterns transition from coarse-grained to fine-grained gradually in the training process. These observations align with our expectation that task-level MoEs are learning to share parameters for similar tasks and avoid interference among dissimilar tasks.

7.2 Correlation with Task Features

To better understand the learned routing decisions, we investigate the relation between the routing decisions and manually-defined task features. In the following, we first describe the methodology of computing correlation, then describe the features we investigate, and finally describe our findings.

Method. For each task in $\mathcal{T}_{train}$, we first compute the routing decisions $\mathbf{D} \in \mathbb{R}^{m \times n}$ using the learned model. For each expert $E^{(i,j)}$, we consider the routing decision $\mathbf{D}_{i,j}$ of all tasks as a feature. Altogether, we have $m \times n$ features of dimension $|\mathcal{T}_{train}|$ (the number of tasks). Additionally, we have t manually-defined features on all tasks, giving t features of dimension $|\mathcal{T}_{train}|$. We compute Pearson correlation coefficient between each pair of learned routing decisions and manual feature, resulting in a $\mathbb{R}^{mn \times t}$ matrix quantifying the correlation between $m \times n$ experts and t manual features.

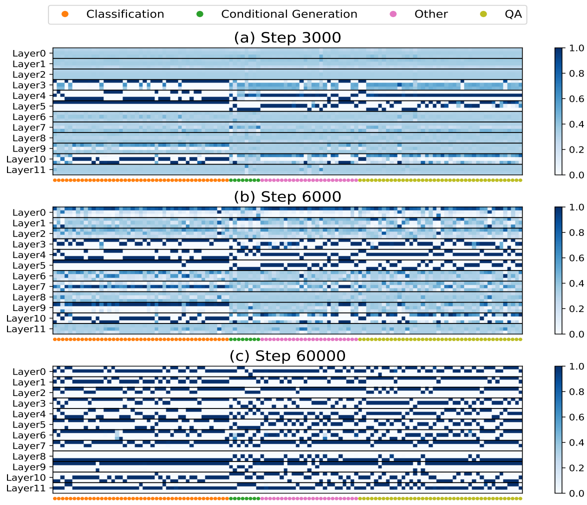

Figure 4: **Routing Decisions Learned During Multi-task Learning** ((c) + (d) + (g) + (k) + (m)). The router is able to distinguish classification tasks from other types of tasks after 3000 steps of the training. It then gradually learns more fine-grained patterns.

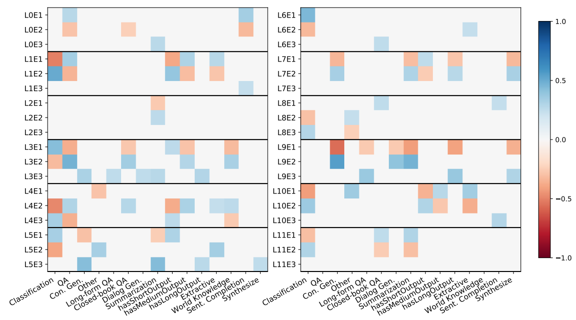

Figure 5: **Pearson Correlation Between Learned Routes and Selected Manual Features.** Correlation with p<0.01 are visualized. “L0E1” stands for expert 1 in layer 0. The correlation is computed based on a (c) + (d) + (g) + (k) model, where (g) means the task embedding table $\mathbf{T}$ is randomly initialized. This suggests that without prior knowledge of the tasks, the router can partially rediscover human categorization of tasks during multi-task learning.

Feature Name	Example	Description
Task Format		
Extractive	SQuAD, Race	Output is always a substring of the input
Sentence Completion	HellaSwag, LAMA-Probes	Requires the model to fill in a blank in the input or continue to generate based on the input
Required Skills and Knowledge		
Linguistic	Blimp, CoLA	Tasks focusing on grammatical correctness, semantic equivalence and linguistic phenomenon
Commonsense	CommonsenseQA	Tasks testing commonsense knowledge and reasoning capabilities
Co-reference	Wino_grande	Tasks requiring co-reference resolution
Multi-hop Reasoning	DROP	Tasks requiring multi-hop/multi-step reasoning
Implicit Knowledge	TriviaQA	Tasks requiring world knowledge (acquired during pre-training)
Synthesize	Break, XSum	Combining ideas and allowing an evolving understanding of text

Table 4: Additional Features on Format, High-level Skills and Knowledge.

Manual Features. We consider the following features in our correlation study[7]. The final feature table ($t \times |\mathcal{T}_{train}|$) is in Table 9.

- **Task Format.** We use the task categories provided in Ye et al. (2021). The top-level labels include `Classification`, `Question Answering`, `Conditional Generation`, and `Others`. Tasks in each category are divided into sub-categories. For example, QA tasks are further categorized into `machine reading comprehension (MRC)`, `multiple-choice QA`, `closed-book QA`, etc.
- **Input/Output Length.** We classify tasks with into three features based on their average input length: `hasShortInput` (shortest 25%), `hasLongInput` (longest 25%), `hasMediumInput` (remainder). We also classify tasks into three features based on their average output length: `hasShortOutput` (< 3 tokens), `hasLongOutput` (> 10 tokens), and `hasMediumOutput` (remainder).
- **Text Domain.** We categorize tasks with into domains such as `Science & Technology`, `Social Network`, `News`, `Web`, `Bio-Medical`, `Review`, `Dialog`, and `Books`.
- **Granularity.** We categorize tasks into `Span-level` (*e.g.*, acronym identification); `Sentence-level` (*e.g.*, tweet classification); `Paragraph-level` (*e.g.*, news summarization) based on their main focus. This is different from input length.
- **Additional Features: Format, High-level Skills and Knowledge[8].** We additionally describe several common task characteristics in Table 4. These include whether a task is `Extractive`, requires `Sentence Completion`, or requires high-level skills such as `Co-reference`.

Findings. Results on selected features are visualized in Fig. 5. Visualization of the complete pairs of expert and feature are in Fig. 7-8. We have the following observations: **(1)** There exists strong correlation between several pairs of routing decisions and manual features. For example, L1E2, L3E1, L6E1 are positively correlated with the feature of Classification, suggesting that these experts are

[7] We admit that several categorization criteria are subjective and they are by no means exhaustive for fully describing a task. We use these features mainly to quantify the relation between human understanding of tasks and the learned routes.

[8] These features are mostly inspired by dataset papers such as SQuAD (Rajpurkar et al., 2016), BLiMP (Warstadt et al., 2020), MNLI (Williams et al., 2018), HotpotQA (Yang et al., 2018), CommonsenseQA (Talmor et al., 2019).

Manual Feature	Top3 Exp	Task	All	~~Top1~~	~~Top3~~
Classification	L1E2	imdb	92.49	91.87	88.70
	L6E1	sms spam	63.54	63.54	62.88
	L3E1	emo	82.06	65.46	16.22
Conditional Generation	L9E2	gigaword	30.00	26.51	17.91
	L5E3	aeslc	14.52	15.31	14.76
	L7E2	kilt_wow	6.39	6.01	4.73
Closed-book QA	L3E2	kilt_trex	31.85	25.63	28.13
	L4E2	kilt_zsre	13.13	11.25	9.38
	L6E3	numer_sense	34.38	33.75	20.00

Table 5: **Performance when top correlated experts are disabled.** "~~Top1~~" means the most positively correlated expert is disabled. Performance gradually drops as more experts are disabled.

Task	All	~~Top1~~	~~Top3~~	~~Rand1~~	~~Rand3~~	~~Least1~~	~~Least3~~
imdb	92.49	91.87	88.70	92.49	91.66	92.49	92.49
sms spam	63.54	63.54	62.88	63.54	63.53	63.54	63.54
emo	82.06	65.46	16.22	82.06	64.13	82.06	82.06

Table 6: **Disabling top/least correlated experts and random experts.** The experts that positively correlate (Top1/Top3) with the "classification" feature contribute more to the performance than randomly selected or least correlated experts (Least1/Least3).

likely to be selected for classification tasks. **(2)** The correlations are strongest with the top-level task category features (*i.e.*, `Classification`, `QA`, `Conditional Generation`), suggesting that the router may understand and categorize tasks in a way similar to us. **(3)** However, correlation does not imply causal relationships. The correlation patterns of `Classification` and `hasShortOutput` are similar, the same applies to `Conditional Generation` and `hasLongOutput`. We cannot conclude whether the router is making router decisions depending on output length, task format, or other hidden aspects.

7.3 Expert Disabling Experiments

We further examine the learned task-level MoE models by disabling experts during evaluation. By "disabling", we simply set the pre-softmax logit to be $-\infty$, so that the second-best expert in that layer will be selected instead. We hypothesize that if an expert corresponds to a critical skill required by a certain type of tasks, then disabling it should bring significant performance drop. **(1)** We select three manual features: `Classification`, `Conditional Generation`, `Closed-book QA`, and select three tasks that belong to these categories. We select the top 3 experts that positively correlate with these features, and disable them during evaluation. Results are listed in Table 5. As expected, these correlated experts are indispensable for the

Task	All	♢ ~~Top1~~	♢ ~~Top3~~	♡ ~~Top1~~	♡ ~~Top3~~
♢ imdb	92.49	91.87	88.70	92.49	92.49
♢ emo	82.06	65.46	16.22	82.06	82.06
♡ kilt_zsre	13.13	13.13	12.50	11.25	9.38
♡ numer_sense	34.38	34.38	34.38	33.75	20.00

Table 7: **Disabling experts associated with different task categories.** ♢=Classification, ♡=Closed-book QA. Performance does not drop significantly when experts relevant to other features are disabled (red area).

task performance. Performance gradually drops as more experts are disabled (All $\rightarrow$ ~~Top1~~ $\rightarrow$ ~~Top3~~). **(2)** For the three classification tasks we select, we further compare the performance when disabling most/least correlated experts and random experts. Results are presented in in Table 6. Results suggest experts that are positively correlated with the classification feature are more important to the final performance. **(3)** We further take two classification tasks (♢) and two closed-book QA tasks (♡), and consider disabling experts correlated with classification and closed-book feature. Results are shown in Table 7. Performance are not influenced significantly when experts relevant to other features are disabled. **To conclude**, this set of experiments suggests that experts that positively correlate with a specific type of tasks are irreplaceable; they greatly contribute to the performance of that type of tasks.

8 Conclusions

Inspired by how humans accumulate skills from past experience and re-use them to solve new tasks, in this paper, we develop and conduct extensive experiments with transformer-based task-level mixture-of-expert (MoE) models, in hope to provide new insights on multi-task learning and cross-task generalization in NLP. Firstly, we empirically investigate importance design choices and quantify their influence on final model. Secondly, in both few-shot and zero-shot settings, we demonstrate that task-level mixture-of-expert models are better at generalizing to new tasks. Finally, by conducting a detailed analysis on the routing decisions, we find they have strong correlations with human-defined task characteristics, even when the decisions are learned spontaneously without no prior knowledge such as pre-computed task representations. We hope our work provide useful advice on training and interpreting multi-task models in NLP and we hope it will inspire future work in improving multi-task learning and cross-task generalization in NLP.

Limitations

Although we have done much analysis on the correlation between learned routes and task characteristics, it is yet challenging to (1) ground each expert to human-understandable language skills; (2) understand their causal relationships. Much more needs to be discussed on how to systematically define the atomic/basic skills that are used in solving NLP tasks. In terms of model optimization, we find that we cannot achieve the best performance using the one-stage training strategy, and our best method takes more training time and needs more delicate hyper-parameters selection compared to the vanilla multi-task model. We hypothesize that there are optimization challenges in training task-level mixture-of-expert models. We hope future work can investigate and address this problem.

Acknowledgments

We thank authors and crowd-workers of all datasets used in our study. We thank huggingface datasets team (Lhoest et al., 2021) for making NLP datasets more accessible. We thank anonymous reviewers, members of USC INK Lab and USC NLP community for their valuable feedback. This work is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007; the DARPA MCS program under Contract No. N660011924033; the Defense Advanced Research Projects Agency with award W911NF-19-20271; NSF IIS 2048211.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6429–6438. IEEE.

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. 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. Association for Computational Linguistics.

Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, page 259–262, New York, NY, USA. Association for Computing Machinery.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

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. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona T. Diab, Zornitsa Kozareva, and Ves Stoyanov. 2021. Efficient large scale language modeling with mixtures of experts. *CoRR*, abs/2112.10684.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attentional mixtures of soft prompt tuning for parameter-efficient multi-task knowledge sharing. *arXiv preprint arXiv:2205.11961*.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, volume 6, pages 6–4. Venice.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages

1644–1650, Online. Association for Computational Linguistics.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1122–1136, Online. Association for Computational Linguistics.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020a. MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6521–6532, Online. Association for Computational Linguistics.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 63–69, Minneapolis, USA. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact: A large-scale dataset for table-based fact verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124.

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

T. Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *ArXiv preprint*, abs/2012.00614.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Z. Chen, and Claire Cui. 2021. Glam: Efficient scaling of language models with mixture-of-experts. *ArXiv*, abs/2112.06905.

Matthew Dunn, Levent Sagun, Mike Higgins, V. U. Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv preprint*, abs/1704.05179.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803, Brussels, Belgium. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on*

New Frontiers in Summarization, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. 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, Montréal, Canada. Association for Computational Linguistics.

Shashank Gupta, Subhabrata Mukherjee, Krishan Subudhi, Eduardo Gonzalez, Damien Jose, Ahmed H Awadallah, and Jianfeng Gao. 2022. Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892. Text Mining and Natural Language Processing in Pharmacogenomics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

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

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online. Association for Computational Linguistics.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. 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 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,* *(AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3577–3599. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, D. Kontokostas, Pablo N. Mendes, Sebastian Hellmann, M. Morsey, Patrick van Kleef, S. Auer, and C. Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

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

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020a. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised cross-task generalization via retrieval augmentation. *arXiv preprint arXiv:2204.07937*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang

Ren. 2020b. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7411–7425, Online. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*, 65(4):782–796.

Irene Manotas, Ngoc Phuoc An Vo, and Vadim Sheinin. 2020. LiMiT: The literal motion in text dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 991–1000, Online. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *ArXiv preprint*, abs/2012.10289.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 165–172. ACM.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: Matching user questions to COVID-19 faqs. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3458–3465. ACM.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *ArXiv preprint*, abs/2006.08328.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gu lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

A. Othman and M. Jemni. 2012. English-asl gloss parallel corpus 2012: Aslg-pc12. In *IEnglish-ASL Gloss Parallel Corpus 2012*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Dimitris Pappas, Petros Stavropoulos, Ion Androutsopoulos, and Ryan McDonald. 2020. BioMRC: A dataset for biomedical machine reading comprehension. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 140–149, Online. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Edoardo M Ponti, Alessandro Sordoni, and Siva Reddy. 2022. Combining modular skills in multitask learning. *arXiv preprint arXiv:2202.13914*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, and Thien Huu Nguyen. 2020. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3285–3301, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

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

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8722–8731.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. 2018. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *International Conference on Learning Representations*.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

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. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR (Poster)*. OpenReview.net.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 93–106, Barcelona, Spain (Online). Association for Computational Linguistics.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. 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 3477–3486, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. Quarel: A dataset and models for answering questions about qualitative relationships. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7063–7071.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. 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 5941–5946, Hong Kong, China. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085, Hong Kong, China. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

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

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Jixuan Wang, Kuan-Chieh Wang, Frank Rudzicz, and Michael Brudno. 2021. Grad2task: Improved few-shot text classification using gradients for task representation. In *Advances in Neural Information Processing Systems*.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hao Zhang, Jae Ro, and Richard Sproat. 2020. Semi-supervised URL segmentation with recurrent neural networks pre-trained on knowledge graph entities. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4667–4675, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 446–456, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, X. Liu, J. Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *ArXiv preprint*, abs/1810.12885.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language usin. *ArXiv preprint*, abs/1709.00103.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. 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 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*.

A Computing Task Representations

In the following, we describe the method to construct the task representations used in §5.3.

TaskEmb. Task2Vec (Achille et al., 2019) is a method to generate tasks embedding for visual classification tasks based fisher information matrix (FIM). It was then extended to NLP domain (Vu et al., 2020; Wang et al., 2021) and was found to be useful. We compute the empirical fisher and use them as task representations, following Vu et al. (2020). Specifically, given a model P_θ parameterized by θ (*e.g.*, a BART-Base model) and a set of labeled examples $\{(x, y)\}$, we first fine-tune the model on the examples, then compute the fisher information matrix:

$$F_\theta = \frac{1}{n}\sum_{i=1}^{n}\left[\nabla_\theta \log P_\theta\left(y^i|x^i\right)\nabla_\theta \log P_\theta\left(y^i|x^i\right)^T\right] \tag{2}$$

To reduce the computational complexity, (1) we only use the diagonal entries of F_θ, following Achille et al. (2019) and Vu et al. (2020); (2) we use a parameter-efficient fine-tuning method named adapter fine-tuning (Houlsby et al., 2019) and only compute the FIM with respect to adapter parameters. (3) we use PCA to reduce the dimension ($d = 768$, which is the same as TextEmb), as we will use these representations as input to our router in the task-level MoE model.

TextEmb and FT-TextEmb. For TextEmb, we first concatenate the input sequence x and the output sequence y into a longer sequence, and feed it to the encoder of BART to get token-level representations. For TextEmb-AVG, we compute the average over tokens for each example, and then average over all examples, to get a final vector as task representation. For TextEmb-BOS, we average the BOS representation of all examples[9]. For fair comparison with TaskEmb, which fine-tunes the model on labeled examples and thus may obtain extra information through this process, we also include FT-TextEmb-AVG and FT-TextEmb-BOS in our comparison. In these two variants, the BART model is first fine-tune on the labeled examples $\{(x, y)\}$.

B Additional Experiment Details

B.1 Multi-task Learning Experiments

We concatenate the D_{train} of the 120 tasks in $\mathcal{T}_{train}$ into a large dataset and use it for multi-task learning. We adopt heterogeneous batching (Aghajanyan et al., 2021), *i.e.*, each batch contains examples from different tasks. For the vanilla multi-task baseline, we train the model for 30,000 steps, with the batch size equals to 32 and the learning rate equals to 3e-5. For BART-Large we use the same setting, except that the learning rate is set to 1e-5. We use validation every 3,000 steps and select the best model based on validation performance.

For the task-level MoE models, they are trained with a basic learning rate of 1e-5, while we set the router with bigger learning rate of 1e-3 based on our pilot experiments following Ponti et al. (2022). For the task representations, we use 1e-2 as learning rate when they are randomly initialized, and 1e-3 when initialized from pre-computed representations. We train the model for 60,000 steps because it takes more exploration time for the routes and experts to be stable. All models are trained with Adam optimizer (Kingma and Ba, 2014).

B.2 Few-shot Adaptation Experiments

For few-shot fine-tuning we mainly follow the experiment setting in Ye et al. (2021). Each task has five different few-shot samples of (D_{train}, D_{dev}). We train on D_{train} for 1000 steps, and validate on D_{dev} every 100 steps. We run a grid search for learning rate {1e-5, 2e-5, 5e-5} and batch size {2,4,8} for each few-shot sample. Finally, the model with best D_{dev} performance is evaluated on D_{test}, the we report the performance on D_{test}.

B.3 Zero-shot Experiments

Data. Following Sanh et al. (2022) and Lin et al. (2022), we use the prompt templates in the Public Pool of Prompts (P3) (Bach et al., 2022) to change texts from various NLP tasks into a unified text-to-text format. To save compute, we use a sub-sampled version of P3 dataset. We use up to 5k examples for D_{train}, 1k examples for both D_{dev} and D_{test} following Lin et al. (2022) for all tasks. We use 36 upstream tasks (which is the same as T0 upstream learning) for $\mathcal{T}_{train}$ and use 10 unseen tasks as our $\mathcal{T}_{test}$. D_{train} for tasks in $\mathcal{T}_{train}$ are used for upstream learning; D_{test} for tasks in $\mathcal{T}_{test}$ are used for reporting the performance. For simplicity, we only keep the prompt that can be evaluated with accuracy, and we report the mean acurracy for all tasks in $\mathcal{T}_{test}$.

Training. (1) For Multi-task BART-Base and Random Task Routing (2/3), we use 1e-5 as the learning rate, 16 as training batch size, and the total training steps is set to be 200k. (2) For the (c) + (d) + (h) + (l) + (m) model, we use 1e-5 as the base learning rate for experts and 1e-3 for the router. We train the model for 200k steps. (3) For the (c) + (d) + (h) + (l) + (n) model, we use 1e-5 as the base learning rate for experts and 1e-3 for router. For the first learning stage we train for 60k steps, and 200k steps for the second stage. For both MoE models we use a batch size as 4. In this zero-shot setting, the task representation is computed by applying TextEmb-AVG (h) to the prompt templates.

C Extended Results and Analysis

C.1 Loss and Performance Discrepancy

In Fig. 6, we plot the D_{dev} loss and performance during multitask learning. We conclude that D_{dev} loss does not align well with the final metrics, and thus validation should be done with the final metrics.

C.2 Full Manual Feature Correlation Results

We show the full results of Pearson Correlation between learned routes and manual features in Figure 7 and Figure 8. Figure 7 is based on routes in the (c) + (d) + (g) + (k) model, and Figure 8 is based on the (c) + (d) + (j) + (k) model.

C.3 Further Investigation on Selection Functions

In our initial experiments, the implementation of softmax does not have temperature annealing.

[9] We later found out that this is less meaningful since BART pre-training does not train these BOS tokens with any special objective.

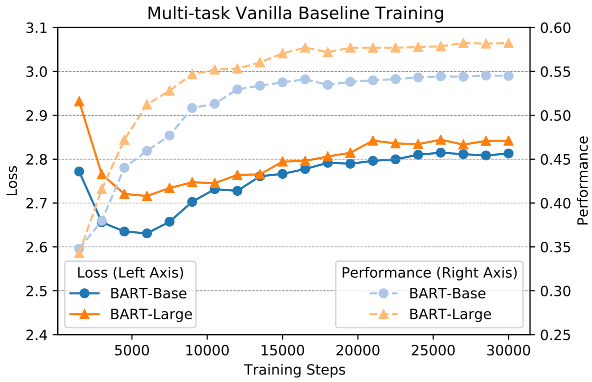

Figure 6: **Dev loss and dev performance discrepancy when training multi-task transformer baselines.** We found that smaller dev loss does not guarantee better dev performance. Dev losses tend to plunge then rise, while dev performance continue to increase. BART-Large outperforms BART-Base despite larger dev loss.

When we include this trick, the performance is comparable to gumbel-softmax ST.

D Discussion on Contemporary Works

Training dynamical models that condition the computation on task information is a growing and active research field. Several contemporary works (Ponti et al., 2022; Gupta et al., 2022; Asai et al., 2022) are studying this problem. We share similar motivations with these works; meanwhile, these works and ours differ in methodology and research focus. We would like to highlight that (1) we conduct extensive analysis on interpreting the learned routes and experts in §7; (2) we use 120 seen tasks and 18 unseen tasks, which is more diverse, and creates a challenging learning setting. We hope our findings are useful to the EMNLP community.

Figure 7: **Pearson Correlation Between Learned Routes and Manual Features.** Correlation with p<0.01 are visualized. The correlation is based on a (c) + (d) + (g) + (k) model.

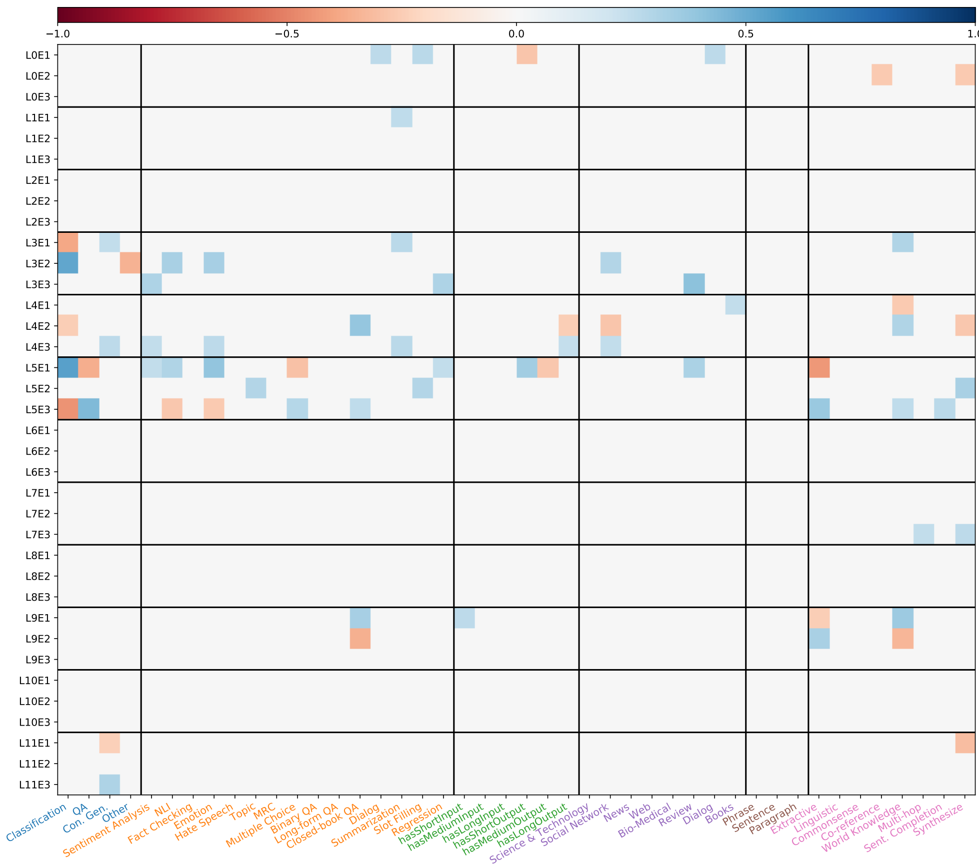

Figure 8: **Pearson Correlation Between Learned Routes and Manual Features.** Correlation with p<0.01 are visualized. The correlation is based on a (c) + (d) + (j) + (k) model.

E Tasks Used and References

We list all the tasks used in this paper in Table 8 and its corresponding manual feature labels in Table 9.

Table 8: Tasks used in this work.

Task Name	Ontology	Reference
acronym_identification	other	Pouran Ben Veyseh et al. 2020
ade_corpus_v2-classification	cls/other	Gurulingappa et al. 2012
ade_corpus_v2-dosage	other/slot filling	Gurulingappa et al. 2012
ade_corpus_v2-effect	other/slot filling	Gurulingappa et al. 2012
adversarialqa	qa/machine reading comprehension	Bartolo et al. 2020
aeslc	cg/summarization	Zhang and Tetreault 2019
ag_news	cls/topic	Gulli (link)
ai2_arc	qa/multiple-choice qa	Clark et al. 2018
amazon_polarity	cls/sentiment analysis	McAuley and Leskovec 2013
anli	cls/nli	Nie et al. 2020
app_reviews	other/regression	Missing
aqua_rat	qa/multiple-choice qa	Ling et al. 2017
art (abductive nli)	other	Bhagavatula et al. 2020
aslg_pc12	other	Othman and Jemni 2012
biomrc	qa/machine reading comprehension	Pappas et al. 2020
blimp-anaphor_gender_agreement	other/linguistic phenomenon	Warstadt et al. 2020
blimp-anaphor_number_agreement	other/linguistic phenomenon	Warstadt et al. 2020
blimp-determiner_noun_agreement_with_adj_irregular_1	other/linguistic phenomenon	Warstadt et al. 2020
blimp-ellipsis_n_bar_1	other/linguistic phenomenon	Warstadt et al. 2020
blimp-ellipsis_n_bar_2	other/linguistic phenomenon	Warstadt et al. 2020
blimp-existential_there_quantifiers_1	other/linguistic phenomenon	Warstadt et al. 2020
blimp-irregular_past_participle_adjectives	other/linguistic phenomenon	Warstadt et al. 2020
blimp-sentential_negation_npi_licensor_present	other/linguistic phenomenon	Warstadt et al. 2020
blimp-sentential_negation_npi_scope	other/linguistic phenomenon	Warstadt et al. 2020
blimp-wh_questions_object_gap	other/linguistic phenomenon	Warstadt et al. 2020
boolq	qa/binary	Clark et al. 2019
break-QDMR	other	Wolfson et al. 2020
break-QDMR-high-level	other	Wolfson et al. 2020
circa	cls/other	Louis et al. 2020
climate_fever	cls/fact checking	Diggelmann et al. 2020
codah	qa/multiple-choice qa	Chen et al. 2019
common_gen	other	Lin et al. 2020b
commonsense_qa	qa/multiple-choice qa	Talmor et al. 2019
cos_e	other/generate explanation	Rajani et al. 2019
cosmos_qa	qa/multiple-choice qa	Huang et al. 2019
crawl_domain	other	Zhang et al. 2020
crows_pairs	other	Nangia et al. 2020
dbpedia_14	cls/topic	Lehmann et al. 2015
definite_pronoun_resolution	other	Rahman and Ng 2012
discovery	cls/other	Sileo et al. 2019
dream	qa/multiple-choice qa	Sun et al. 2019
duorc	qa/machine reading comprehension	Saha et al. 2018
e2e_nlg_cleaned	other	Dušek et al. 2020, 2019
eli5-askh	qa/long-form qa	Fan et al. 2019
eli5-asks	qa/long-form qa	Fan et al. 2019
eli5-eli5	qa/long-form qa	Fan et al. 2019
emo	cls/emotion	Chatterjee et al. 2019
emotion	cls/emotion	Saravia et al. 2018
empathetic_dialogues	cg/dialogue	Rashkin et al. 2019
ethos-directed_vs_generalized	cls/hate speech detection	Mollas et al. 2020
ethos-disability	cls/hate speech detection	Mollas et al. 2020
ethos-gender	cls/hate speech detection	Mollas et al. 2020
ethos-national_origin	cls/hate speech detection	Mollas et al. 2020
ethos-race	cls/hate speech detection	Mollas et al. 2020
ethos-religion	cls/hate speech detection	Mollas et al. 2020
ethos-sexual_orientation	cls/hate speech detection	Mollas et al. 2020
financial_phrasebank	cls/sentiment analysis	Malo et al. 2014
freebase_qa	qa/closed-book qa	Jiang et al. 2019
gigaword	cg/summarization	Napoles et al. 2012
glue-cola	cls/other	Warstadt et al. 2019
glue-mnli	cls/nli	Williams et al. 2018
glue-mrpc	cls/paraphrase	Dolan and Brockett 2005
glue-qnli	cls/nli	Rajpurkar et al. 2016
glue-qqp	cls/paraphrase	(link)
glue-rte	cls/nli	Dagan et al. 2005; Bar-Haim et al. 2006 Giampiccolo et al. 2007; Bentivogli et al. 2009
glue-sst2	cls/sentiment analysis	Socher et al. 2013
glue-wnli	cls/nli	Levesque et al. 2012
google_wellformed_query	cls/other	Faruqui and Das 2018
hate_speech18	cls/hate speech detection	de Gibert et al. 2018
hate_speech_offensive	cls/hate speech detection	Davidson et al. 2017
hatexplain	cls/hate speech detection	Mathew et al. 2020
health_fact	cls/fact checking	Kotonya and Toni 2020
hellaswag	qa/multiple-choice qa	Zellers et al. 2019
hotpot_qa	qa/machine reading comprehension	Yang et al. 2018
imdb	cls/sentiment analysis	Maas et al. 2011

Continued on next page

Task Name	Ontology	Reference
jeopardy	qa/closed-book qa	(link)
kilt_ay2	other/entity linking	Hoffart et al. 2011
kilt_fever	cls/fact checking	Thorne et al. 2018
kilt_hotpotqa	qa/closed-book qa	Yang et al. 2018
kilt_nq	qa/closed-book qa	Kwiatkowski et al. 2019
kilt_trex	qa/closed-book qa	Elsahar et al. 2018
kilt_wow	cg/dialogue	Dinan et al. 2019
kilt_zsre	qa/closed-book qa	Levy et al. 2017
lama-conceptnet	qa/closed-book qa	Petroni et al. 2019, 2020
lama-google_re	qa/closed-book qa	Petroni et al. 2019, 2020
lama-squad	qa/closed-book qa	Petroni et al. 2019, 2020
lama-trex	qa/closed-book qa	Petroni et al. 2019, 2020
liar	cls/fact checking	Wang 2017
limit	other	Manotas et al. 2020
math_qa	qa/multiple-choice qa	Amini et al. 2019
mc_taco	qa/binary	Zhou et al. 2019
medical_questions_pairs	cls/paraphrase	McCreery et al. 2020
mocha	other/regression	Chen et al. 2020a
multi_news	cg/summarization	Fabbri et al. 2019
numer_sense	qa/closed-book qa	Lin et al. 2020a
onestop_english	cls/other	Vajjala and Lučić 2018
openbookqa	qa/multiple-choice qa	Mihaylov et al. 2018
paws	cls/paraphrase	Zhang et al. 2019
piqa	other	Bisk et al. 2020
poem_sentiment	cls/sentiment analysis	Sheng and Uthus 2020
proto_qa	other	Boratko et al. 2020
qa_srl	other	He et al. 2015
qasc	qa/multiple-choice qa	Khot et al. 2020
quail	qa/multiple-choice qa	Rogers et al. 2020
quarel	qa/multiple-choice qa	Tafjord et al. 2019a
quartz-no_knowledge	qa/multiple-choice qa	Tafjord et al. 2019b
quartz-with_knowledge	qa/multiple-choice qa	Tafjord et al. 2019b
quoref	qa/machine reading comprehension	Dasigi et al. 2019
race-high	qa/multiple-choice qa	Lai et al. 2017
race-middle	qa/multiple-choice qa	Lai et al. 2017
reddit_tifu-title	cg/summarization	Kim et al. 2019
reddit_tifu-tldr	cg/summarization	Kim et al. 2019
ropes	qa/machine reading comprehension	Lin et al. 2019
rotten_tomatoes	cls/sentiment analysis	Pang and Lee 2005
samsum	cg/summarization	Gliwa et al. 2019
scicite	cls/other	Cohan et al. 2019
sciq	qa/multiple-choice qa	Welbl et al. 2017
scitail	cls/nli	Khot et al. 2018
search_qa	qa/closed-book qa	Dunn et al. 2017
sick	cls/nli	Marelli et al. 2014
sms_spam	cls/other	Almeida et al. 2011
social_i_qa	qa/multiple-choice qa	Sap et al. 2019
spider	cg/other	Yu et al. 2018
squad-no_context	qa/closed-book qa	Rajpurkar et al. 2016
squad-with_context	qa/machine reading comprehension	Rajpurkar et al. 2016
superglue-cb	cls/nli	de Marneffe et al. 2019
superglue-copa	qa/multiple-choice qa	Gordon et al. 2012
superglue-multirc	qa/multiple-choice qa	Khashabi et al. 2018
superglue-record	qa/machine reading comprehension	Zhang et al. 2018
superglue-rte	cls/nli	Dagan et al. 2005; Bar-Haim et al. 2006 Giampiccolo et al. 2007; Bentivogli et al. 2009
superglue-wic	cls/other	Pilehvar and Camacho-Collados 2019
superglue-wsc	cls/other	Levesque et al. 2012
swag	qa/multiple-choice qa	Zellers et al. 2018
tab_fact	cls/fact checking	Chen et al. 2020b
trec	cls/other	Li and Roth 2002; Hovy et al. 2001
trec-finegrained	cls/other	Li and Roth 2002; Hovy et al. 2001
tweet_eval-emoji	cls/emotion	Barbieri et al. 2020
tweet_eval-emotion	cls/emotion	Barbieri et al. 2020
tweet_eval-hate	cls/emotion	Barbieri et al. 2020
tweet_eval-irony	cls/emotion	Barbieri et al. 2020
tweet_eval-offensive	cls/emotion	Barbieri et al. 2020
tweet_eval-sentiment	cls/emotion	Barbieri et al. 2020
tweet_eval-stance_abortion	cls/emotion	Barbieri et al. 2020
tweet_eval-stance_atheism	cls/emotion	Barbieri et al. 2020
tweet_eval-stance_climate	cls/emotion	Barbieri et al. 2020
tweet_eval-stance_feminist	cls/emotion	Barbieri et al. 2020
tweet_eval-stance_hillary	cls/emotion	Barbieri et al. 2020
tweet_qa	qa/machine reading comprehension	Xiong et al. 2019
web_questions	qa/closed-book qa	Berant et al. 2013
wiki_auto	cls/other	Jiang et al. 2020
wiki_bio	cg/other	Lebret et al. 2016
wiki_qa	cls/other	Yang et al. 2015
wiki_split	cg/other	Botha et al. 2018
wikisql	cg/other	Zhong et al. 2017
wino_grande	qa/multiple-choice qa	Sakaguchi et al. 2020
wiqa	qa/multiple-choice qa	Tandon et al. 2019
xsum	cg/summarization	Narayan et al. 2018
yahoo_answers_topics	cls/topic	(link)

Continued on next page

Task Name	Ontology	Reference
yelp_polarity	cls/sentiment analysis	Zhang et al. 2015; (link)
yelp_review_full	other/regression	Zhang et al. 2015; (link)
cnn_dailymail	cg/summarization	Nallapati et al. 2016
wiki_hop	qa/multiple-choice qa	Welbl et al. 2018

F Random Task Partition

Different from the original random task partition used in Ye et al. (2021), we remove yelp_polarity and freebase_qa from $\mathcal{T}_{test}$ because we observe unusual instability when doing few-shot fine-tuning on these tasks.

```
{
    "train": ['glue-mrpc', 'math_qa', 'quarel', 'e2e_nlg_cleaned', 'tweet_eval-stance_atheism', 'lama-squad', 'tab_fact', 'aqua_rat', 'tweet_eval-emoji', 'glue-wnli', 'codah', 'tweet_eval-offensive', 'wiki_qa', 'blimp-ellipsis_n_bar_1', 'openbookqa', 'sms_spam', 'acronym_identification', 'blimp-determiner_noun_agreement_with_adj_irregular_1', 'ethos-national_origin', 'spider', 'definite_pronoun_resolution', 'hellaswag', 'superglue-wsc', 'numer_sense', 'ade_corpus_v2-dosage', 'blimp-ellipsis_n_bar_2', 'kilt_ay2', 'squad-no_context', 'google_wellformed_query', 'xsum', 'wiqa', 'tweet_eval-stance_abortion', 'reddit_tifu-tldr', 'ade_corpus_v2-effect', 'qa_srl', 'ethos-religion', 'commonsense_qa', 'jeopardy', 'biomrc', 'superglue-multirc', 'ethos-race', 'eli5-askh', 'glue-qqp', 'paws', 'ethos-directed_vs_generalized', 'glue-sst2', 'mocha', 'tweet_eval-hate', 'glue-rte', 'blimp-anaphor_number_agreement', 'lama-conceptnet', 'hate_speech_offensive', 'superglue-wic', 'boolq', 'kilt_hotpotqa', 'quartz-no_knowledge', 'aslg_pc12', 'sick', 'tweet_eval-stance_climate', 'tweet_eval-sentiment', 'crows_pairs', 'glue-mnli', 'medical_questions_pairs', 'break-QDMR-high-level', 'qasc', 'imdb', 'ethos-gender', 'trec-finegrained', 'adversarialqa', 'onestop_english', 'web_questions', 'duorc', 'yelp_review_full', 'swag', 'proto_qa', 'scitail', 'tweet_eval-stance_feminist', 'limit', 'common_gen', 'scicite', 'blimp-irregular_past_participle_adjectives', 'social_i_qa', 'anli', 'kilt_zsre', 'cosmos_qa', 'superglue-record', 'squad-with_context', 'emotion', 'blimp-existential_there_quantifiers_1', 'race-middle', 'kilt_wow', 'sciq', 'wino_grande', 'rotten_tomatoes', 'superglue-cb', 'poem_sentiment', 'ropes', 'reddit_tifu-title', 'piqa', 'climate_fever', 'lama-google_re', 'search_qa', 'wiki_auto', 'mc_taco', 'blimp-wh_questions_object_gap', 'hotpot_qa', 'emo', 'kilt_nq', 'kilt_trex', 'quartz-with_knowledge', 'dbpedia_14', 'yahoo_answers_topics', 'app_reviews', 'superglue-copa', 'blimp-anaphor_gender_agreement', 'hate_speech18', 'gigaword', 'multi_news', 'aeslc', 'quail'],
    "dev": ['cos_e', 'kilt_fever', 'eli5-asks', 'trec', 'eli5-eli5', 'art', 'empathetic_dialogues', 'tweet_qa', 'wikisql', 'lama-trex', 'tweet_eval-stance_hillary', 'discovery', 'tweet_eval-emotion', 'liar', 'wiki_bio', 'dream', 'ade_corpus_v2-classification', 'health_fact', 'samsum', 'financial_phrasebank'],
    "test": ['quoref', 'wiki_split', 'ethos-disability', 'superglue-rte', 'glue-cola', 'ethos-sexual_orientation', 'blimp-sentential_negation_npi_scope', 'ai2_arc', 'amazon_polarity', 'race-high', 'blimp-sentential_negation_npi_licensor_present', 'tweet_eval-irony', 'break-QDMR', 'crawl_domain', 'glue-qnli', 'hatexplain', 'ag_news', 'circa'],
}
```

G Manually-Defined Features

Task Name	Science Technology	Social Network	News	Web	Bio-Medical	Review	Dialog	Books	Financial	Phrase	Sentence	Paragraph	Extractive	Linguistic	Commonsense	Co-reference	World Knowledge	Multi-hop	Sentence Completion	Synthesize
acronym_identification	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0
ade_corpus_v2-classification	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ade_corpus_v2-dosage	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
ade_corpus_v2-effect	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
adversarialqa	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0
aeslc	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
ag_news	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ai2_arc	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
amazon_polarity	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
anli	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
app_reviews	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
aqua_rat	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
art	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
aslg_pc12	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1
biomrc	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
blimp-anaphor_gender_agreement	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
blimp-anaphor_number_agreement	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-determiner_noun_agreement_with_adj_irregular_1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-ellipsis_n_bar_1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-ellipsis_n_bar_2	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-existential_there_quantifiers_1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-irregular_past_participle_adjectives	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-sentential_negation_npi_licensor_present	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-sentential_negation_npi_scope	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
blimp-wh_questions_object_gap	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
boolq	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
break-QDMR	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1
break-QDMR-high-level	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1
circa	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0
climate_fever	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
codah	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
common_gen	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
commonsense_qa	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
cos_e	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
cosmos_qa	0	1	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
crawl_domain	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1
crows_pairs	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
dbpedia_14	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
definite_pronoun_resolution	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0
discovery	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0
dream	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0
duorc	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0
e2e_nlg_cleaned	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
eli5-askh	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
eli5-asks	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
eli5-eli5	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0
emo	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
emotion	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
empathetic_dialogues	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-directed_vs_generalized	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-disability	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-gender	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-national_origin	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-race	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-religion	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ethos-sexual_orientation	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
financial_phrasebank	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
freebase_qa	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
gigaword	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
glue-cola	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
glue-mnli	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
glue-mrpc	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
glue-qnli	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
glue-qqp	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
glue-rte	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
glue-sst2	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
glue-wnli	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
google_wellformed_query	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
hate_speech18	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
hate_speech_offensive	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
hatexplain	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
health_fact	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
hellaswag	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
hotpot_qa	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
imdb	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
jeopardy	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
kilt_ay2	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
kilt_fever	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
kilt_hotpotqa	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
kilt_nq	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
kilt_trex	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
kilt_wow	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0
kilt_zsre	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
lama-conceptnet	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
lama-google_re	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
lama-squad	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
lama-trex	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
liar	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
limit	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0
math_qa	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
mc_taco	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
medical_questions_pairs	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
mocha	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
multi_news	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1
numer_sense	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0
onestop_english	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
openbookqa	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0
paws	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
piqa	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
poem_sentiment	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
proto_qa	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
qa_srl	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
qasc	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0
quail	0	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0
quarel	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
quartz-no_knowledge	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
quartz-with_knowledge	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
quoref	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
race-high	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
race-middle	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
reddit_tifu-title	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
reddit_tifu-tldr	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
ropes	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
rotten_tomatoes	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
samsum	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1
scicite	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
sciq	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
scitail	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
search_qa	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
sick	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
sms_spam	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
social_i_qa	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
spider	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
squad-no_context	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
squad-with_context	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
superglue-cb	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0
superglue-copa	0	1	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
superglue-multirc	1	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0
superglue-record	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
superglue-rte	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
superglue-wic	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
superglue-wsc	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
swag	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
tab_fact	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
trec	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
trec-finegrained	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-emoji	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
tweet_eval-emotion	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-hate	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-irony	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-offensive	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-sentiment	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-stance_abortion	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-stance_atheism	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-stance_climate	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-stance_feminist	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_eval-stance_hillary	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
tweet_qa	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
web_questions	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
wiki_auto	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
wiki_bio	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
wiki_qa	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
wiki_split	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1
wikisql	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
wino_grande	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0	1	0
wiqa	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
xsum	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
yahoo_answers_topics	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
yelp_polarity	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
yelp_review_full	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Table 9: Full feature table used for analysis in §7.