A Universal Discriminator for Zero-Shot Generalization

Anonymous ACL submission

Abstract

Generative modeling has been the dominant approach for large-scale pretraining and zero-shot generalization. In this work, we challenge this convention by showing that discriminative approaches perform substantially better than generative ones on a large number of NLP tasks. Technically, we train a single discriminator to predict whether a text sample comes from the true data distribution, similar to GANs. Since many NLP tasks can be formulated as selecting from a few options, we use this discriminator to predict the concatenation of input and which option has the highest probability of coming from the true data distribution. This simple formulation achieves state-of-the-art zero-shot results on the T0 benchmark, outperforming T0 by 16.0%, 7.8%, and 11.5% respectively on different scales. In the finetuning setting, our approach also achieves new state-of-the-art results on a wide range of NLP tasks, with only 1/4 parameters of previous methods. Meanwhile, our approach requires minimal prompting efforts, which largely improves robustness and is essential for real-world applications. Furthermore, we also jointly train a generalized UD in combination with generative tasks, which maintains its advantage on discriminative tasks and simultaneously works on generative tasks.

1 Introduction

Generative modeling has been the dominant approach for large-scale pretraining and zero-shot generalization (Brown et al., 2020; Artetxe et al., 2021; Rae et al., 2021). Combined with prompts (Brown et al., 2020), most of the natural language processing (NLP) tasks can be formulated into the fill-in-the-blank format and perform generative language modeling. Based on the unified generative formulation, pretrained models such as GPT-3 (Brown et al., 2020), BERT (Devlin et al., 2019; Schick and Schütze, 2020), T5 (Raffel et al., 2019), can perform zero-shot inference on new tasks.

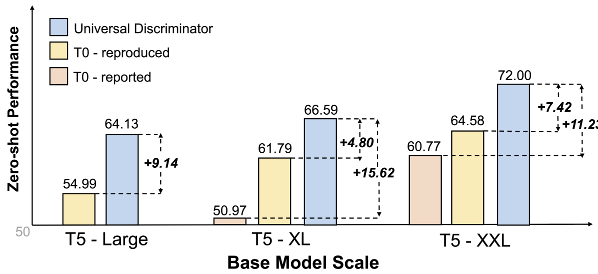

Figure 1: Average zero-shot performance over 11 zero-shot tasks for our Universal Discriminator and T0 (Sanh et al., 2021). Our universal discriminator significantly outperforms T0 across three different scales.

More recent work (Sanh et al., 2021) proposed to further pretrain a generative T5 (Raffel et al., 2019) with multitask prompted datasets and has substantially enhanced the performance of zero-shot generalization. In contrast, methods based on discriminative modeling (Devlin et al., 2019) have not been able to achieve state-of-the-art performance on zero-shot learning. The adoption of discriminative approaches for zero-shot learning has been limited in the literature.

In this work, we challenge the convention of zero-shot learning and propose to study and improve discriminative approaches. This is motivated by the fact that many NLP tasks can be framed as selecting from a few options; e.g., telling whether sentence A entails sentence B, or predicting which answer is correct for a given question. We call these tasks *discriminative tasks*. As we will discuss in later sections, a significant portion of NLP tasks is in fact discriminative tasks. We hypothesize that discriminative approaches perform better for discriminative tasks.

To verify the hypothesis, we propose the **universal discriminator (UD)**, which substantially improves zero-shot generalization over the previous generative state-of-the-art (SOTA) (Sanh et al., 2021), as Figure 1 shows. The main idea is to train a single discriminator to predict whether a text sample comes from the true data distribution of natural language, similar to GANs (Goodfellow et al., 2014). Given a set of training tasks with labeled data, we construct a dataset with positive and negative examples, where positive ones are in-distribution natural language samples and negative ones are out-of-distribution. There are two major types of discriminative tasks. The first type is tasks with multiple options, such as multi-choice question answering and news classification. We fill the options into the sentences and the ones with correct options are considered positive samples. The second type is tasks with yes/no options, which can be formulated as a binary discrimination problem itself. For example, natural language inference aims to predict whether a premise entails a hypothesis. In this case, we use a prompt to concatenate the premise A and the hypothesis B into a sentence "Premise: A. Hypothesis: B." If entailment holds, this sample is treated as positive in-distribution samples and otherwise negative out-of-distribution ones.

For the performance of zero-shot generalization, our approach achieves new state-of-the-art on the T0 benchmark, outperforming T0 by 16.0%, 7.8%, and 11.5% respectively on different scales. UD also achieves state-of-the-art performance on a wide range of supervised NLP tasks, using only 1/4 parameters of previous methods. Compared with the previous generative prompt-based methods, our universal discriminator requires minimal prompting, which is simple, robust, and applicable in real-world scenarios.

In addition, we also generalize UD to a larger scope of tasks, such that UD can perform discriminative and generative tasks at the same time. Specifically, we extend UD to the encoder-decoder architecture for training on generative tasks, and restrict the model's prediction on "yes"/"no" tokens for jointly training discriminative tasks. Results prove that generalized UD maintains UD's advantages on discriminative tasks and achieves comparable results on generative tasks (See § 3.4).

2 Related Work

2.1 Zero-Shot Generalization Using PLMs

Pretrained language models (PLM) can transfer knowledge from training data to downstream tasks. Prompting methods further narrow the gap between training data and downstream tasks. Schick and Schütze (2020) reformulate NLP tasks into cloze filling using prompts so that PLMs can conduct zero-shot inference by generating tokens given prompted inputs. Meng et al. (2022) use PLMs to generate class-conditioned texts with the guidance of prompts without seeing any task-specific data. Most recently, researchers have introduced natural language prompts to unify various kinds of tasks and propose a multi-task prompted training framework to achieve great zero-shot performance even faced with unseen downstream tasks (Wei et al. (2021); Sanh et al. (2021); Chung et al. (2022)). However, zero-shot learning has been dominated by generative approaches.

2.2 Prompt-based and Prompt-free Methods in NLP

Prompting is the method of reformatting NLP tasks using natural language templates to adapt to downstream tasks (Raffel et al., 2019; Schick and Schütze, 2020). To reduce the instability and labor costs brought by prompting, researchers have tried various approaches (Liu et al. (2021a); He et al. (2021a)) to learn continuous prompts.

Recently, prompt-free methods are also being explored. Mahabadi et al. (2022) adopts task-specific adapters to learn task descriptions implicitly for few-shot learning with PLMs. It has also been indicated that using null prompts without task-specific templates can achieve decent performance compared with manually-designed prompts on various tasks (Logan IV et al. (2021)).

Our work further shows that those widely used lengthy instructive prompts are not necessary for zero-shot learning. Actually, minimal prompting performs better with our discriminative formulation in the multi-task zero-shot learning setting.

2.3 Discriminative Models in NLP

PLMs trained with masked language modeling (MLM) (Devlin et al., 2019; Liu et al., 2019) can be finetuned in a discriminative manner for downstream tasks. ELECTRA (Clark et al., 2020) trains a discriminator to detect whether a token has been replaced. WKLM (Xiong et al., 2019) employs an entity-centric approach for pretraining and predicts whether an entity has been replaced. However, finetuning for these methods is usually based on one separate CLS head per task, which is not suitable for zero-shot generalization.

Recently, prompting has been combined with token-level discriminators based on ELECTRA for few-shot learning (Yao et al., 2022; Xia et al., 2022). While these are also discriminative approaches, there are a few key differences from our approach. The biggest difference between them and us is that: we unify all discriminative tasks into one single task with minimal prompting, showing extremely good zero-shot generalization. Moreover, these methods are specific to ELECTRA-like pretraining, while our approach accepts arbitrary pretrained encoders. In our experiments, we will also make a direct comparison with these approaches to demonstrate our effectiveness.

3 Approach

Previous works (Sanh et al., 2021; Wei et al., 2021) have shown that prompted multi-task training can greatly improve zero-shot performance on unseen tasks. One intuitive reason behind the validity of this improvement is that all the NLP tasks share a common ability that allows LMs to solve unseen tasks based on the data from other training tasks. To test this idea and even enhance zero-shot generalization, a direct way is explicitly defining what this "common ability" is. Here, we define this "common ability" by designing a new general task of "discriminating whether a text sample comes from the true data distribution of natural language".

We will first formulate the learning problem (§ 3.1), and then define the concept *discriminative tasks* (§ 3.2), followed by describing how we transform discriminative tasks into our shared formulation. In § 3.3 and § 3.4, we will study our UD, respectively on discriminative tasks and on a generalized scope of both discriminative and generative tasks.

3.1 Multi-Task Training for Zero-Shot Generalization

Now we describe the learning problem we aim to solve in this work. We adopt the same setting as in Sanh et al. (2021). The input to our problem is a set of training tasks with labeled data, and the goal is to train a model that generalizes to unseen test tasks. The training and test tasks are constrained to have distinct task types for the evaluation of cross-task-type generalization. A pre-trained model is jointly trained on the set of training tasks and directly evaluated on the set of test tasks in a zero-shot manner.

3.2 Discriminative Tasks

We use the term "discriminative tasks" to refer to tasks that can be framed as selecting from a few options.

More concretely, there are two types of discriminative tasks. The first type is tasks with multiple options, such as multi-choice question answering and news classification. The problem can be framed as selecting the right option from multiple ones, where the options are either customized for each sample (e.g., multi-choice question answering) or shared within the task (e.g., news classification). The second type is tasks with yes/no options, such as paraphrase identification and natural language inference. Given a sample of these tasks, a model is asked to predict a yes/no (or true/false) answer.

It is important to notice that discriminative tasks constitute a significantly large portion of modern NLP research tasks. For example, all of the test tasks of the T0 benchmark (Sanh et al., 2021), SuperGLUE (Wang et al., 2019a), GLUE (Wang et al., 2019b), and 85+% tasks in BBH benchmark (Suzgun et al., 2022) are discriminative tasks.

Also note that our definition of discriminative tasks has a larger scope compared to the conventional notion of "classification" which usually refers to tasks with a non-customized, fixed set of labels. In contrast, discriminative tasks might have sample-customized options, e.g., multi-choice question answering and coreference resolution.

3.3 A Universal Discriminator

Given a text sample x, let $P(\text{true}|x)$ be the probability that x is sampled from the true data distribution of natural language. We train a universal discriminator (UD), denoted as $D(x)$, to estimate the probability $P(\text{true}|x)$ for each text sample x. From another perspective of contrastive learning (Oord et al., 2018), this problem can also be viewed as learning a partial order of the probability distribution. Specifically, for two text samples x_1 and x_2, if $P(\text{true}|x_1) > P(\text{true}|x_2)$, the UD is expected to predict $D(x_1) > D(x_2)$. This contrastive view is essential for tasks with multiple options, i.e., learning to select from a few options based on the partial order given by UD.

Figure 2 compares the multi-task prompted formulation of T0 and the formulation of our UD. In the following, we will show how we use this formulation of UD to unify and solve discriminative tasks.

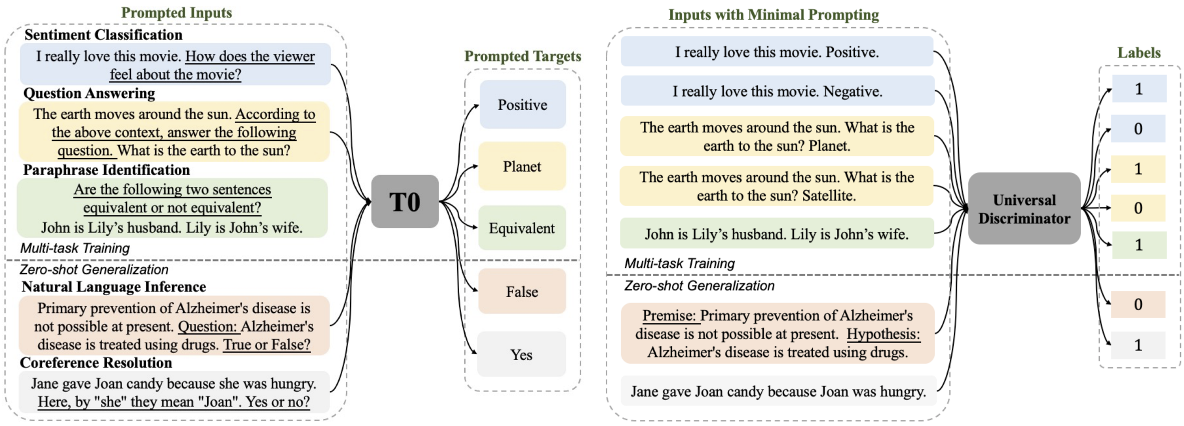

Figure 2: An overview that compares the multi-task prompted formulation of T0 (Sanh et al., 2021) and the formulation of our universal discriminator. The underlines mark natural language prompts. The universal discriminator uses a shared formulation of the discriminative tasks—determining whether a sample comes from the true data distribution of natural language.

3.3.1 Unifying Discriminative Tasks

We assume that for any task, the concatenation of input and the correct option follows the true data distribution of natural languages, while the concatenation of input and the other wrong options deviates much from the true data distribution.

Given this assumption, we claim that almost all discriminative tasks are equivalent to our defined task (i.e., estimating $P(\text{true}|x)$) above. Here, "equivalent" has bi-directional meanings: on one hand, there exists a reduction* from UD's task (say, task U) to any discriminative task (say, task A): given a piece of labeled training data for task A, we can generate several pieces of labeled training data for task U.

On the other hand, there exists another reduction from any discriminative task A to UD's task U: given a piece of testing data for task A, we can generate several pieces of testing data for task U such that by first predicting $D(\cdot)$ on them and then using a mapping from task U's outputs to task A's outputs, we can generate the answer for task A.

Based on the definition of discriminative tasks in § 3.2, there are two main categories, multi-choice tasks and yes/no tasks. We will discuss each category in detail as follows (also see Table 4 in appendix for specifics).

Multi-Choice Tasks For multi-choice tasks, we concatenate the text input x_{in} with each choice $\{c_i\}_{i=1}^{N_c}$ to form samples. For example, for multi-choice question answering, we concatenate the given paragraph and question with each answer candidate. See Table 4 for more task formulations. During training, the concatenated samples with the correct choice are given label 1 (true) for UD and the other incorrect ones are given label 0 (false). During testing, similarly, we concatenate the text input x_{in} with each choice $\{c_i\}_{i=1}^{N_c}$ to form several samples $\{(x_{in}, c_i)\}_{i=1}^{N_c}$ and ask UD for their $D(\cdot)$ scores. We then select the sample with the maximal $D(\cdot)$ score and output its corresponding choice.

Tasks with Yes/No Choices For yes/no tasks, we directly treat the text input x_{in} as a sample and assign its 0/1 label based on its yes/no label. During training, we use x_{in} with its assigned 0/1 label as UD's training data. During testing, we first get the output of UD on x_{in}, $D(x_{in})$, and then output answer yes/no based on whether $D(x_{in}) > 0.5$†.

Empirical experiments suggest that unifying tasks with Yes/No choices in such a new way can produce better zero-shot performance than using the same method for Multi-Choice Tasks. We provide two justifications here: First, the Yes/No answer tokens here don't contain specific information and thus the model cannot benefit from concatenation. Second, the two tokens Yes/No are asymmetric in the training dataset which may result in the model uniformly assigning higher scores for one of them no matter what the task input is.

Minimal Prompting A key principle we follow for task formulation is minimal prompting. From Table 4, one can see that our prompts are min-

*In complexity theory, a reduction is an algorithm transforming one problem A into another problem B such that a solution for problem B could also be used to solve problem A.

†We note that more delicate threshold search might be possible, but we find it performs well using a constant 0.5.

imal in the sense that they are mostly just concatenations of different elements from the raw input, discarding most of the previously instructive prompting words. This is very different from T0 (Sanh et al., 2021) and other generative approaches (Brown et al., 2020; Schick and Schütze, 2020) that add lengthy task descriptions with different wordings into the prompts.

We argue that there are two major benefits of minimal prompting. First, previous work (Liu et al., 2021b) has shown that zero-shot and few-shot performances are very sensitive to the prompts used for inference. Minimal prompting is more robust and requires less prompt engineering efforts at test time. This is especially important for true zero-shot real-world applications as there is no data available for choosing the right prompt. Second, as we will show in our experiments, UD performs much better with minimal prompts than lengthy descriptive prompts, while generative approaches do not work well with minimal prompts. This is also consistent with our motivation that all the NLP tasks share a common ability: "discriminating whether a text sample comes from the true data distribution" and UD is attempting to learn "what kind of concatenation between input and option makes it look like the true language?", which does not rely much on the descriptions for each task. On the other hand, T0 attempts to generate the answer directly basing on all the information it gets, so prompts provide an extra source of information and are helpful. See § 4.4.1 for our ablation study on minimal prompts.

Note that it is also important to use minimal prompts to resolve ambiguity in some cases. For example, consider the natural language inference (NLI) task that predicts whether a premise A entails a hypothesis B. Simply concatenating A and B is ambiguous, because the model cannot tell which is the premise. The model also is not aware that this is an NLI task. To resolve this kind of ambiguity, we use a minimal prompt "Premise: A. Hypothesis: B." instead, as shown in Table 4.

3.3.2 Architecture

UD can use any pre-trained encoder model as the backbone. In this work, we experiment with the T5 encoder and DeBERTa (He et al., 2021b). Since T5 is an encoder-decoder model, we only use the encoder part. For the T5 backbone, we perform mean pooling over the last-layer encoder features, followed by a dropout layer and a linear layer to predict a scalar logit. For the DeBERTa backbone, we use the last-layer feature of the first token, followed by a two-layer perceptron with dropout to also output a scalar logit. We train UD with the binary cross entropy loss.

3.4 A Generalized Universal Discriminator

To further study how the discriminative approaches work in combination with generative tasks, we also propose to experiment with a generalized version of UD (denoted as generalized UD).

Different from the previous UD that only uses an encoder as the backbone model, the generalized UD employs an encoder-decoder architecture. In the following, we experiment with the T5 model. Generalized UD takes both discriminative and generative tasks into consideration, and is jointly trained over both types of tasks at the same time.

For discriminative tasks, they are reformulated into binary classification tasks through minimal prompting, as is described in § 3.3.1. Specifically, it takes the minimal prompted texts into the encoder and uses the decoder to predict over {"Yes", "No"}. In such cases, generalized UD is optimized with the binary cross-entropy loss. For generative tasks, they take the form of "input-and-target" pairs. Generalized UD is fed with the textual inputs, and generates the targets through decoding. For generative tasks, generalized UD is trained to optimize the cross-entropy loss.

4 Experiments

4.1 Experimental Setup

We performed extensive experiments to validate the performance of the zero-shot generalization of our UD. We follow the same zero-shot setting as T0 (Sanh et al., 2021) by training on multi-task datasets and evaluating a held-out set of tasks that are never seen during training.

Datasets The original T0 training set consists of 38 tasks of 8 different types. There are in total 21/38 discriminative training tasks, with which we train the UD. The evaluation set covers four types of tasks, including natural language inference (RTE (Candela et al., 2006), CB (De Marneffe et al., 2019), ANLI/R1-R3 (Nie et al., 2020)), coreference resolution (WSC (Levesque et al., 2012), Winogrande (Sakaguchi et al., 2020)), sentence completion (COPA (Roemmele et al., 2011), StoryCloze (Mostafazadeh et al., 2017),

(a) On 11 discriminative test tasks following the T0 benchmark.

Base Model	Method	#Params	Natural Language Inference					Sentence Completion			Coreference		WSD	Avg.
			RTE	CB	ANLI1	ANLI2	ANLI3	COPA	Hella.	Story.	WSC	Wino.	WiC	
Decoder-only	GPT-3	175B	63.5	46.4	34.6	35.4	34.5	91.0	78.9	83.2	65.4	70.2	-	-
Decoder-only	GLaM	137B	56.3	39.3	39.7	35.5	34.1	90.0	76.7	81.1	82.1	71.3	50.6	59.7
MoE Decoder-only	GLaM	64B	66.8	33.9	40.9	38.2	40.9	90.0	77.1	82.5	83.5	73.4	50.5	61.6
Decoder-only	PaLM	540B	72.9	51.8	48.0	44.2	45.7	93.0	83.4	84.6	89.1	81.1	59.1	68.5
Decoder-only	FLAN	137B	78.3	64.1	47.7	43.9	47.0	90.6	56.4	92.2	80.8	67.3	-	-
ELECTRA	PE-CLS	335M	60.2	57.4	34.1	34.4	36.4	92.7	44.1	96.0	62.8	56.3	50.7	56.8
	PE-PROB	335M	54.0	49.2	32.3	33.3	33.5	81.9	36.7	89.5	64.3	50.7	50.9	52.4
	PE-REP	335M	69.0	61.3	36.1	35.0	39.4	91.2	47.0	96.8	70.0	56.2	51.1	58.5
DeBERTaV3	UD (ours)	304M	71.1	76.8	43.8	41.3	45.7	96.0	60.7	97.4	66.4	83.6	53.3	66.9
T5-Large	T0 ⋆	800M	75.1	55.5	32.9	32.3	33.7	84.6	28.2	94.0	63.0	54.6	51.2	55.0
	UD (ours)	400M	**83.8**	**80.4**	**36.8**	**34.2**	**42.2**	**90.0**	**56.1**	**96.4**	**68.3**	**62.9**	**54.6**	**64.1**
T5-XL	T0 †	3B	64.6	45.4	33.8	33.1	33.3	72.4	27.3	84.0	65.1	51.0	50.7	51.0
	T0 ⋆	3B	**79.7**	68.9	**43.1**	**38.5**	42.3	**94.1**	31.5	97.5	68.8	61.3	**54.1**	61.8
	UD (ours)	1.5B	78.7	**73.2**	41.2	36.3	**45.4**	94.0	**70.1**	**97.9**	**72.1**	**70.6**	53.0	**66.6**
T5-XXL	T0 †	11B	80.8	70.1	43.6	38.7	41.3	90.0	33.6	92.4	61.5	59.9	56.6	60.8
	T0 ⋆	11B	**85.8**	73.3	47.3	42.0	46.1	94.4	31.5	98.4	62.8	72.8	56.0	64.6
	UD (ours)	5.5B	80.5	87.5	49.0	42.9	48.8	95.0	77.4	**98.6**	73.1	82.2	57.1	72.0
	UD+ (ours)	5.5B	82.0	**89.3**	**53.4**	**48.1**	**51.0**	**96.0**	**78.9**	96.7	**75.0**	**86.4**	**58.5**	**74.1**

(b) On 13 discriminative BigBench tasks following the T0 benchmark

Model	T0-Large	UD-large	T0-XL	UD-XL	T0-XXL	UD-XXL	UD+-XXL
BigBench (Avg.)	39.6	**43.5**	44.8	**48.9**	47.4	55.5	**58.7**

(c) On 22 discriminative BBH tasks

Model	T0-Large	Flan-T5-Large	UD-Large	T0-XL	Flan-T5-XL	UD-XL	T0-XXL	Flan-T5-XXL	UD-XXL	UD+-XXL
BBH (Avg.)	38.9	39.5	**44.2**	40.4	44.6	**47.3**	45.0	49.4	51.3	**56.7**

Table 1: Zero-shot performance of our UD and baselines. Results in the first block are reported by previous work, respectively from GPT-3 (Brown et al., 2020), GLaM (Du et al., 2022), PaLM (Chowdhery et al., 2022), and FLAN (Wei et al., 2021). Note that we provide these reported results for reference, and do not compare directly. Some of the reported tasks are evaluated on the test split, while we follow the better baseline method T0 to report on validation splits. Results with † are reported by Sanh et al., and results with ⋆ are reproduced in our framework. We reproduced the three variants of prompting ELECTRA (Xia et al., 2022) under our setting, denoted as "PE-CLS", "PE-PROB", "PE-REP". Results for Flan-T5-Large/Xl/XXL (Chung et al., 2022) are reproduced by testing zero-shot performance on their released checkpoints. In the same group, T0 and Flan-T5 has 2x model parameters compared to UD. For abbreviation, we denote UD based on T5-XX as "UD-XX", e.g., UD-XL refers to UD based on the T5-XL model.

Hellaswag (Zellers et al., 2019)), and word sense disambiguation (WiC (Pilehvar and Camacho-Collados, 2018)). Following T0, we use accuracy on the validation split as the evaluation metric. For prompt-based baselines, we report the average accuracy over multiple prompts for each test task. Besides, we also evaluate zero-shot performance on 13 BigBench (Srivastava et al., 2022) tasks, which are also adopted by T0 (Sanh et al., 2021), and 22 BBH tasks (Suzgun et al., 2022), which are adopted by Flan-T5 (Chung et al., 2022).

Baselines We primarily compare our method with T0 (Sanh et al., 2021), which is a generative approach. Another baseline is prompting ELECTRA (Xia et al., 2022) which is a recent work on discriminative modeling. Since it was proposed in a different setting (i.e., a few-shot setting or direct zero-shot inference without any finetuning), we reproduced their method under our multitask zero-shot setting for comparison.

For a fair comparison, we follow T0 to use the T5-V1.1-LM-Adapted (Raffel et al., 2019) as the backbone model, and we experimented with three different scales, respectively 800M, 3B, and 11B. For UD, it only makes use of the encoder of T5-v1.1 and additionally replaces the output layer with a classification head.

In addition, we provide reported zero-shot results of several large language models (with hundreds of billions of parameters) for reference, including GPT-3 (Brown et al., 2020), GLaM (Du

Dataset	SOTA	UD+-XXL
QQP	**90.60**	90.44
DREAM	91.80	**94.95**
QuAIL	87.20	**88.13**
IMDB	97.30	**97.44**
AgNews	**95.58**	95.56
OBQA	87.20	**89.20**
STSB	92.30	**92.90**
CSQA	**84.90**	84.68
SST-2	97.30	**97.48**
QNLI	96.50	**96.56**
AbductiveNLI	89.80	**93.20**
VitaminC	91.10	**92.62**
MNLI	**92.10**	92.03
MCScript	97.30	**98.03**
MCScript 2.0	97.90	**98.01**
AdversarialNLI (r3)	53.50	**67.83**
COLA	**71.50**	71.42
Avg.	89.05	**90.62**

Table 2: Results on fully-supervised tasks for UD, which is based on the encoder of T5-xxl. Previous sota model (Tay et al., 2022) has 4x model parameters compared to UD.

et al., 2022), PaLM (Chowdhery et al., 2022), and FLAN (Wei et al., 2021). We also reproduce zero-shot results of a recent work Flan-T5 (Chung et al., 2022) by evaluating their released checkpoints on BBH tasks[‡]. Note that Flan-T5's training data sets are much broader than ours, so results for Flan-T5 here are only for reference but not a fair comparison.

Training During training, we truncate the input sequence to 256 tokens and use a batch size of 256. For optimization, we use the Adam optimizer with a fixed learning rate of 1e-5 and a dropout rate of 0.1. Each experiment is trained with 10, 8, and 5 epochs respectively for 800M, 3B, and 11B models.

4.2 Main Results on Zero-Shot Tasks

UD Zero-Shot Results The main results are presented in Table 1. We compare methods of similar scales. Results in Table 1(a) show that our UD substantially outperforms the T0 baseline on average by a large margin of around 9, 5, and 7 points respectively at Large, XL, and XXL scales. Comparing the results of UD-T5-Large, UD-DeBERTaV3, and prompting ELECTRA, both variants of UD also substantially outperform prompting ELECTRA by more than 6 points. On BIG-Bench datasets, results in Table 1(b) show that our UD outperforms the T0 baseline by a margin of around 4-8 points. On BBH datasets, results in Table 1(c) show that our UD constantly outperforms T0 and Flan-T5 by a margin of around 2-5 points, even though our UD is only trained on a small fraction of Flan-T5's training sets. Overall, these results demonstrate the advantages of UD at every scale, and a broad range of tasks compared with baselines.

Another interesting finding is that the advantages of UD significantly increase along with scaling. When scaling from Large-scale to XL-scale (i.e., around 3.75x of the parameters), the average performance improves by around 2 points. However, when scaling from XL-scale to XXL-scale (i.e., 3.6x of the parameters), the improvements of average zero-shot performance enlarge to 8 points. Based on the observation, we hypothesize that UD can achieve even better performance of zero-shot generalization if further scaling to an even larger models, which we leave to future work.

To further boost the zero-shot performance, we also train a new variant of UD at 11B scale by scaling to more training tasks, including the discriminative English tasks used in Wang et al. (2022), and the discriminative English tasks used in Tay et al. (2022). The new model is denoted as UD+. UD+ achieves the highest average accuracy among all the zero-shot evaluation tests.

Generalized UD Zero-Shot Results The zero-shot results of generalized UD on 11 T0 discriminative test tasks and on 13 Big-Bench tasks are respectively reported in Table 7(a) and Table 7(b) in appendix. In addition, to test how generalized UD performs on zero-shot generative tasks, we also report results on 15 generative tasks from Big-Bench. Results are presented in Table 7(c).

Analyses are as follows. First, comparing the results of generalized UD and T0, generalized UD still holds significant improvements on discriminative tasks. Second, comparing generalized UD with our previous UD (in Table 1), we observe there is a slight decrease in average performance, proving that adding generative tasks into training could have impacted a little bit, in trade for capability for handling generative tasks. Third, on 15 generative tasks, both generalized UD and T0 show comparable results.

[‡]T0 test sets are included in Flan-T5's training data sets, so we can't test its zero-shot performance on those data sets.

	Natural Language Inference					Sentence Completion			Coreference		WSD	Avg.
	RTE	CB	ANLI1	ANLI2	ANLI3	COPA	Hella.	Story.	WSC	Wino.	WiC	
UD (Minimal)	**83.8**	**80.4**	36.8	**34.2**	**42.2**	**90.0**	**56.1**	**96.4**	**68.3**	**62.9**	**54.6**	**64.1**
UD (Instructive)	72.2	64.5	**37.0**	33.4	39.7	85.3	45.2	96.0	65.4	53.9	50.9	58.5
T0 (Minimal)	61.6	**57.8**	30.6	30.3	33.4	67.2	**33.8**	66.6	60.9	52.8	**51.7**	49.7
T0 (Instructive)	**75.1**	55.5	**32.9**	**32.3**	**33.7**	**84.6**	28.2	**94.0**	**63.0**	**54.6**	51.2	**55.0**

Table 3: Zero-shot performance for UD and T0 respectively with instructive and minimal prompts. Instructive prompts are lengthy descriptions of tasks (Sanh et al., 2021), while minimal prompts use a simple concatenation of input data.

4.3 SOTA Results on Finetuned Tasks

To explore how UD performs on fully-supervised tasks, we finetuned UD for a wide range of downstream tasks and reported their results in Table 2. For each finetuning experiment, the maximum training epoch is set to be 10. We search a hyperparameter space with learning rate in {2e-5, 1e-5, 5e-6}, batch size in {32, 64, 128}. We select the best checkpoint using a validation set with early stopping.

From results in Table 2, we find that UD can achieve remarkable performance on most of the downstream tasks. We achieve state-of-the-art performance on 12 out of the 17 tasks we evaluated. The results also show that more challenging tasks (tasks that require more knowledge) will benefit more from the multi-task training period, especially some QA tasks.

4.4 Ablation Study

We have also conducted ablation studies to further explore how several factors affect the performance of zero-shot generalization. Please see appendix for further ablation studies on UD with different base models (§ C.1) and generalizing UD to a broader domain (§ C.2).

4.4.1 Instructive Prompts vs Minimal Prompts

UD employs minimal prompts that use simple concatenation, while previous approaches rely on lengthy instructive prompts to provide more detailed instructions (Sanh et al., 2021; Wei et al., 2021; Brown et al., 2020). Statistically, we count the average number of prompt words (excluding raw input) for both minimal and instructive prompts, and statistics are respectively 0.4 versus > 10. We compare these two types of prompts in the following experiment. We adopt the instructive prompts from T0 and apply them on UD without changing the discriminator formulation. To construct minimal prompts for T0, we remove all the instructive words similar to UD.

Results are shown in Table 3. We observe that minimal prompts yield better performance for UD than instructive prompts. In contrast, for T0, instructive prompts perform much better than minimal prompts. These results are consistent with our motivation that UD tends to unify the tasks better with a shared discrimination formulation. As a result, task-specific instructions are not necessary and might hurt generalization performance. Generative approaches, on the other hand, rely on instructive prompts to better distinguish different tasks and generate specific answers directly.

5 Conclusions

Universal Discriminator is a discriminating model for predicting whether a sample comes from the true data distribution, which is a new formulation for all discriminative NLP tasks. Experiments show that UD sets the new state-of-the-art for zero-shot generalization on many benchmarks. UD is high-performing with minimal prompting, and thus is more robust and applicable in practice. A generalized UD can also solve generative tasks at the same time which keeps UD's advantage on discriminative tasks and has comparable performance on generative tasks.

Inspite of this, generalized UD may not handle certain complex generation tasks very well (e.g., summarization) and its performance on generative tasks is not significantly better than other models' in comparison to its advantage on discriminative tasks over other models. We leave expanding UD to solve a broader range of generative tasks and achieve greater performance advantage as our future work.

References

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. 2021. Efficient large scale language modeling with mixtures of experts. arXiv preprint arXiv:2112.10684.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. CoRR, abs/2005.14165.

Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché-Buc, editors. 2006. Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers, volume 3944 of Lecture Notes in Computer Science. Springer.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. CoRR, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In proceedings of Sinn und Bedeutung, volume 23, pages 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.

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 P. 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, Zhifeng Chen, and Claire Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In ICML, volume 162 of Proceedings of Machine Learning Research, pages 5547–5569. PMLR.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. Advances in neural information processing systems, 27.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021a. Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021b. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. CoRR, abs/2111.09543.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning. Citeseer.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021a. GPT understands, too. CoRR, abs/2103.10385.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. arXiv preprint arXiv:2103.10385.

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.

Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. arXiv preprint arXiv:2106.13353.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3638–3652.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. arXiv preprint arXiv:2202.04538.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, pages 46–51.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In ACL, pages 4885–4901. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Mohammad Taher Pilehvar and José Camacho-Collados. 2018. Wic: 10, 000 example pairs for evaluating context-sensitive representations. CoRR, abs/1808.09121.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. arXiv preprint arXiv:2112.11446.

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. CoRR, abs/1910.10683.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning, pages 90–95.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In AAAI, pages 8732–8740. AAAI Press.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. 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 Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. CoRR, abs/2110.08207.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. CoRR, abs/2009.07118.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv preprint arXiv:2206.04615.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. arXiv preprint arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks.

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. CoRR, abs/2109.01652.

Mengzhou Xia, Mikel Artetxe, Jingfei Du, Danqi Chen, and Ves Stoyanov. 2022. Prompting electra: Few-shot learning with discriminative pre-trained models. arXiv preprint arXiv:2205.15223.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. arXiv preprint arXiv:1912.09637.

Yuan Yao, Bowen Dong, Ao Zhang, Zhengyan Zhang, Ruobing Xie, Zhiyuan Liu, Leyu Lin, Maosong Sun, and Jianyong Wang. 2022. Prompt tuning for discriminative pre-trained language models. arXiv preprint arXiv:2205.11166.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In ACL (1), pages 4791–4800. Association for Computational Linguistics.

A Examples of Minimal Prompt

Here we provide Table 4 for some examples of how to construct minimal prompted data according to § 3.3.1.

Category	Task Type	Our Minmal Prompt	Label
yes/no	Paraphrase Identification	John is Lily's husband. Lily is John's wife	1
		John is Lily's husband. Lily is John's mother.	0
	Natural Language Inference	Premise: Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44. Hypothesis: Dana Reeve had an accident.	1
		Premise: Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44. Hypothesis: Christopher Reeve had an accident.	0
multi-choice	Coreference Resolution	Jane gives Joan candy because Joan was hungry.	1
		Jane gives Joan candy because Jane was hungry.	0
	Question Answer	The earth moves around the sun. What is the earch to the sun? Planet	1
		The earth moves around the sun. What is the earch to the sun? Satellite	0
	Topic Classification	Open Source Apps Developer SugarCRM Releases Sugar.Sales 1.1. Science and technology	1
		Open Source Apps Developer SugarCRM Releases Sugar.Sales 1.1. Sports	0
	Sentence Completion	A boy is running down a track. The boy lifts his body above the height of a pole.	1
		A boy is running down a track. The boy stands on his hands and springs.	0
	Sentiment Classification	I really love this movie. Positive	1
		I don't like this movie. Negative	1

Table 4: Examples of how we unify discriminative tasks. The underlined text represents additional words not present in raw inputs. Note that this is just our implementation of the UD formulation and there can be other ways of task formulation under the UD framework. Some tasks can either be yes/no tasks or multi-choice tasks, depending on how options are provided.

B Full Experiment Results

B.1 Evaluation on Big-Bench

Here we report the full results for 13 tasks in the Big-Bench Srivastava et al. (2022), which is also utilized in original T0 paper (Sanh et al., 2021). All the tasks from BIG-Bench are ensured unseen in our training set for the zero-shot setting. The results are displayed in Table 5, where UD outperforms T0 by 4-8 points on different scales.

B.2 Evaluation on BBH

Here we report the full results for 22 discriminative tasks from BBH (Suzgun et al., 2022). For reference, we reproduce Flan-T5(Chung et al., 2022)'s zero-shot performance on BBH tasks by evaluating their

Model	code desc.	conce -ptual	known unknowns	logic grid	logic deduction	miscon -ceptions	novel concepts	strate -gyqa	wino -why	syllo -gisms	movie dialog	lang -uage_id	vita -minc	Avg.
UD-DeBERTaV3	76.7	64.1	76.1	39.9	54.9	50.2	50.0	59.9	45.8	50.4	57.7	13.3	61.5	53.9
T0-Large⋆	14.1	40.4	**60.4**	**38.0**	**41.2**	50.0	10.0	**52.3**	**49.7**	50.3	46.8	16.0	46.2	39.6
UD-Large	**51.7**	**54.4**	47.8	33.4	34.6	**50.2**	**26.5**	47.0	45.7	**50.6**	**51.7**	**16.3**	**55.8**	**43.5**
T0-XL⋆	23.4	48.1	64.6	**42.5**	**50.1**	**52.7**	25.0	53.1	**45.4**	50.2	47.7	19.0	60.0	44.8
UD-XL	**53.3**	**73.8**	**65.2**	37.2	37.8	48.0	**35.3**	**53.1**	45.3	**50.4**	**50.1**	**22.9**	**63.7**	**48.9**
T0-XXL†	36.7	62.5	63.0	39.6	55.4	**52.5**	15.6	52.7	47.4	**51.8**	53.8	20.7	64.7	47.4
UD-XXL	61.7	71.8	76.1	38.0	59.1	49.3	**61.8**	61.3	45.9	50.1	57.3	21.6	67.2	55.5
UD+-XXL	**63.3**	**82.5**	**84.8**	**39.2**	**67.5**	49.3	58.8	**64.2**	**47.5**	50.4	**57.9**	**27.3**	**70.2**	**58.7**

Table 5: Zero-shot performance of Universal Discriminator and T0 on Big-Bench test tasks used in T0 paper. Results with † are reported by Sanh et al., and results with ⋆ are reproduced in our framework.

Dataset	T0-Large	Flan-T5-Large	UD-Large	T0-XL	Flan-T5-XL	UD-XL	T0-XXL	Flan-T5-XXL	UD-XXL	UD+-XXL
boolean_expression	48.4	49.6	**64.0**	47.6	54.8	**68.4**	46.4	56.8	**68.4**	66.0
causal_judgement	56.2	59.4	**61.5**	58.8	59.9	**63.6**	62.0	60.9	**65.2**	63.6
data_understanding	30.4	18.8	**30.4**	38.8	34.8	**41.2**	**63.2**	56.8	51.6	53.2
disambiguation_qa	54.4	34.8	**68.4**	61.2	**66.8**	65.2	64.4	66.8	**67.2**	66.8
formal_fallacies	54.4	**55.6**	50.4	52.4	**54.0**	46.4	52.0	55.2	54.0	**58.8**
geometric_shapes	0.0	**21.6**	9.6	0.0	**20.0**	9.6	11.2	**31.2**	9.6	9.6
hyperbaton	**72.0**	59.6	71.2	52.4	58.8	**66.8**	63.2	70.8	68.0	**82.0**
logical_deduction_five_objects	34.8	**40.0**	32.8	38.8	**48.0**	39.2	46.4	53.6	58.4	**65.2**
logical_deduction_seven_objects	27.6	**40.4**	25.2	37.6	**52.4**	32.0	50.4	60.0	56.4	**67.2**
logical_deduction_three_objects	49.2	37.6	**60.4**	62.8	64.8	**69.2**	65.6	74.4	80.8	**83.2**
movie_recommendation	51.4	55.0	**60.4**	55.0	47.4	**69.6**	61.0	38.5	73.2	**78.8**
navigate	58.8	56.4	**63.6**	60.4	59.2	58.4	65.6	60.8	63.2	64.8
penguins_in_a_table	36.3	32.9	**36.3**	34.3	**42.5**	41.1	40.4	41.1	39.7	**46.6**
reasoning_about_colored_objects	39.2	**40.4**	36.4	41.6	47.2	**54.4**	56.8	61.6	57.2	**63.2**
ruin_names	23.0	22.6	**44.4**	21.8	**33.5**	24.4	17.8	34.7	35.6	**68.8**
snarks	48.3	56.1	**74.7**	45.5	55.6	**73.0**	55.1	72.5	75.3	**82.0**
sports_understanding	53.2	**55.6**	54.8	47.6	**52.4**	51.6	52.8	**60.0**	57.6	56.0
temporal_sequences	13.2	**25.2**	23.6	24.8	22.4	**63.2**	14.8	28.8	43.2	**60.8**
tracking_shuffled_objects_five_objects	**12.8**	12.4	12.0	12.8	12.0	**13.2**	12.0	15.2	12.4	**20.0**
tracking_shuffled_objects_seven_objects	7.6	8.4	**9.6**	8.8	**9.2**	8.4	8.0	13.2	8.4	**14.0**
tracking_shuffled_objects_three_objects	33.2	**33.6**	31.2	33.6	32.8	**34.8**	29.6	24.4	**33.6**	20.8
web_of_lies	51.2	**52.4**	51.2	51.2	**52.4**	47.6	50.8	50.0	50.4	**56.8**
Avg.	38.9	39.5	**44.2**	40.4	44.6	**47.3**	45.0	49.4	51.3	**56.7**

Table 6: Zero-shot performance of Universal Discriminator, T0, and Flan-T5 on BBH test tasks (Suzgun et al., 2022).

public checkpoints. All the tasks from BBH are ensured unseen in our training set for the zero-shot setting. The results are displayed in Table 6, where UD constantly performs better than T0 and Flan-T5 on all the scales even though Flan-T5 is trained on a much broader scope of tasks than UD is.

B.3 Experiment Results for Generalized Universal Discriminator

Here we report the zero-shot results of generalized UD on 11 T0 test tasks (Table 7(a)), 13 Big-Bench tasks selected by T0 paper (Table 7(b)), and 15 generative tasks from Big-Bench (Table 7(c)). We select the top 15 uncommon generative tasks from BigBench basing on ascending order of data size. We assume that tasks with smaller data sizes are less common and more likely to be unrelated to our training data and more suitable for zero-shot tests. We observe that generalized UD still holds significant improvements on discriminative tasks and gets comparable results on generative tasks.

C More Ablation Studies

C.1 Ablation on Base Models

We also study the effects of using different backbone pretrained models. We experiment with three backbone models of different types, respectively the encoder part of an encoder-decoder model, an encoder model, and a decoder model. Specifically, we use the T5 encoder, DeBERTa (He et al., 2021b), and GPT (Radford et al., 2018) respectively for these three types. It is noteworthy that though similar in architecture for both T5 encoder and DeBERTa, they are pretrained with different self-supervised language modeling tasks, which in fact leads to huge differences in zero-shot generalization, as we will show in Table 8.

Results of different backbone models are presented in Table 8. Among all three types of backbone models, the encoder backbone models appear to be the most suitable type of backbone, where both encoder models of two scales respectively achieve the best and the second best results, outperforming all the others by more than 5 points.

Using the same number of parameters (i.e., 1.5B), both DeBERTa-V2 and T5-Encoder significantly outperform GPT-XL, which demonstrates that a bidirectional architecture works better than the unidirectional architecture for the discriminator formulation. In addition, DeBERTa-V2 outperforms T5-Encoder by 7 points, implying that not only model architecture but also the self-supervised pretraining task determines the ability of UD discrimination. Models pretrained with masked language modeling tasks are more suitable for UD.

(a) On 11 discriminative test tasks following the T0 benchmark.

Method	**Natural Language Inference**					**Sentence Completion**			**Coreference**		**WSD**	Avg.
	RTE	CB	ANLI1	ANLI2	ANLI3	COPA	Hella.	Story.	WSC	Wino.	WiC	
T0-XL	**79.7**	68.9	43.1	38.5	42.3	**94.1**	31.5	**97.5**	**68.8**	61.3	**54.1**	61.8
GenUD-XL	71.5	**80.4**	**43.1**	**39.5**	**42.6**	94.0	**55.8**	96.7	63.5	**75.5**	52.8	**65.0**

(b) On 13 discriminative Big-Bench tasks following the T0 benchmark.

Model	code desc.	conce -ptual	known unknowns	logic grid	logic deduction	miscon -ceptions	novel concepts	strate -gyqa	wino -why	syllo -gisms	movie dialog	lang -uage_id	vita -minc	Avg.
T0-XL	23.4	48.1	64.6	**42.5**	50.1	**52.7**	25.0	53.1	45.4	50.2	47.7	**19.0**	60.0	44.8
GenUD-XL	**60.0**	**64.1**	**69.6**	38.2	**52.8**	48.9	**44.1**	**57.1**	**46.5**	**50.4**	**50.9**	15.5	**66.8**	**48.9**

(c) On 15 generative tasks from Big-Bench

Model	auto debugging	simple arithmetic	repeat copy logic	sufficient information	simple text editing	scientific press release	codenames	emoji movies	penguins in a table	few shot nlg	operators	tense	geometric shapes	chinese remainder theorem	temporal sequences	Avg.
T0-XL	11.2	6.7	**25.8**	33.8	7.5	**6.7**	**44.8**	**8.7**	**11.4**	17.4	**10.5**	80.7	0.0	0.0	14.0	**18.6**
GenUD-XL	**15.5**	6.7	8.2	**34.4**	**12.6**	6.4	25.1	0.0	8.1	**20.5**	3.7	**80.9**	0.0	0.0	**33.5**	17.0

Table 7: Zero-shot performance for generalized UD and T0 on discriminative and generative tasks. We select the top 15 uncommon generative tasks from BigBench basing on ascending order of data size. (We assume that datasets with smaller sizes are less common, and more suitable for zero-shot tests.) The metrics are respectively accuracy for discriminative tasks and ROUGE1 for generative tasks. "GenUD" denotes our generalized UD method.

	Base Model	**Natural Language Inference**					**Sentence Completion**			**Coreference**		**WSD**	Avg.
		RTE	CB	ANLI1	ANLI2	ANLI3	COPA	Hella.	Story.	WSC	Wino.	WiC	
Encoder	DeBERTa-V3 (304M)	71.1	76.8	43.8	41.3	45.7	96.0	60.7	97.4	66.4	83.6	53.3	66.9
	DeBERTa-V2 (1.5B)	77.6	80.4	43.2	39.3	44.8	95.0	67.2	98.2	74.0	82.1	56.0	68.9
Enc-Dec	T5-Encoder (400M)	75.1	55.5	32.9	32.3	33.7	84.6	28.2	94.0	63.0	54.6	51.2	55.0
	T5-Encoder (1.5B)	79.7	68.9	43.1	38.5	42.3	94.1	31.5	97.5	68.8	61.3	54.1	61.8
Decoder	GPT-XL (1.5B)	71.1	75.0	30.4	31.8	37.8	71.0	40.9	87.7	62.5	54.5	50.3	55.7

Table 8: Ablation study on different backbone models. We experiment with base models of different architectures and scales. "Enc-Dec" refers to models that are pretrained in an encoder-decoder manner.

Setting	Accuracy
True Data vs Manually-Generated Data	80.0
True Data vs Model-Generated Data	74.4

Table 9: The accuracy of UD discriminating real data and generated data. Each time we feed UD with a real sample x from the real-world data distribution, and a generated sample x'. If UD assigns higher score to x than x' (i.e., $D(x) > D(x')$), it is considered an accurate prediction. We experimented with two approaches to generate data, manual generation and model-based generation.

The impacts of the architecture and pretraining tasks of backbone models are even larger than the influence of scale, as we also observe that an encoder model with 300M parameters (i.e., DeBERTaV3) achieves much better performance than the T5 encoder and GPT-XL with 1.5B parameters.

C.2 How Well UD Generalizes to a Broader Domain?

In the previous sections, we have trained UD to solve the task of discriminating whether a text sample comes from the true data distribution of natural language. So far we have constrained the problem to supervised labeled tasks. However, this discrimination problem formulation is in fact general and can be applied to a broader domain of natural language. We conduct the following experiment to see how UD generalizes.

To test whether a model discriminates against the true data distribution, a straightforward way of verification is to compare the probability of real data with that of some generated, fake data. This form of verification is not specific to any downstream task and can be viewed as generalizing to a broader domain. Formally, given a text sample x, let $D(x)$ be the output of UD, which estimates the probability that x is sampled from the true data distribution, i.e., $P(\text{true}|x)$. Given a true data sample x and a generated data sample x', we expect a well-trained UD to predict $D(x) > D(x')$.

Specifically, we randomly select 2,600 real data samples x from the validation set of the T0 training data and generate the data x' in two different ways: model-based generation and manual generation.

For a model-based generation, we utilize the T0-Large model with a paraphrase prefix "Paraphrase the sentence:" to generate data x'. It is expected that the generated samples x' are similar to true samples x to some extent but demonstrate some flaws that are unique to generated data. For a manual generation, we manually create some conflict or contradiction in the real sample x. Specifically, we manually attach wrong answers to the original data and obtain x' , which is similar to what we have done in constructing negative samples in our main framework.

We then use our universal discriminator based on T5-Encoder Large to compute the probability $D(x)$ and $D(x')$ for both real and generated data. As displayed in Table 9, we find that the universal discriminator assigns a higher score for x than x' 80% of the time for manually-generated data. When tested with model-generated data, UD assigns a high probability for real data in 74% of the cases. This is probably because manually generated data are more paradoxical and logically incoherent and thus are easier for UD to discriminate. Overall, these results demonstrate that the discrimination ability of UD is not limited to the downstream tasks on which it was trained, but is also generalizable to a broader domain of text data. This indicates a possibility of extending UD to other scenarios such as model pretraining and generation tasks.