On the Compositional Generalization in Versatile Open-domain Dialogue

Tingchen Fu[1]†, Xueliang Zhao[2]†, Lemao Liu[3], Rui Yan[1,4]*
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]The University of Hong Kong [3]Tencent AI Lab
[4]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education
lucas.futingchen@gmail.com xlzhao22@connect.hku.hk
redmondliu@tencent.com ruiyan@ruc.edu.cn

Abstract

Previous research has demonstrated the potential of multi-task learning to foster a conversational agent's ability to acquire a variety of skills. However, these approaches either suffer from interference among different datasets (also known as negative transfer), or fail to effectively reuse knowledge and skills learned from other datasets. In contrast to previous works, we develop a sparsely activated modular network: (1) We propose a well-rounded set of operators and instantiate each operator with an independent module; (2) We formulate dialogue generation as the execution of a generated programme which recursively composes and assembles modules. Extensive experiments on 9 datasets verify the efficacy of our methods through automatic evaluation and human evaluation. Notably, our model outperforms state-of-the-art supervised approaches on 4 datasets with only 10% training data thanks to the modular architecture and multi-task learning. [1]

1 Introduction

Building an open-domain dialogue system is an intriguing and challenging task. A good open-domain chatbot should be equipped with a well-rounded set of skills (Roller et al., 2021) including but not limited to providing an informative response, showing different emotions, keeping a consistent persona and conducting commonsense inference. Up to now, with more and more datasets proposed to train multiple conversation skills (e.g., Wizard of Wikipedia (Dinan et al., 2019), Personachat (Zhang et al., 2018)), multi-task learning is an efficient way to grasp all the versatile skills and quickly transfer to newly emerging datasets (Roller et al., 2021).

However, as a core problem in multi-task learning, it is not easy to strike a balance between transfer and interference (negative transfer) among multiple datasets (Rosenbaum et al., 2019). For this, recent researchers mainly follow two lines of research. On one line, Roller et al. (2021) and Shuster et al. (2022a) simply mix all the datasets together to embody the blended skill required in dialogue. They update all the model parameters to minimize the loss of all of the data, which is also dubbed dense training (Gururangan et al., 2022). In spite of its simplicity, it easily incurs interference among different datasets (Aribandi et al., 2022). On the other line, Li and Liang (2021) learn multiple skills and store the knowledge from different datasets with different sets of parameter-efficient architectures. This approach eliminates underlying negative transfer among different corpora, but hinders positive transfer at cost. The model has to learn from scratch rather than reuse past knowledge every time a new corpus comes.

Inspired by recent advancements in neuroscience (Dehaene et al., 2021) suggesting that the human brain represents knowledge in a modular way, we incorporate this as an inductive bias and present a compositional modular architecture to balance transfer and interference (Rosenbaum et al., 2019). By decomposing the knowledge for dialogue into relative independent modules (Mittal et al., 2022), a neural model thus decides which module to invoke for different tasks or different samples. However, there are two challenges in applying modular architecture to building a versatile open-domain chatbot. First, the generation task is different from question answering, where the neural module network accomplishes impressive performance (Andreas et al., 2016; Hu et al., 2017; Gupta et al., 2020). It is untouched how to apply the ideology of modularity to the auto-regressive generation process. Second, the modules used in neural module network (Andreas et al., 2016) are typically

† Tingchen Fu and Xueliang Zhao contribute equally to this work. This work was done during their internship at Tencent AI Lab.

* Corresponding author: Rui Yan(ruiyan@ruc.edu.cn)

[1] The code is available at https://github.com/TingchenFu/ACL23-ModularDialogue

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13585–13605
July 9-14, 2023 ©2023 Association for Computational Linguistics

trained with end-task supervision. Without intermediate supervision or specialized training data for each module (Ponti et al., 2022), modules might perform homogeneous functions rather than perform their predefined functions as intended (Gupta et al., 2020, 2021).

To deal with the above problems, in this paper, we present a neural modular framework for blended-skill dialogue generation. The principle of our approach is to decompose the generation process into the recursive execution of basic operators by various modules. Specifically, (1) as an attempt to conduct generation tasks in a modular way, we introduce content modules for basic content synthesis and linguistics modules for linguistically-related surface realization. In addition, a programmer is trained to produce a reverse-polish-style code (Burks et al., 1954) which schedules the modules to produce the final response. (2) To overcome the homogeneity of modules, we construct pseudo labels and provide weak supervision signals to facilitate the training of each module. Since the output of the programmer and the modules are discrete and thus not differentiable, we employ Gumbel-Softmax trick to produce "soft" sentences as the output of modules at training time, and employ reinforcement learning to bridge the gap between the programmer and the modules.

Extensive experiments are conducted on 9 open-domain datasets. Our approach surpasses other models on a similar parameter scale and achieves a new state-of-the-art by multi-task training on all the 9 corpora. Notably, our model outperforms state-of-the-art supervised approaches on DailyDialog, EmpatheticDialog, LIGHT and Cornell Movie with only 10% training data, demonstrating that our modular framework could compose existing skills more efficiently to attain superior performance on out-of-distribution data.

2 Related Work

2.1 Open Domain Dialogue

Most early attempts on dialogue generation construct dialogue systems using manually created rules or templates (Weizenbaum, 1966; Wallace, 2009). The advancements in the field of machine translation (Ritter et al., 2011; Gehring et al., 2017; Vaswani et al., 2017) have served as inspiration for a number of explorations to construct end-to-end open-domain dialogue generation models (Shang et al., 2015; Vinyals and Le, 2015). Following that, the vanilla encoder-decoder architecture is widely employed to improve response quality, and it has undergone several revisions to enhance response diversity (Xing et al., 2017; Zhao et al., 2017; Tao et al., 2018), model conversation context structure (Xing et al., 2018; Zhang et al., 2019), and regulate response characteristics (Wang et al., 2018; See et al., 2019; Wang et al., 2020a). Smith et al. (2020) and Shuster et al. (2020) initiate the study of equipping the open-domain conversation agent with a well-rounded set of skills, whose key idea is to conduct simultaneous multi-task training on the blended data. These models have demonstrated encouraging results in skill blending and skill selection thanks to the careful design of the training scheme. BlenderBot (Roller et al., 2021) demonstrates how large-scale models can further promote the concurrent acquisition of several skills. BlenderBot 2.0 is created as a result of the additions made by Komeili et al. (2022) and Xu et al. (2022), who offer BlenderBot the capacity to access the Internet and memorize lengthy history respectively.

2.2 Multi-task Learning with Pre-trained Language Models

Multi-task learning is a common paradigm to transfer knowledge from multiple related tasks to enhance generalization capacity and has shown promising results in a variety of NLP tasks (Zhang and Yang, 2021; Crawshaw, 2020). Large-scale pre-trained language models (PLMs) have presented brand-new difficulties for multi-task learning. Aghajanyan et al. (2021) propose pre-finetuning which refines the pre-trained representations through massively multi-task learning. In spite of its efficiency, pre-finetuning may result in catastrophic forgetting of the pre-training task. To alleviate this issue, Aribandi et al. (2021) propose multi-task pre-training which bridges the gap between pre-training and finetuning data distributions. T0 (Sanh et al., 2021) is an early attempt to induce the zero-shot generalization capability of PLMs through explicit multi-task learning, which converts NLP tasks into a manually-collected prompted form. Another prevalent paradigm in multi-task learning using PLMs is instruction tuning, in which the PLMs encode task-specific instructions together with input and produce task output (Wei et al., 2021; Mishra et al., 2022; Wang et al., 2022). Despite promising results, these methods may suffer from the negative transfer problem

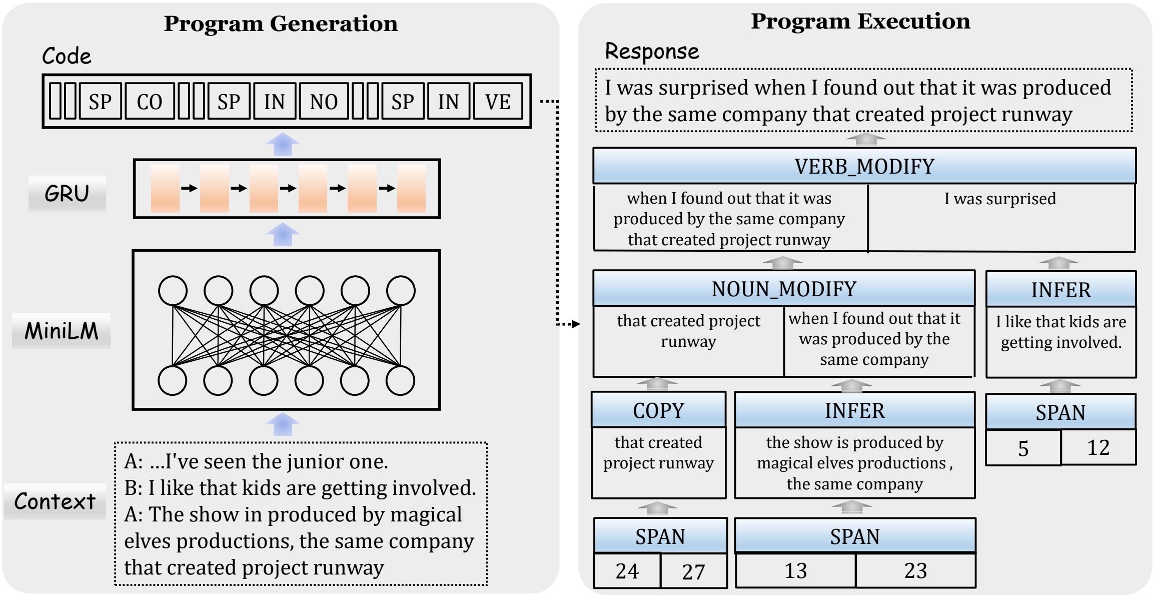
Program Generation
Code
SP
CO
SP
IN
NO
SP
IN
VE
GRU
MiniLM
Context
A: ...I've seen the junior one.
B: I like that kids are getting involved.
A: The show in produced by magical elves productions, the same company that created project runway
Program Execution
Response
I was surprised when I found out that it was produced by the same company that created project runway
VERB_MODIFY
when I found out that it was produced by the same company that created project runway
I was surprised
NOUN_MODIFY
that created project runway
when I found out that it was produced by the same company
INFER
I like that kids are getting involved.
COPY
that created project runway
INFER
the show is produced by magical elves productions , the same company
SPAN
5
12
SPAN
24
27
SPAN
13
23

Figure 1: The workflow of the proposed modular generation framework. We use the following abbreviations: SP=SPAN, CO=COPY, IN=INFER, NO=NOUN_MODIFY, VE=VERB_MODIFY. We omit the operands of SPAN from program generation to reduce redundancy.

due to the practice of activating all parameters for different tasks. To mitigate this issue, researchers have resorted to parameter-efficient methods which allocate separate adapters for each task (Mahabadi et al., 2021), and compositional modules which only activate relevant parts of the models (Ponti et al., 2022). Our method is orthogonal to earlier efforts in that it attempts to mitigate the unexplored negative transfer problem in auto-regressive decoding.

2.3 Neural Modular Network

The concept of neural module networks has drawn a lot of interest in a variety of computer vision and natural language processing tasks. Andreas et al. (2016) initially propose neural module network, which parses questions into linguistic substructures and builds question-specific deep networks from compositional modules, to conduct visual question answering. Following this work, several attempts have been made to eliminate the need for mediate supervision on semantic parsers (Hu et al., 2018; Mao et al., 2019), directly forecast the instance-specific network architectures in an end-to-end way (Hu et al., 2017), infer the answer with a purely symbolic executor (Yi et al., 2018), and perform visual co-reference resolution (Kottur et al., 2018). Gupta et al. (2020) and Chen et al. (2020) propose employing neural module networks in response to questions in machine reading comprehension. Another line of closely related works to ours are generative neural module networks, which activate a module when generating the next token (Yang et al., 2019; Tian and Oh, 2020) or only utilize modular architecture for encoder (Le et al., 2022). Our research differs significantly from theirs in that we break down dialogue response generation into independent operations in order to reduce catastrophic forgetting in each module.

3 Preliminary

For open-domain dialogue generation, each datum can be thought of a pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{y}$ is the response and $\boldsymbol{x}$ is the dialogue context composed of history utterances and other external resources such as background knowledge (CMU_DoG (Zhou et al., 2018b)), persona of speakers (ConvAI2 (Zhang et al., 2018)) or conversation setting (LIGHT (Urbanek et al., 2019)). The goal of an open-domain dialogue generation model is to generate $\boldsymbol{y}$ given $\boldsymbol{x}$ and exhibit the necessary skills to be more human-like.

In the proposed modular generation framework, a programmer $p_\theta(\boldsymbol{c}|\boldsymbol{x})$ takes the dialogue context as input and produces a code sequence $\boldsymbol{c} = [c_1, c_2, \cdots, c_n]$, where n is the length of the code. Based on the generated code, different modules are activated to perform different functions. The execution of the code produces a response in the end. The workflow of our framework is shown in Figure 1.

The rest of our paper is structured as follows.

We illustrate the modular architecture in §4, including the implementation of the programmer and execution of the code with modules. In §5, we elaborate the training algorithm to cope with the paucity of human annotation and discrete optimization. The experiment results and further analysis are displayed in §6 and §7 respectively.

4 Modular Framework

In this section, we elaborate on how the modular generation framework works. Briefly, a programmer first generates a code in a special reverse-polish-style programme language. Then we execute the code with a stack to store the intermediate result. When encountering some specific operators in code, we just activate corresponding modules to fulfill the function of the operator.

4.1 Module Definition

Our modules are devised to perform basic atomic tasks and realize the function of some operators. From another perspective, operators are high-level abstractions of modules. According to their functions and the format of input and output, there are 3 types of modules, namely span operator, content operator and linguistics operator. The Span operator is responsible for selecting a span from the dialogue context given the start index and the end index, whose role is similar to `QUESTION_SPAN` and `PASSAGE_SPAN` in Chen et al. (2020). Content operators (`COPY`, `PARAPHRASE` and `INFER`) generate diverse new content based on the input text. The linguistics operators (`COMPOUND`, `VERB_MODIFY` and `NOUN_MODIFY`) combine two texts together to form complex sentence or compound sentence. The computation result of the linguistics operators could also serve as the operands to other linguistics operators and content operators. We list all the operators used in our framework in Appendix A.

In the implementation, each content operator and linguistics operator are corresponding to an auto-regressive generation module $\mathcal{M}(y_i^m|x^m, y_{<i}^m)$, where x^m, y^m are the input and output of the module $\mathcal{M}$. They are parameterized as standard transformers. We initialize the parameters of these models using pre-trained T5-small (Raffel et al., 2019). Customizing different modules according to their intended purpose might lead to better performance, but we focus on the overall framework in this paper and leave the sophisticated design of modules for future work.

The key insight behind the module instantiation is to decompose the response generation process into relatively independent and composable pieces. Although our framework bears similarities with previous works in visual QA (Andreas et al., 2016; Hu et al., 2017) and image captioning (Tian and Oh, 2019; Yang et al., 2019), the crucial difference of our framework lies in the sparsity of dependency between these highly abstract operators, which allows the respective learning of each module possible and thus eliminates the intra-operator interference.

4.2 Programme Generation

The programmer maps the natural language dialogue context to an executable programme in a reverse polish notation style. The code tokens (the vocabulary of the programmer) consist of two parts, namely the operator defined in Table 8 and the position index of the dialogue context. Following the design of Gupta et al. (2019) and Chen et al. (2020), the core architecture for programme generation is a MiniLM (Wang et al., 2020b) reader and a 1-layer GRU. At the t-th timestep, assume the embedding of past generated code tokens are $[\mathbf{h}_1^c, \mathbf{h}_2^c, \cdots, \mathbf{h}_{t-1}^c]$ and the dialogue context representation encoded by BERT is $\mathbf{H}^x = [\mathbf{h}_1^x, \mathbf{h}_2^x, \cdots, \mathbf{h}_l^x]$, where l is the length of the context. We first calculate $\mathbf{h}_t$, the hidden state of GRU at the current step:

$$\mathbf{h}_t = \mathrm{GRU}(\mathbf{h}_{t-1}, \mathbf{h}_{t-1}^c). \tag{1}$$

Then we apply the attention mechanism to compute the context vector $\mathbf{s}^x$:

$$\begin{aligned} \mathbf{s}^x &= \sum_{i=1}^{l} \mathbf{w}_i \mathbf{h}_i^x, \\ \mathbf{w} &= \mathrm{Softmax}(\mathbf{h}_t^{\mathrm{T}} \mathbf{H}^x), \end{aligned} \tag{2}$$

The history code vector $\mathbf{s}^c$ is computed in the same way. Afterwards, we concatenate the $\mathbf{s}^c$, $\mathbf{s}^x$ and GRU hidden state $\mathbf{h}_t$ together and computes:

$$\mathbf{s} = \mathbf{W}[\mathbf{s}^x; \mathbf{s}^c; \mathbf{h}_t], \tag{3}$$

where $[\cdot;\cdot]$ denotes the vector concatenation operation. Finally, the probability distribution of the next code token c_t is:

$$\Pr(c_t) = \mathrm{Softmax}(\mathbf{s}^{\mathrm{T}}[\mathbf{H}^x; \mathbf{E}^o]), \tag{4}$$

where $\mathbf{H}^x$ plays the role of position index embedding and $\mathbf{E}^o$ is a trainable operator embedding. We refer our readers to Chen et al. (2020) for more details about the programmer.

4.3 Programme Execution

As mentioned before, the generated programme is essentially a reverse polish expression (Burks et al., 1954). Therefore, we maintain a stack to assist the execution of the programme. To be more specific, given a generated code $\boldsymbol{c} = [c_1, c_2, \cdots, c_n]$ and an empty stack, we scan every code token in $\boldsymbol{c}$ one by one and take actions according to the current code token c_i:

- If c_i is a position index, push it into the stack;
- If c_i is `SPAN` operator, pop the top two items in the stack. Take the two items as the start index and end index to select a span. Push the span into the stack.
- If c_i is one of the content operators, pop the top one items in the stack and send it into the corresponding content module. Push the generated text into the stack.
- If c_i is one of the linguistics operators, pop the top two items in the stack, which should be two sentences. concatenate them together and send them into the corresponding linguistics module. Push the generated sentence into the stack.

Generally, the execution of programme bears similarity to that of push-down automata. The motivation behind this is to isolate different procedures in dialogue generation and use a stack to temporarily store the intermediate result, which is the only medium for message passing between modules.

At the end of the code execution, the item(s) left in the stack is popped out. To improve fluency, we attempt to polish the stack output with another neural network, but it seems that directly concatenating the outputted sentences together is enough.

5 Learning Details

5.1 Weak Supervision

Training the programmer and the modules jointly with the response as the only supervision signal is challenging (Gupta et al., 2020). More importantly, without the supervision of intermediate output, we have no idea whether the modules differentiate into intended functions. Annotating training data with human labor for every module is costly and we instead use heuristically obtained pseudo labels for substitution.

Algorithm 1 A high-level algorithm for producing pseudo labels.

```
Input: A pair of $(\boldsymbol{x}, \boldsymbol{y})$, a similarity function $\mathrm{sim}(\cdot,\cdot)$, a syntactic relation classifier $\mathrm{dis}(\cdot,\cdot)$, threshold $\psi_1$, $\psi_2$,
Initialize an empty code sequence $\boldsymbol{c}$ and pseudo-labeled datasets $\mathcal{D}^{op}$ for all modules
Use parsing tools to parse $\boldsymbol{y}$ into a tree $\mathcal{T}$.
for Segment $s$ among the in-order traverse sequence do
    Search a span $s'$ from $\boldsymbol{x}$ that is most similar to $s$ and
    Locate the start and end position of the span and append them into $\boldsymbol{c}$.
    Append SPAN into $\boldsymbol{c}$
    if $\mathrm{sim}(s, s') > \psi_2$ then
        Append COPY into $\boldsymbol{c}$.
        Add the pair $(s, s')$ into $\mathcal{D}^{copy}$
    else if $\mathrm{sim}(s, s') < \psi_1$ then
        Append INFER into $\boldsymbol{c}$.
        Add the pair $(s, s')$ into $\mathcal{D}^{infer}$
    else
        Append PARAPHRASE into $\boldsymbol{c}$.
        Add the pair $(s, s')$ into $\mathcal{D}^{paraphrase}$
    end if
    if One child of $s$ has been visited (denoted as $s^{chi}$) and the parent of $s$ has not been visited yet then
        Append OP=dis$(s, s^{chi})$ into $\boldsymbol{c}$.
        Add the pair $(s, s^{chi})$ into $\mathcal{D}^{op}$
    else if All children of $s$ have been visited and the parent of $s$ has been visited too (denoted as $s^{par}$) then
        Append OP=dis$(s, s^{par})$ into $\boldsymbol{c}$.
        Add the pair $(s, s^{par})$ into $\mathcal{D}^{op}$
    end if
end for
Return The pseudo-code label $\boldsymbol{c}$ and $\mathcal{D}^{op}$
```

Algorithm 1 is a high-level illustration of how we make pseudo labels. More details could be found in Appendix C.

5.2 Reinforcement Learning.

When trained respectively, the programmer and the modules may not adapt to each other well when directly assembled together. Therefore, we propose to further optimize the programmer with policy gradient (Sutton et al., 1999),

$$J(\theta) = \mathbb{E}_{\boldsymbol{c} \sim p_\theta(\boldsymbol{c}|\boldsymbol{x})}[r(\boldsymbol{c})] \quad (5)$$

and design the reward $r(\boldsymbol{c})$ as the similarity between the generated hypothesis and the ground truth response. Combining the ratio-likelihood trick, we have

$$\nabla_\theta J(\theta) = \mathbb{E}_{\boldsymbol{c} \sim p_\theta(\boldsymbol{c}|\boldsymbol{x})}[\nabla_\theta \log p_\theta(\boldsymbol{c}|\boldsymbol{x}) r(\boldsymbol{c})], \quad (6)$$

where $g(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{c})$ represents the execution of the code to generate response.

Dataset	Metric	BART	R2C2	Prefix	MS	Ours	SOTA
Cornell Movies	Rouge-1	10.93	9.97	7.11	11.37	**11.56**	12.11 (He et al., 2021)
DailyDialog	BLEU-1	43.58	40.12	34.68	43.04	**45.90**	42.84 (Chen et al., 2022)
CMU_DoG	Rouge-1	13.69	12.16	13.75	13.71	**15.50**	15.37 (Martins et al., 2022)
LIGHT	unigram-F1	14.52	11.91	13.80	14.57	**15.90**	15.88 (Shuster et al., 2022b)
EmpatheticDialog	Rouge-1	16.21	14.76	16.17	14.88	**18.62**	16.13 (Li et al., 2022a)
Wizard of Wikipedia	unigram-F1	33.24	30.94	30.02	29.14	**36.29**	36.00 (Li et al., 2022b)
ConvAI2	unigram-F1	19.24	17.09	15.81	17.51	19.79	20.50 (Shuster et al., 2022a)
Mutual	Rouge-L	17.22	18.03	17.77	15.33	17.26	22.70 (Liu et al., 2022)
CommonsenseDialog	Rouge-1	14.95	13.79	13.67	13.03	**15.15**	14.97 (Zhou et al., 2021)

Table 1: Experiment results in all-task MTL setting. Numbers in bold means that the improvement over baselines is statistically significant(t-test, $p<0.05$).

Dataset	Metric	BART	R2C2	MS	Ours
CM	Rouge-1	10.16	9.73	10.83	**11.08**
DailyDialog	BLEU-1	33.48	33.16	32.59	**35.06**
CMU_DoG	Rouge-1	12.78	11.49	11.51	**13.05**
LIGHT	unigram-F1	9.79	10.47	10.98	**13.30**
ED	Rouge-1	15.15	11.13	13.69	**15.58**
ConvAI2	unigram-F1	14.22	14.66	14.75	**15.36**
WoW	unigram-F1	19.56	17.36	20.10	**23.80**
Mutual	Rouge-L	12.82	10.43	13.33	**13.63**
CD	Rouge-1	13.13	13.30	12.51	**14.50**

Table 2: Experiment results in the leave-one-out setting. CM = Cornell Movies, ED = EmpatheticDialog, WoW = Wizard of Wikipedia and CD = CommonsenseDialog. Numbers in bold mean that the improvement over baselines is statistically significant(t-test, $p<0.05$).

In addition, to facilitate end-to-end training, we apply Gumbel-Softmax trick (Jang et al., 2017) to overcome the differentiable obstacle owing to the discrete nature of natural language when optimizing the modules. Formally, instead of selecting one token from module-predicted vocabulary distribution $\mathcal{M}(y_i^m|x^m, y_{<i}^m)$, the content modules and linguistics modules sample a “soft word”:

$$y_i^* = \text{Gumbel}(\mathcal{M}(y_i^m|x^m, y_{<i}^m), \tau), \quad (7)$$

where τ is the temperature of sampling.

6 Experiment

6.1 Experimental Setup

Setting. To comprehensively evaluate the multi-task learning ability and the generalization ability, suppose we have N datasets, we evaluate our proposed framework in three settings: (1) All Task MTL. In this setting, we train our model on the mixed union of N datasets and evaluate it on each individual dataset. (2) Leave-one-out. In this setting, we train our model on $N-1$ datasets and test on the left one dataset to evaluate a model's zero-shot generalization ability. (3) Low-resource. To further evaluate the generalization capability of our method, after training on other $N-1$ datasets in the leave-one-out setting, we fine-tune the model on the left dataset with only 10% data available, and test the model on the left one dataset.

Datasets. We use $N = 9$ datasets to evaluate our framework: Cornell Movies (Danescu-Niculescu-Mizil and Lee, 2011), DailyDialog (Li et al., 2017), CMU_DoG (Zhou et al., 2018b), LIGHT (Urbanek et al., 2019), EmpatheticDialog (Rashkin et al., 2019), ConvAI2 (Dinan et al., 2020), Wizard of Wikipedia (Dinan et al., 2019), Mutual (Cui et al., 2020) and CommonsenseDialog (Zhou et al., 2021). Each dataset embodies one or more specific skills. More details about the datasets could be found in Appendix B.

Baselines. We use **BART** (Lewis et al., 2020) as one of our baselines, which is a standard sequence-to-sequence transformer pre-trained on the same corpus as Liu et al. (2019); We also compare against **R2C2**, a BlenderBot-like open-domain dialogue model trained in a multi-task way by Shuster et al. (2022a) and hold the current state-of-the-art on many datasets (Zhang et al., 2022). For parameter-efficient technique in multi-task learning, we compare our method with **prefix-tuning** (Li and Liang, 2021). We also draw a comparison with the recent proposed **Modular Skill (MS)** (Ponti et al., 2022), a modular network that allows each task to choose its skill toolkit and optimize the global skill inventory together with the choice of each task jointly. For a fair comparison, we use BART-large (406M) and R2C2-base (400M) in our experiments. The parameter scale of Prefix-tuning (415M) and MS (448M) are both comparable with ours.

6.2 Main Result

All Task MTL. The experiment results are shown in Table 1. We could observe that (1) our proposed approach outperforms BART and R2C2 on most datasets. The advantage of our modular framework over prefix-tuning is also obvious, possibly because prefix-tuning hinders positive transfer among corpora. To have a more comprehensive understanding of our approach, we investigate the schedule frequency of modules on different datasets and it reveals that our modular design captures some distinctive patterns in different corpora. More information could be found in Appendix F.

(2) Meanwhile, we also provide the performance of the current state-of-the-art for each individual dataset[2]. We could observe that when trained in a multi-task way, our framework is superior or comparable to the SOTA without a sophisticated design of model architecture and learning algorithm for each individual dataset, which further verifies the capacity of our model to transfer knowledge from other corpus and manipulate multiple skills.

Leave-one-out. The results are shown in Table 2. There is a gap in performance between the baseline and ours, especially on Wizard of Wikipedia. It can be understood that Wizard of Wikipedia is less similar to other datasets since it contains some formal sentences from Wikipedia. Thus, zero-shot generalization on the dataset is more difficult. We can conclude that our model generalizes better than BART and R2C2, possibly because the modular framework could recursively compose the computations by modules to cope with new situations with existing knowledge. Besides, the comparison with MS further verdict the necessity of intermediate supervision for each module.

Low-resource The results are shown in Table 3. The proposed method attains a better performance than BART. Notably, our modular generation framework surpasses the fully supervised approach on DailyDialog, EmpatheticDialog, LIGHT and CommonsenseDialog, validating the potential of the compositional modular paradigm as a general method in the low-resource setting.

7 Further Analysis

7.1 Single Transfer Relation

To explore whether our framework enhances transfer in a multi-task learning scenario, we further draw a comparison in a single-task scenario where we train and test our model and all the baselines on each individual dataset. The experiment results are shown on Table 4. When comparing with Table 1, we could see that our approach achieves a positive transfer on most datasets while negative transfer is more common for baseline methods. It demonstrates that our modular design effectively alleviates the intra-operator transfer.

7.2 Pair-wise Transfer Relations

To have a closer look at the transfer relation among the datasets, we evaluate the transfer among datasets in a pair-wise multi-task learning setup. We use CommonsenseDialog, LIGHT, CMU_DoG and EmpatheticDialogie since they are diverse enough to be representative. The experiment results are shown in Table 5. Our approach attains positive transfer or at least avoids drop on most dataset pairs, while for BART the opposite is true. Besides that, an interesting trend manifests in individual relationships. For example, CMU_DoG and EmpatheticDialog seem to promote each other whilst LIGHT and CommonsenseDialog tend to hurt each other.

7.3 Ablation Study

An ablation study is conducted to explore how different mechanisms and components contribute to the performance. We compare our approach with the following variants: (1) *-span*: The `SPAN` operator is removed and we always select the entire dialogue context as a "span". (2) *-linguistic*: The linguistics operator is replaced with a direct concatenation of two input segments. (3) *-warm*: The warm-up procedure is removed. (4) *-reward*: The reinforcement learning of programmer is removed. The results are shown in Table 6. The result reveals that warm-up is indispensable to the proposed method, and the conclusion is in coincidence with Gupta et al. (2020, 2021). The span operator and the linguistic operators are also helpful to the performance. The decline in appropriateness of *-span* and *-linguistic* validates the necessity of them.

[2] Some numbers are directly cited from the papers

Dataset	Metrics	BART	R2C2	Prefix	MS	Ours
Cornell Movie	Rouge-1	10.70	9.19	8.03	9.31	**12.21**
DailyDialog	BLEU-1	41.86	42.13	39.57	42.82	**45.54**
CMU_DoG	Rouge-1	14.28	13.96	12.41	14.21	**15.15**
LIGHT	unigram-F1	14.34	12.61	12.34	14.00	**15.98**
EmpathicDialogue	Rouge-1	15.94	16.12	16.01	15.70	**17.79**
ConvAI2	unigram-F1	18.29	18.08	18.18	17.42	18.50
Wizard of Wikipedia	unigram-F1	31.85	33.41	29.83	30.00	33.52
Mutual	Rouge-L	18.28	10.43	13.33	13.64	18.66
CommonsenseDialog	Rouge-1	13.97	13.92	13.42	12.97	**14.61**

Table 3: Experiment results for fine-tuning in low-resource (10% data) setting. Numbers in bold mean that the improvement over the best supervised method is statistically significant. (t-test, $p<0.05$)

Dataset	Metric	BART	R2C2	MS	Ours
Cornell Movie	Rouge-1	10.09	8.30	11.17	10.38
DailyDialog	BLEU-1	43.00	42.25	43.73	43.72
CMU_DoG	Rouge-1	15.04	12.92	13.78	15.16
LIGHT	unigram-F1	15.46	14.71	14.35	15.14
EmpatheticDialog	Rouge-1	16.43	17.43	15.11	17.35
Wizard of Wikipedia	unigram-F1	35.30	34.85	34.37	36.70
ConvAI2	unigram-F1	20.72	19.89	19.11	20.16
Mutual	Rouge-L	20.60	22.26	17.51	20.02
CommonsenseDialog	Rouge-1	14.81	15.04	14.42	14.97

Table 4: Experiment results on each individual dataset.

	LIGHT	CMU_DoG	ED	CD
LIGHT	15.46	14.22	15.98	14.04
CMU_DoG	14.29	15.04	18.06	14.34
ED	15.00	15.27	16.43	14.41
CD	15.09	14.60	16.64	14.81

	LIGHT	CMU_DoG	ED	CD
LIGHT	15.14	14.93	17.71	14.64
CMU_DoG	15.63	15.17	18.03	15.19
ED	15.15	15.30	17.35	15.61
CD	15.05	15.37	18.30	14.97

Table 5: Pair-wise transfer relation of BART (top) and our method (bottom) on four datasets. The entry at (row i, column j) indicates performance on dataset j using a model trained on datasets i and j. ED = EmpathicDialog and CD = CommonsenseDialog.

	CMU_DoG	LIGHT	ED	CD
ours	15.50	15.90	18.62	15.15
-span	14.81	14.43	18.17	14.75
-linguistic	15.44	13.79	17.62	15.02
-warm	12.58	13.03	16.71	12.25
-reward	15.10	15.62	17.98	14.35

Table 6: Ablation results on four datasets. ED = EmpatheticDialog, CD = CommonsenseDialog

	1(%)	2(%)	3(%)	Avg
BART	21	57	22	2.01
R2C2	12	59	29	2.17
Prefix	39	31	30	1.91
MS	17	52	31	2.14
Ours	9	47	44	2.35

Table 7: Human evaluation results in all-task MTL setting.

7.4 Qualitative Evaluation

Automatic metrics are not perfect for evaluating an open-domain task (Dinan et al., 2019) and human evaluation is necessary. Concretely, in the all-task MTL setting, we randomly sample 300 responses from each dataset generated by ours and baseline methods and recruit well-educated native speakers to rate them. Each annotator is required to give a score ranging from 1 to 3. 1 means the response is correct in grammar and fluent; 2 means the response is coherent to the context and satisfies the requirements of 1. 3 means the response exhibits versatile skills if necessary including showing empathy, grounding on knowledge, commonsense inference, etc. Besides, the response should also meet the requirements of 2. Agreement of the annotators is measured via Fleiss' kappa (Fleiss, 1971). As is shown in Table 7, the responses generated by our approach enjoy a higher quality, demonstrating the superiority of the modular generation framework. The evaluation results are also consistent with automatic evaluation.

A case study could be found in Appendix F.

8 Conclusions

In this work, we utilize the ideology of modular networks to address the transfer-interference problem in multi-task learning. We implement a model architecture that allows the composition of different modules to fulfill complicated functions and eliminate interference among modules. We apply our method to dialogue generation and conduct extensive experiments to verdict its efficacy. We hope our work would inspire relevant research in the community.

Ethic Considerations

The use of our approach could result in improved dialogue systems that enhance the quality of life for many individuals, especially in light of the widespread use of AI in everyday life. For instance, a more effective chatbot integrated with electronic gadgets will boost both productivity and user experience. On the other hand, the implementation of conversation systems could result in employment losses in some domains such as call centers.

Limitations

This work focuses on mitigating the negative transfer and catastrophic forgetting issue in multi-task dialogue generation. All technologies built upon the large-scale PLM more or less inherit their potential harms (Bender et al., 2021). Besides, we acknowledge some specific limitations within our methods:

1. The construction of pseudo labels requires dependency parsing with spaCy, which is time-consuming. But we only construct pseudo labels offline in the training processing and it causes no latency at inference.

2. We instantiate our modular framework using MiniLM (Wang et al., 2020b) as the backbone of the reader within the programmer, and T5 (Raffel et al., 2019) as the backbone for the content operators and linguistic operators. We did not try other instantiations although the modular framework does not depend on the specific initialization choice of modules. Theoretically, any generative PLM could be the backbone of these linguistic and content modules.

3. We aim at decomposing the response generation into relatively independent and composable operators. Currently, the division of dialogue skills and module functions is in a heuristic way inspired by linguistics. Thus it remains a future research question about how to design modular architecture in a more data-driven way.

Acknowledgement

We thank all the reviewers and chairs for their suggestions and recommendation. This work was supported by National Natural Science Foundation of China (NSFC Grant No. 62122089), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Inter-disciplinary Platform for the ”Double-First Class” Initiative, Renmin University of China. We wish to acknowledge the support provided by Public Policy and Decision-making Research Lab, Renmin University of China and the Public Computing Cloud, Renmin University of China.

References

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.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

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*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. 2021. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2111.10952*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Arthur W Burks, Don W Warren, and Jesse B Wright. 1954. An analysis of a logical machine using parenthesis-free notation. *Mathematical tables and other aids to computation*, 8(46):53–57.

Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, Biao Cheng, and Nan Duan. 2022. DialogVED: A pre-trained latent variable encoder-decoder model for dialog response generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4852–4864, Dublin, Ireland. Association for Computational Linguistics.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2020. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. MuTual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416, Online. Association for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. *arXiv preprint arXiv:1106.3077*.

Stanislas Dehaene, Hakwan Lau, and Sid Kouider. 2021. What is consciousness, and could machines have it? *Robotics, AI, and Humanity*, pages 43–56.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition*, pages 187–208. Springer.

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.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1243–1252. JMLR.org.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural module networks for reasoning over text. *arXiv preprint arXiv:1912.04971*.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *International Conference on Learning Representations*.

Nitish Gupta, Sameer Singh, Matt Gardner, and Dan Roth. 2021. Paired examples as indirect supervision in latent decision models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5774–5785, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1121–1133, Online. Association for Computational Linguistics.

Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2018. Explainable neural computation via stack neural module networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 53–69.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 804–813.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478.

Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2018. Visual coreference resolution in visual dialog using neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–169.

Hung Le, Nancy Chen, and Steven Hoi. 2022. Vgnmn: Video-grounded neural module networks for video-grounded dialogue systems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3377–3393.

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.

Qintong Li, Piji Li, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. 2022a. Knowledge bridging for empathetic dialogue generation.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

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

Yu Li, Baolin Peng, Yelong Shen, Yi Mao, Lars Liden, Zhou Yu, and Jianfeng Gao. 2022b. Knowledge-grounded dialogue generation with a unified knowledge representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 206–218, Seattle, United States. Association for Computational Linguistics.

Ruibo Liu, Guoqing Zheng, Shashank Gupta, Radhika Gaonkar, Chongyang Gao, Soroush Vosoughi, Milad Shokouhi, and Ahmed Hassan Awadallah. 2022. Knowledge infused decoding. In *International Conference on Learning Representations*.

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

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Pedro Henrique Martins, Zita Marinho, and Andre Martins. 2022. former: Infinite memory transformer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5468–5485, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487.

Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. 2022. Is a modular architecture enough? *arXiv preprint arXiv:2206.02713*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. 2019. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

A. See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *NAACL*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2453–2470.

Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022a. Language Models that Seek for Knowledge: Modular Search & Generation for Dialogue and Prompt Completion. *CoRR*, abs/2203.13224. Version 1.

Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2022b. Am I me or you? state-of-the-art dialogue models cannot maintain an identity. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2367–2387, Seattle, United States. Association for Computational Linguistics.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*.

Junjiao Tian and Jean Oh. 2019. Image captioning with compositional neural module networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3576–3584. International Joint Conferences on Artificial Intelligence Organization.

Junjiao Tian and Jean Oh. 2020. Image captioning with compositional neural module networks. *arXiv preprint arXiv:2007.05608*.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 673–683, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Richard S. Wallace. 2009. The anatomy of a.l.i.c.e.

Qiansheng Wang, Yuxin Liu, Chengguo Lv, Zhen Wang, and Guohong Fu. 2020a. Cue-word driven neural response generation with a shrinking vocabulary. *ArXiv*, abs/2010.04927.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Yansen Wang, Chen-Yu Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *ACL*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*.

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*.

Joseph Weizenbaum. 1966. Eliza: A computer program for the study of natural language communication between man and machine. volume 9, pages 36–45. ACM.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, M. Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*.

Chen Xing, Wei Yu Wu, Yu Wu, Ming Zhou, Yalou Huang, and Wei-Ying Ma. 2018. Hierarchical recurrent attention network for response generation. In *AAAI*.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197.

Xu Yang, Hanwang Zhang, and Jianfei Cai. 2019. Learning to collocate neural modules for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4250–4260.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31.

Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. ReCoSa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3721–3730, Florence, Italy. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

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

Yu Zhang and Qiang Yang. 2021. A survey on multitask learning. *IEEE Transactions on Knowledge and Data Engineering*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018a. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4623–4629, Stockholm, Sweden. International Joint Conferences on Artificial Intelligence Organization.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018b. A Dataset for Document Grounded Conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021. Commonsense-focused dialogues for response generation: An empirical study. *arXiv preprint arXiv:2109.06427*.

A Details about Operators

The operators used in our framework are listed in Table 8.

B More Details about Datasets

Cornell Movies (Danescu-Niculescu-Mizil and Lee, 2011) contains large-scale fictional conversations extracted from raw movie script. thus covering abundant topics and emotional change. The dataset is used for training and evaluating a chatbot to quickly capture the emotional change in dialogue and respond accordingly.

LIGHT (Urbanek et al., 2019) is about situated interaction between characters in a text adventure game. The dialogue context includes not only the historical utterance but also the persona and action of the speakers together with the background setting. The skill in this dataset is grounding discussion on the dynamic environment.

EmpatheticDialogue (Rashkin et al., 2019) is a crowd-sourced dataset in which a speaker describes his or her situation and a listener responds with empathy. The dataset provides the emotion labels for interlocutors at each turn, but we do not include that in the dialogue context of our experiment.

ConvAI2 (Dinan et al., 2020) is the dataset used for NeurIPS 2018 competition and is adapted from PERSONACHAT (Zhang et al., 2018). In conversation, the interlocutors are required to exhibit a given persona and try to know the persona of the partner at the same time. The dataset mainly focuses on the skill of getting to know each other and engaging in friendly greeting conversation.

Wizard of Wikipedia (Dinan et al., 2019) is a knowledge-grounded dataset in which an interlocutor plays the apprentice and asks questions while the other interlocutor plays the wizard and gives informative responses. The wizard has access to background knowledge from Wikipedia. We only include the golden knowledge in the dialogue context. The dataset is for training and evaluating the skill of grounding conversation on knowledge.

CMU_DoG (Zhou et al., 2018b) is also a dataset for knowledge-grounded dialogue. In each conversation, two interlocutors discuss a given movie. The basic information of the movie including rating, release year, review and main plots are provided as background knowledge. Similarly, we use only the golden knowledge. The dataset focuses on grounding knowledge.

Mutual (Cui et al., 2020) is collected from Chinese students' English listening comprehension exams. The model needs to generate a logically correct continuation of the conversation based on historical utterances. The dataset facilitates reasoning ability on social etiquette and relationships.

CommonsenseDialog (Zhou et al., 2018a) consists of two parts. The first part is extracted from the existing dialogue dataset using ConceptNet while the second part is crowd-sourced asking the crowd workers to exhibit social commonsense in an interacting environment. We only use the crowd-sourced part to avoid overlap with other datasets used in our experiments. The dataset requires the skill of performing latent or explicit commonsense inference in communication.

DailyDialog (Li et al., 2017) is a dataset intended to reflect conversations occurring in daily life, covering a wide range of domains and topics. The dataset is also annotated with the topic, emotion and utterance act, but we only use the history utterance as the dialogue context.

Since the test set of ConvAI2 and Mutual is not publicly released, we conduct validation on a separate subset (10%) of the training set and test on the original validation set.

The statistics of our datasets are listed in Table 9.

C More Details about Weak Supervision

Since algorithm 1 is only a high-level description, we provide more details here about how to produce our pseudo training data. In implementation, we use spaCy [3] as our parsing tool. It outputs a parsing tree and every token in the sentence is a node. We process the token-level parsing tree into a segment-level parsing tree by merging the nodes into verb phrases. Specifically, we merge all the nodes and within the subtree of a verb node unless it is another verb node or its nearest verb ancestor is another verb node. The edge between the verb nodes is kept unchanged. As a result, we parse the golden response into a segment tree $\mathcal{T}$. To traverse all the segments in the tree, we use a pseudo in-order traverse because in the parsing tree a node may have more than two children and a traditional in-order traverse does not work here. Precisely, we for every node to visit, we first visit its first child, then the node itself, and finally all the other children. In algorithm 1, the similarity function

[3] https://spacy.io/

Operator	Input	Output	Description
SPAN	v0: start index; v1: end index	text	Select a span from dialogue context
COPY	v: text	text	Copy the input text
PARAPHRASE	v: text	text	Paraphrase the input text
INFER	v: text	text	Take the input as premise and infer a hypothesis
NOUN_MODIFY	v0: text v1: text	text	Connect one clause to another to modify a noun
VERB_MODIFY	v0: text v1: text	text	Connect one clause to another to modify a verb
COMPOUND	v0: text v1: text	text	Connect the two sentences with a conjunct

Table 8: The operators used in programme generation.

	Training	Validation	Test	Resp.Length
Cornell Movies (Danescu-Niculescu-Mizil and Lee, 2011)	110,161	13,914	13,701	10.84
DailyDialog (Li et al., 2017)	76,005	8,069	7,740	11.61
CMU_DoG (Zhou et al., 2018b)	66,333	3,270	10,502	18.53
LIGHT (Urbanek et al., 2019)	93,784	5,623	11,268	12.98
EmpatheticDialogue (Rashkin et al., 2019)	64,635	5,738	5,259	11.72
Wizard of Wikipedia (Dinan et al., 2019)	74,092	3,939	3,865	13.02
ConvAI2 (Dinan et al., 2020)	131,438	7,801	-	11.48
Mutual (Cui et al., 2020)	7,088	886	-	13.02
CommonsenseDialog (Zhou et al., 2021)	25,552	3,268	1,158	8.86

Table 9: Statistics of the datasets used in our experiments. Resp.Length is the abbreviation for the length of response (number of words).

$\text{sim}(\cdot,\cdot)$ is unigram F1[4] (Dinan et al., 2019). We set ϕ_1 to be 0.35 and ϕ_2 to be 0.75. The syntactic relation classifier $\text{cls}(\cdot,\cdot)$ is based on the dependency relation r between the two verb nodes in two segments:

$$\text{cls}(s_1, s_2) = \begin{cases} \texttt{COMPOUND}, & \text{r} = \text{conj}, \\ \texttt{VERB\_MODIFY}, & \text{r} = \text{advcl} \\ \texttt{NOUN\_MODIFY}, & \text{r} \in \{\text{relcl, acl}\} \end{cases} \tag{8}$$

D More Implement Details

All the content modules and linguistics modules are sequence-to-sequence transformers initialized with T5-small (Raffel et al., 2019). The reader within the programmer is a bidirectional 6-layer transformer with an embedding size of 512. Its parameters are initialized from MiniLM (Wang et al., 2020b). The GRU in the programmer is 1-layer with the dimension of hidden state 512. All the models are learned with Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We sweep learning rate from $[5e-6, 1e-5, 2e-5, 4e-5, 6e-6, 8e-5]$ and sweep batch size from $[16, 32, 64, 128, 256]$.

	CMU_DoG	LIGHT	ED	CD
BART (406M)	626.06	617.95	619.94	623.17
Ours (327M)	308.39	355.43	310.22	406.97

Table 10: Average inference time (ms) of BART and our method on four datasets. ED = EmpatheticDialog, CD = CommonsenseDialog.

We set the weight decay as $1e-2$ and sweep the warmup steps from $[1000, 2000, 4000]$. The gradient clip is set to 2.0 to avoid the explosion of the gradient. The reward for reinforcement is implemented as the unigram-F1. We keep the temperature τ to be 1.0 through our experiment. A cosine learning schedule is applied to adjust the learning rate during training. An early stop on the validation set is adopted. We truncate the input dialogue context to a maximum length of 480. We conduct experiments on two RTX 3090. We use greedy search for decoding and report the performance averaged in three repetitive experiments.

E Inference Speed

We further compare the decoding speed at inference time with BART to see whether the modular

[4] https://github.com/facebookresearch/ParlAI/blob/master/parlai/core/metrics.py

Context	A: Excuse me, could you help me pick out a lotion? B: Sure, what is the problem? A: I got poison oak while hiking, and I need something to help me with the itching. B: I can suggest a product called techne that comes in a lotion or cream. Which do you prefer? Hikers tell me that the cream is best because it stays on longer.
BART	I prefer the cream.
R2C2	I prefer the cream because it stays on longer.
Human	Is there anything else I can do to help with the itching?
Code	[36,42, SPAN, COPY, 44, 48, SPAN, PARAPHRASE, VERB_MODIFY, 18, 24, SPAN, INFER, NOUN_MODIFY]
Ours	I think it's better because the cream stays on longer than I do in my hiking shoes.

Table 11: A case study from DailyDialog dataset (Li et al., 2017).

generation framework suffers from high latency. We conduct experiments on CMU_DoG, LIGHT, EmpatheticDialog and CommonsenseDialog with an RTX 3090. The experiment results are shown in Table 10. From the table, we could observe that our model has a lower inference latency than BART. We gauge the reason that the modules in our framework are much smaller in scale. Meanwhile, those modules only attend to the partial text selected by SPAN rather than the entire dialogue context.

F Case Study

To have an intuitive understanding of how our modular framework takes effect, we show three cases in all-task MTL setting in Table 12, Table 11 and Table 13. Besides, we are also interested in whether the schedule frequency of each module varies in different datasets. We believe the difference of schedule frequency exhibits some intrinsic feature of the dataset. From Figure 2 to Figure 19, we conjecture that the schedule frequency gives out the linguistic style of a dataset. For example, in Wizard of Wikipedia (Dinan et al., 2019), a portion of sentences are directly copied from the knowledge; In ConvAI2 (Dinan et al., 2020), some sentences are paraphrased from the given persona of speakers.

Context	A: hi . I'm Billy. I work as an electrician in Texas. B: I very rarely speak to people Billy , because I'm so famous. A: Really? How? I do a lot of work on my mom and dads farm. Persona: I am a famous twitch streamer. I am working on a building a robot to be my body double. I helped design the game starcraft. I've an allergy to water.
BART	That's cool, I'm a famous twitch streamer .
R2C2	I'm a recluse.
Human	I am working on a robot so I can have company that doesn't annoy me .
Code	[6, 67, SPAN, COPY, 70, 77, SPAN, PARAPHRASE, NOUN_MODIFY]
Ours	I'm an electrician and work on the game starcraft to be my body double

Table 12: A case study from ConvAI2 dataset (Dinan et al., 2020).

Context	A: Did you like Angelina Jolie's performance? Shame about the wings.... Would you watch it again? B: Do you agree with the rating of rotten tomatoes given to that movie? A: yes. the movie was ok. Knowledge: Maleficent is a powerful fairy living in the moors, a magical forest realm bordering a human kingdom.
BART	I agree with that rating. I think it 's a good movie.
R2C2	Yes, I agree with you.
Human	Yes, I like Angelina performance. She is a great actress .
Code	38, 43, SPAN, INFER, 40, 44, SPAN, INFER, 0, 9, SPAN, PARAPHRASE
Ours	Yes, I liked her performance. She was very good. What did you think of her performance?

Table 13: A case study from CMU_DoG dataset. (Zhou et al., 2018b)

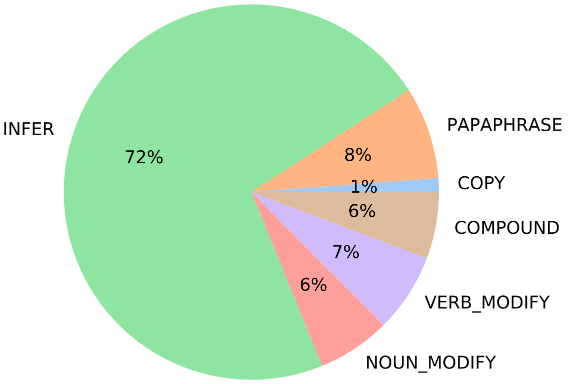

Figure 2: The distributions of content operators and linguistics operators on cmudog

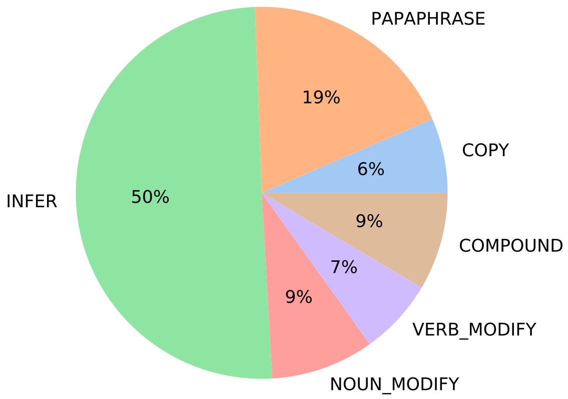

Figure 3: The distributions of content operators and linguistics operators on wizard

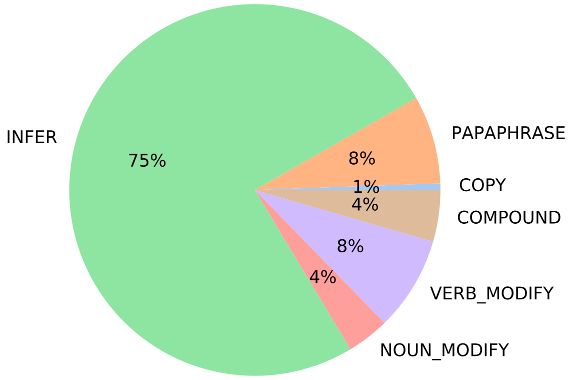

Figure 4: The distributions of content operators and linguistics operators on commonsense

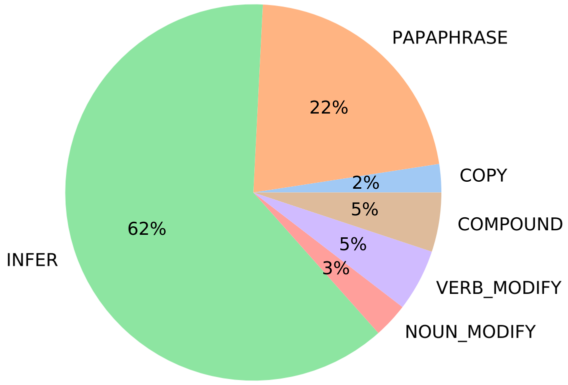

Figure 5: The distributions of content operators and linguistics operators on convai2

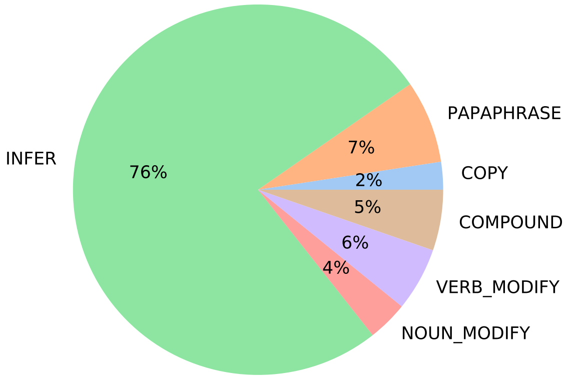

Figure 6: The distributions of content operators and linguistics operators on daily

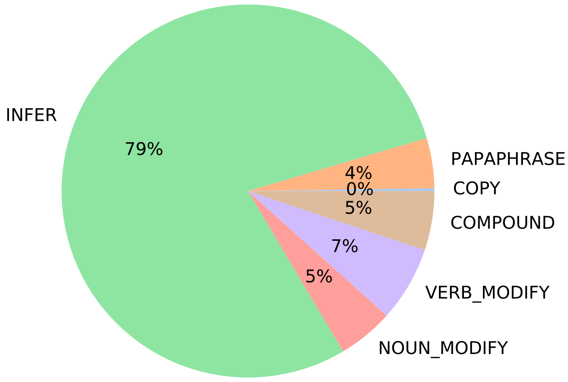

Figure 7: The distributions of content operators and linguistics operators on empathic

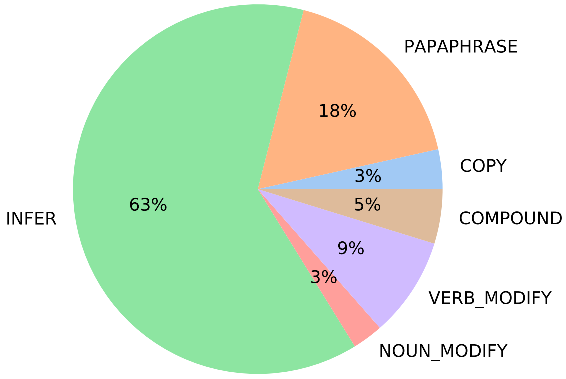

Figure 8: The distributions of content operators and linguistics operators on mutual

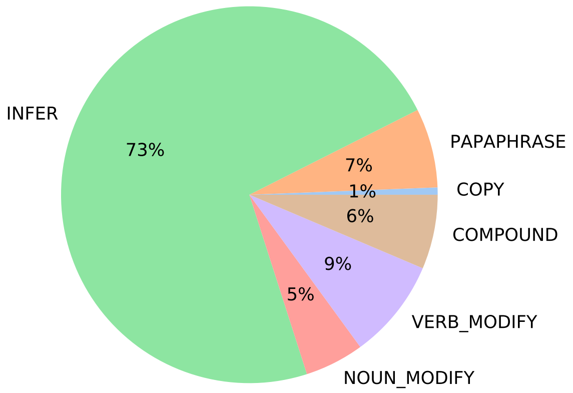

Figure 9: The distributions of content operators and linguistics operators on light

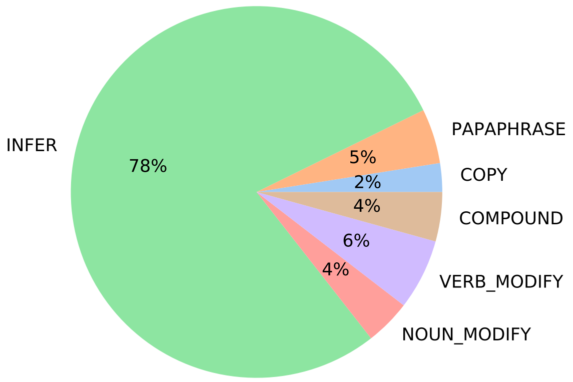

Figure 10: The distributions of content operators and linguistics operators on cornell

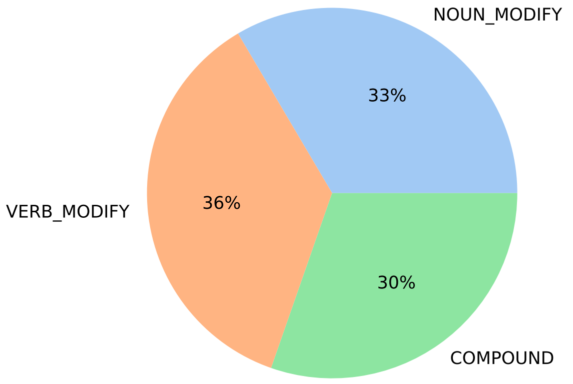

Figure 11: The distributions of linguistics operators on cmudog

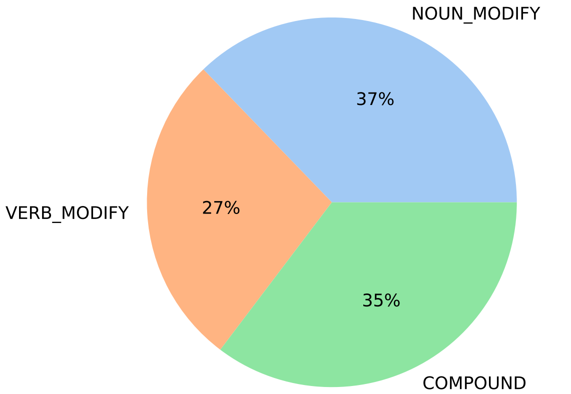

Figure 12: The distributions of linguistics operators on wizard

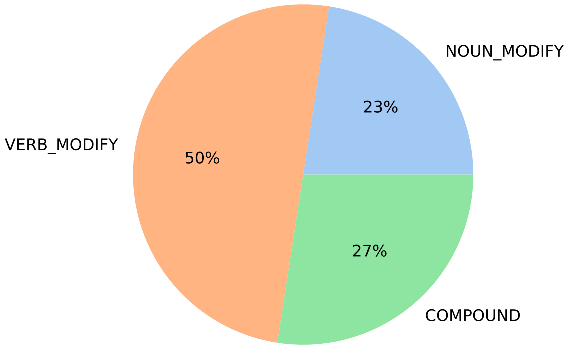

Figure 13: The distributions of linguistics operators on commonsense

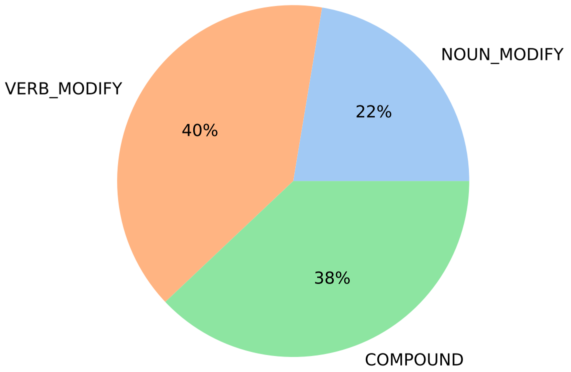

Figure 14: The distributions of linguistics operators on convai2

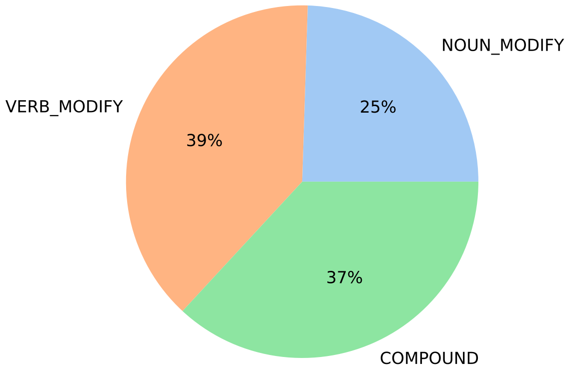

Figure 15: The distributions of linguistics operators on daily

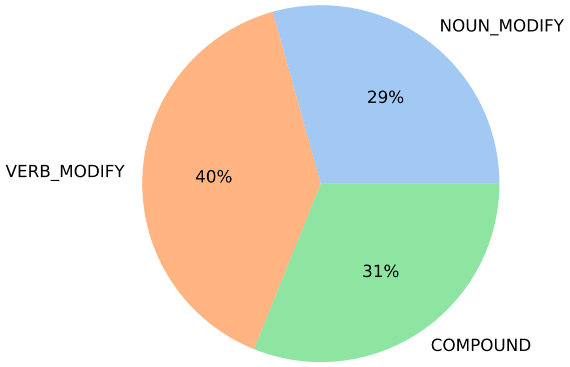

Figure 16: The distributions of linguistics operators on empathic

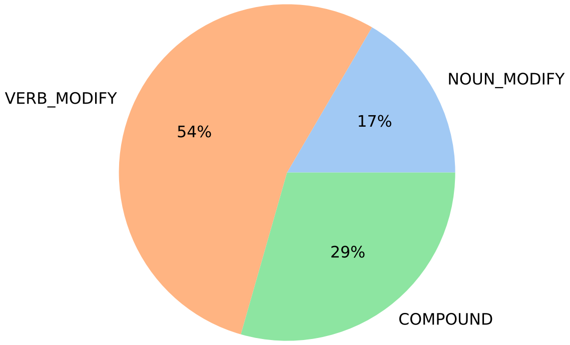

Figure 17: The distributions of linguistics operators on mutual

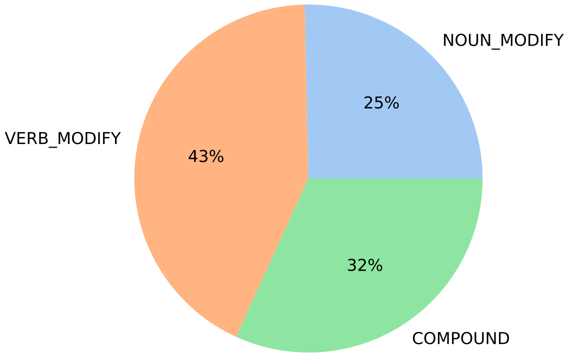

Figure 18: The distributions of linguistics operators on light

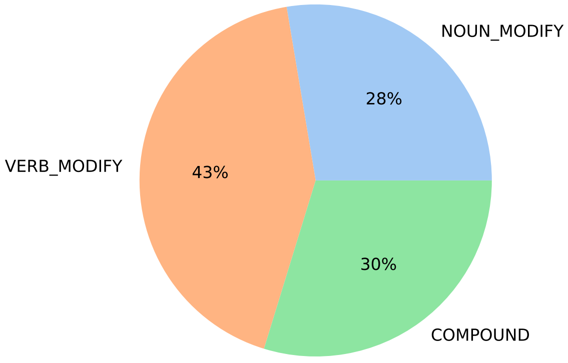

Figure 19: The distributions of linguistics operators on cornell

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☑ A2. Did you discuss any potential risks of your work?
Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
6,7

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
6, 7

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
6,7

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
6,7