

---

# Few-Shot Out-of-Domain Transfer of Natural Language Explanations

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Recently, there has been an increasing interest in models that generate natural  
2 language explanations (NLEs) for their decisions. However, training a model to  
3 explain its decisions in natural language requires the acquisition of task-specific  
4 NLEs, which is time- and resource-consuming. A potential solution is the out-of-  
5 domain transfer of NLEs, where explainability is transferred from a domain with  
6 rich data to a domain with scarce data via few-shot transfer learning. In this work,  
7 we introduce and compare four approaches for few-shot transfer learning for NLEs.  
8 We transfer explainability from the natural language inference domain, where a  
9 large dataset of human-written NLEs already exists, to the domains of hard cases  
10 of pronoun resolution, and commonsense validation. Our results demonstrate that  
11 few-shot transfer far outperforms both zero-shot transfer and single-task training  
12 with few examples. We also investigate the scalability of the few-shot transfer of  
13 explanations, both in terms of training data and model size.

## 14 1 Introduction

15 Recent developments have made it possible to have AI models learn from natural language expla-  
16 nations (NLEs) for the ground-truth label at training time and generate such explanations for their  
17 decisions at deployment time [Park et al., 2018, Camburu et al., 2018, Hendricks et al., 2016, Kim  
18 et al., 2018, Ling et al., 2017, Rajani et al., 2019, Camburu et al., 2020, Narang et al., 2020, Kumar  
19 and Talukdar, 2020]. Such models are inspired by how humans learn (not only from labels but also  
20 from demonstrations and explanations Lombrozo, 2012, 2006) and explain themselves in natural  
21 language. For example, at training time, a self-driving car would not only be told that it has to stop  
22 in a certain environment, but it would additionally be told why this is the case, e.g., “Because the  
23 traffic light in front is red.”. When deployed, the self-driving car would be able to provide such  
24 natural language explanations for its actions, reassuring the passengers. Hence, such models are  
25 highly appealing.

26 In order to train a model to generate NLEs, it is required that humans annotate a training dataset with  
27 NLEs. However, large datasets of explanations, such as e-SNLI Camburu et al. [2018], are time-  
28 consuming and expensive to gather. One approach to solve this problem is to transfer explanations  
29 from a different domain, via few-shot or zero-shot transfer learning. The usual setup for few-shot  
30 out-of-domain transfer learning is when there is transfer learning from a “parent” task with abundant  
31 training examples, to a “child” task that only has a few training examples Thrun [1996], Ravi and  
32 Larochelle [2017]. In few-shot NLE generation, apart from the few training NLEs on the child task  
33 and the abundant NLEs on the parent task, we assume the availability of abundant training labels for  
34 both tasks. This makes it a novel setup in few-shot learning. In this work, we introduce four few-shot  
35 transfer learning methods for NLEs that utilize the abundant training labels for both parent and child  
36 tasks.

To our knowledge, there are only two existing works that address few-shot out-of-domain transfer of NLEs. The first work introduces the WT5 model Narang et al. [2020] as a proof-of-concept for NLE transfer across domains without any training NLEs on the child task (zero-shot). However, this is presented only as a non-primary experiment with cherry-picked examples and is not studied in depth. Furthermore, to obtain these results, the authors use the largest T5 model (with 11B parameters) Raffel et al. [2019], which is inaccessible for researchers with limited computational resources. The second relevant work Erliksson et al. [2021] introduce one method for few-shot out-of-domain transfer of NLEs, but only employ automatic evaluation metrics, which have been shown to only weakly correlate with human judgment [Kayser et al., 2021].

In this work, we are the first to comprehensively study few-shot out-of-domain transfer learning of NLEs. We introduce and compare four training approaches for out-of-domain few-shot transfer of NLEs. We instantiate our few-shot learning approaches on e-SNLI [Camburu et al., 2018] as parent task and WinoGrande [Sakaguchi et al., 2020] and ComVE [Wang et al., 2020] as child tasks. We show the extent to which few-shot out-of-domain transfer learning of NLEs is currently feasible and provide insight into which learning techniques work best in this setup. We perform extensive human and automatic evaluation and show that few-shot out-of-domain transfer significantly outperforms the single-task and zero-shot baselines. We also investigate the scalability of the best approaches, both in terms of data and model sizes.

## 2 Experimental Setup

### 2.1 Datasets

**e-SNLI.** The task of natural language inference Dagan et al. [2006] is a common task for measuring natural language understanding. The task consists of a premise and a hypothesis for which one has to determine if they are in a relation of (i) *entailment* (if the premise entails the hypothesis), (ii) *contradiction* (if the hypothesis contradict the premise), or (iii) *neutral* (if neither entailment nor contradiction holds). The e-SNLI dataset Camburu et al. [2018] consists of human-written explanations on top of the influential Stanford Natural Language Inference (SNLI) dataset Bowman et al. [2015]. An example from e-SNLI is:

**Premise:** An adult dressed in black holds a stick.  
**Hypothesis:** An adult is walking away, empty-handed.  
**Label:** contradiction  
**Explanation:** Holds a stick implies using hands so it is not empty-handed.

We select e-SNLI as parent dataset due to its large size (approx. 570K instances) and high-quality explanations.

**WinoGrande.** We select WinoGrande (WG) Sakaguchi et al. [2020] as a child task, since it requires implicit knowledge, which we want to capture in the NLEs. The WinoGrande dataset consists of 40,398 binary questions of pronoun resolution that follow the Winograd Schema format Levesque et al. [2012]. Because of the lack of a publicly available test set (testing happens through its leaderboard,<sup>1</sup> which has submission limitations), we do a random split of the original training dataset into 39,130 training instances (called WG-train) and 1,268 validation instances (called WG-dev). For testing, we use the original WinoGrande development set, which we denote by WG-test. We manually construct NLEs for 100 examples from WG-train, 50 examples from WG-dev, and 100 examples from WG-test. An NLE example is:

The geese prefer to nest in the fields rather than the forests because in the \_\_\_\_ predators are very visible.  
**Options:** fields, forests. **Answer:** fields.  
**Explanation:** The fields are more open spaces than the forests, hence predators are more visible there.

**ComVE.** We select Commonsense Validation and Explanation (ComVE) Wang et al. [2020] as a child task, because it is a commonsense reasoning task for which there are good-quality human-

<sup>1</sup><https://leaderboard.allenai.org/winogrande/submissions/public>

Table 1: T5 input/target formats for each task, used for all models.

Task	Input Format	Target Format
e-SNLI	explain nli premise: [premise] hypothesis: [hypothesis]	[relation] explanation: [explanation]
WinoGrande	explain schema: [schema start] _ [schema end] options: [option 1], [option 2].	[correct option] explanation: [explanation]
ComVE	explain ComVE Sentence 1: [statement 1] Sen- tence 2: [statement 2]	[nonsensical statement id] explanation: [explanation]

generated NLEs. ComVE consists of three tasks: A, B, and C, where only tasks A and C are relevant for this work. ComVE-A is the classification task of identifying which statement out of a pair of statements makes sense and which does not. On the other hand, the ComVE-C task provides the statement (from the pair) that does not make sense and requires the model to generate NLEs for why that is the case. In order to form a classification task with explanations, we merge task A and task C by matching the nonsensical statements, as suggested by Majumder et al. [2021]. The resulting task can be described as: “given a pair of sentences, identify which one does **not** make sense, and explain why”. Here is an example from the resulting ComVE dataset:

**Statement 1:** He drinks milk.  
**Statement 2:** He drinks apple.  
**Answer:** Statement 2 does not make sense.  
**Explanation:** An apple is a whole food and unable to be drunk without being juiced.

The resulting ComVE dataset consists of 10,000 training, 1,000 validation, and 1,000 test instances. Each instance consists of a pair of statements, a label, and three human-generated NLEs. We use all three NLEs per example only in the full test set. For training, we use up to one NLE per example. For human evaluation, we sample the test dataset down to 100 instances, to save human-annotation costs.

## 2.2 Base Model

We use the T5 Raffel et al. [2019] generative language model due to its good generative abilities, and because it is used in the WT5 model to generate high-quality NLEs Narang et al. [2020]. More specifically, we choose the “Base” model Raffel et al. [2019] with 220M parameters (we call it T5-Base) due to its good trade-off of performance and computational requirements.

For T5, tasks are distinguished only via their task-specific input/target formats. We follow the input/target format for e-SNLI from Narang et al. [2020], and obtain the input formats for WG and ComVE in a similar manner (see Table 1). When training on examples without NLEs, “explain” and “explanation:” are not included in the input/target format. We have observed in early experiments that the exact choice of input/target formats does not significantly affect performance.

## 2.3 Few-Shot Transfer Learning Methods

In Table 2, we describe all models that we use. This includes the four new few-shot transfer learning methods for NLE generation: MT-MT, MT-FT, FT-MT, and FT-FT. These correspond to all combinations of fine-tuning and multi-task learning between the parent task, the child task, and the number of NLEs that we use for few-shot transfer. Note that when training on the union of the child dataset (CT) and [number] of NLEs, we avoid repeating examples from CT that overlap with the [number] of NLEs. Along with the four few-shot transfer learning methods, we add two single-task baseline approaches and two zero-shot transfer approaches.

The two single-task baselines aim to verify the extent to which the parent task helps with the transfer of NLEs. The first single-task baseline, T5b(CT, [number]), is trained on the child task with [number] NLEs. The second single-task baseline, T5b(CT)-ft([number]), is first trained on the child task and then fine-tuned on [number] NLEs (few-shot).

To measure the contribution of the few added training NLEs, we introduce two zero-shot baselines, called WT5-MT and WT5-FT. The WT5-MT zero-shot baseline is the WT5 training approach [Narang et al., 2020], which consists of multi-task learning on the union of the e-SNLI, the SNLI,

Table 2: Model names: legend. Notations are introduced from top row to bottom row.

Abbreviation	Meaning
CT	The training dataset of the child task, without explanations.
[number]	A [number] of training examples from CT with explanations.
T5b([datasets])	Multi-task learning of T5-Base on the union of the [datasets].
[model]–ft([datasets])	Fine-tuning of the [model] on the union of the [datasets].
MT–MT([number])	T5b(e-SNLI, CT, [number])
MT–FT([number])	T5b(e-SNLI, CT)–ft([number])
FT–MT([datasets])	T5b(e-SNLI)–ft([datasets])
FT–FT([number])	T5b(e-SNLI)–ft(CT)–ft([number])
WT5–MT	T5b(e-SNLI, SNLI, CT)
WT5–FT	T5b(e-SNLI, SNLI)–ft(CT)
T5–Large: [model]	The [model] with T5–Large instead of T5–Base.
Heuristic baseline	A ComVE baseline that uses the correct statement as an NLE.

and the child dataset. WT5–FT is a variation of WT5 that uses fine-tuning on the child task instead of multi-task learning. These WT5 baselines combine in multi-task setting the e-SNLI and SNLI datasets, to train the model to switch between classification and NLE generation for a better zero-shot downstream performance. For ComVE, we also add a heuristic baseline (as suggested by Majumder et al. [2021]), given by selecting as an NLE the correct statement of the pair of statements. This baseline serves to judge the triviality of the NLEs generated by the other approaches.

## 2.4 Training Details

The training objective is given by cross-entropy loss with targets as described in Table 1. We use the AdamW optimizer [Loshchilov and Hutter, 2019] and linear learning rate scheduler with warmup over 10% of the training. The hyperparameters’ ranges for all models are described in Appendix A. No early stopping is needed, because we use a learning rate scheduler and the number of training epochs is a hyperparameter. We do not use gradual unfreezing Howard and Ruder [2018], because it has been shown that gradual unfreezing does not help when applied to the T5 language model Raffel et al. [2019].

At each stage of training, the best hyperparameter combinations are either selected by the perplexity relative to target NLEs on the dev set of the child task (CT), by dev accuracy on CT, or by NLE perplexity on the e-SNLI dev set, whichever is most suitable. The selection criteria for each model, along with the best hyperparameters are given in Table 7 in Appendix A. Model-generated explanations are obtained via beam search with a beam width of 5.

## 2.5 Human Evaluation

We use Amazon Mechanical Turk to evaluate the model-generated NLEs, with three annotators per instance. The evaluation procedure for each test example is in three steps. For the evaluation procedure we follow Majumder et al. [2021] and Kayser et al. [2021]. First, annotators are required to predict the correct classification label for the example. This forces them to think about resolving the example themselves. Second, they have to select one of four options for whether the NLE is a valid and satisfactory explanation to justify the selected label: Yes, Weak Yes, Weak No, or No. Third, they have to select shortcomings of the explanation, if any. The multiple-choice options are: “does not make sense”, “insufficient justification”, “irrelevant to the task”, “too trivial”, and “none”. These choices are to guide the annotators to think about the answer to the main question about NLE quality.

As suggested by Kayser et al. [2021], for each example, the annotators are provided with two (shuffled) NLEs, one from a model and one from the test set. This serves as grounding to the annotator’s score of the model-generated NLE. The annotators were paid 1\$ per 10 pairs of NLEs.

Additionally, there are multiple checks placed in the data collection form to ensure high-quality annotations. Most notably, in each group of 10 instances, at least 90% of the labels have to be answered correctly and at least 90% of the ground-truth NLEs have to be annotated by Yes or Weak Yes. These are reasonable assumptions for both WG and ComVE, judging by the quality of the ground-truth and model-generated NLEs. The final check requires that at most 80% of the

Table 3: Performance of models with WinoGrande as child task. The columns Yes, Weak Yes, Weak No, and No present the percentage of NLE validity scores given by the human annotators. Note that only correctly classified examples are included in these scores. The final column shows the inter-annotator agreement measured by Fleiss’ kappa. Perplexity relative to the 100 test NLEs on WinoGrande was omitted, because more reliable human evaluation scores are available.

Models	SNLI acc%	e-SNLI ppl	WG acc%	Acc @100	Yes%	Weak Yes%	Weak No%	No%	NLE score	Fleiss’ kappa
T5b(WG)–ft(50)	n/a	n/a	59.7	63	18.0	18.0	13.2	50.8	34.4	0.09
T5b(WG, 50)	n/a	n/a	57.2	66	20.2	15.2	15.2	49.5	35.4	0.18
WT5–FT	38.0	3.60	<b>60.2</b>	65	4.6	4.1	4.6	86.7	8.9	0.05
WT5–MT	<b>91.0</b>	2.27	58.0	55	4.2	3.0	<b>4.2</b>	88.5	7.7	0.17
MT–MT(50)	89.3	2.27	53.6	49	14.3	14.3	13.6	57.8	28.3	0.12
MT–FT(50)	90.2	<b>2.22</b>	56.0	63	<b>26.5</b>	17.5	18.0	<b>38.1</b>	<b>44.1</b>	0.12
FT–MT(WG, 50)	83.6	2.27	54.6	54	14.8	14.8	13.0	57.4	29.0	0.07
FT–FT(50)	68.2	2.33	58.2	65	23.6	<b>21.5</b>	13.3	41.5	<b>42.4</b>	0.21

model-generated NLEs should be annotated by Yes or Weak Yes. We included this check to ensure that the annotators are more critical, and we estimated this threshold manually.

For each of the two child tasks, all models are evaluated on 100 examples from the test dataset of the task, for which the evaluation procedure requires there to be ground-truth NLEs to be used for grounding the annotator’s score. Similarly to Camburu et al. [2018], the NLE evaluation is only done on correctly labeled examples, as it is expected that an incorrect label is not supported by the model with a correct explanation. For each model, we report the percentage of results for each of the four possible responses given by the annotators: Yes, Weak Yes, Weak No, and No. See Appendix B for pictures of the forms that were used to collect the data from the annotators.

### 3 Results

Following Kayser et al. [2021], we use an aggregated score (we call “NLE score”) of the four categories (Yes, Weak Yes, No, Weak No) to compare the NLE generation quality, where Yes, Weak Yes, Weak No and No are given weights 1, 2/3, 1/3, and 0, respectively. This aggregation has two goals: first, to provide a single metric to compare the methods, and second, to account for the subjective nature of choosing between close labels such as Yes and Weak Yes.

For every model comparison, we report the statistical significance via the paired Student’s t-test for equal variances [Yuen and Dixon, 1973], with single-tailed p-values and 0.05 statistical significance threshold. We assume that all individual scores are independent.

For all models, we report the inter-annotator agreement on the scores (Yes, Weak Yes, Weak No, No) via the Fleiss’ kappa measure [Fleiss et al., 1971]. Higher values of Fleiss’ kappa mean that the annotators agree more about the scores. The kappa values can be interpreted as suggested by Landis and Koch [1977], where negative values signify poor agreement, values between 0.01 and 0.20 are slight agreement, and values between 0.21 and 0.40 are fair agreement. In this work we do not obtain values higher than 0.40, and most values are around or higher than 0.10.

#### 3.1 WinoGrande

The results in Table 3 show that out of the four compared approaches, only MT–FT and FT–FT outperform all baselines in terms of the aggregated NLE score, in a statistically significant way, with p-values of 0.04 or lower ( $p < 0.05$ ). Amongst the two best approaches, MT–FT outperforms FT–FT in terms of NLE score, but the comparison is not statistically significant, with  $p = 0.34$ . Although the two approaches perform similarly, we select MT–FT over FT–FT for the method scalability experiments in Section 3.4 because of its slightly better NLE score (44.1 vs. 42.4). Both MT–FT and FT–FT, which use the 50 child task NLEs in separate training regimes, significantly outperform both MT–MT and FT–MT, which use a combination of WG with 50 NLEs instead. This suggests that the 50 explanations require their own training regime, as they are insignificant relative to the sizes of WG (approx. 40k) and e-SNLI (approx. 570k). Another possible explanation of the low quality of NLEs of MT–MT and FT–MT could be their close-to-chance-level WG accuracy (53.6% and 54.6%, resp.),

Table 4: Performance of models with ComVE as child task. The columns Yes, Weak Yes, Weak No, and No present the percentage of NLE validity scores given by the human annotators. Note that only correctly classified examples are included in these scores. The final column shows the inter-annotator agreement measured by Fleiss’ kappa.

Models	SNLI acc%	e-SNLI ppl	ComVE acc%	Acc @100	Yes%	Weak Yes%	Weak No%	No%	NLE score	Fleiss’ kappa
T5b(ComVE)–ft(50)	n/a	n/a	<b>87.8</b>	88	21.6	7.6	4.2	66.7	28.0	0.3
T5b(ComVE, 50)	n/a	n/a	83.1	79	19.4	5.5	3.8	71.3	24.3	0.29
WT5–FT	86.6	2.25	85.7	85	13.7	11.4	<b>3.1</b>	71.8	22.4	0.08
WT5–MT	<b>90.8</b>	2.27	76.2	72	11.1	10.6	5.6	72.7	20.1	0.11
MT–MT(50)	<b>91.0</b>	<b>2.2</b>	82.8	82	21.5	<b>20.7</b>	7.7	50.0	37.9	0.1
MT–FT(50)	90.4	2.25	80.6	79	22.4	<b>21.5</b>	5.5	50.6	38.5	0.33
FT–MT(ComVE, 50)	75.9	2.53	85.5	76	24.6	12.7	10.5	52.2	36.5	0.19
FT–FT(50)	87.6	2.27	86.5	79	<b>33.3</b>	15.6	7.2	<b>43.9</b>	<b>46.1</b>	0.17
Heuristic baseline	n/a	n/a	100	100	<b>31.3</b>	13.3	4.3	51.0	41.7	0.06

201 especially given that they guess the labels of only 49 and 54 out of the 100 examples used for human  
202 evaluation, respectively.

203 Fine-tuning twice in the FT–FT model T5b(eSNLI)–ft(WG)–ft(50) results in a decreased accuracy  
204 relative to SNLI (68.2%) and decreased e-SNLI perplexity (2.33 vs. 2.27 for the second-worse),  
205 but exhibits relatively high WG accuracy (58.2%) compared to the other three transfer learning  
206 approaches. More generally, the multi-task learning approaches that include the parent task (WT5–  
207 MT, MT–MT, and MT–FT) outperform their fine-tuning counterparts (WT5–FT, FT–MT, and FT–FT)  
208 in terms of e-SNLI accuracy and perplexity.

209 The four main multi-task models underperform their single-task counterparts in terms of WG accuracy.  
210 This is likely due to the domain shift between parent and child tasks. All non-zero-shot models  
211 significantly outperform the zero-shot baselines (WT5–FT and WT5–MT) in terms of NLE score  
212 across all four categories (Yes, Weak Yes, Weak No, No). Surprisingly, the WT5–FT model exhibits  
213 catastrophic forgetting of the e-SNLI task, with only 38% accuracy and 3.6 test perplexity. On the  
214 other hand, it exhibits the highest WG accuracy at 60.2%.

215 For qualitative results, see Appendix C where six NLEs from all models are presented. While most  
216 of the correct model-generated NLEs are trivial, such as this example from the FT–FT model:

217 Logan ended the one-side relationship they were having with Robert, which made  
218 \_\_\_\_ feel relieved.  
219 **Answer:** Logan.  
220 **Explanation:** Logan ending a one-side relationship makes Logan feel relieved.

221 there are non-trivial NLEs generated by the models, such as the two below generated by MT–FT:

222 Christopher declined when asked by Ian to play a game of chess because \_\_\_\_ liked  
223 to play intellectual games.  
224 **Answer:** Ian. **Explanation:** Chess is a type of intellectual game, and Christopher  
225 declined when asked by Ian to play a game of chess.

226 Brett was browsing the internet while he found the information unlike Randy, \_\_\_\_  
227 prefers using books.  
228 **Answer:** Randy. **Explanation:** If Brett is browsing the internet, then he is more  
229 likely to find information than Randy, who prefers to use books.

### 230 3.2 ComVE

231 The results of the human evaluation, which are given in Table 4, show that the FT–FT model  
232 significantly outperforms all single-task and zero-shot baselines, and all three other compared  
233 methods, in terms of NLE score, in a statistically significant way, with p-values of at most 0.03.  
234 Its performance is also higher than that of the heuristic baseline (46.1 vs. 41.7), but the p-value is  
235 only 0.13, which is not statistically significant. Looking more closely, the No score of FT–FT is  
236 higher than that of the heuristic baseline (43.9% versus 51.0%) with  $p = 0.05$ , and the difference is

Table 5: Automatic evaluation of the models with ComVE as child task. B-1,2,3,4 stand for BLEU-1,2,3,4.

Models	ComVE ppl	B-1	B-2	B-3	B-4	METEOR	BERTScore
T5b(ComVE)–ft(50)	5.21	<b>45.2</b>	<b>29.5</b>	<b>19.5</b>	<b>13.1</b>	<b>21.5</b>	<b>83.4</b>
T5b(ComVE, 50)	9.51	27.4	16.6	10.2	6.4	19.1	81.8
WT5–FT	65.25	24.6	15.1	9.7	6.5	13.5	74.8
WT5–MT	36.15	22.8	12.0	6.4	3.6	12.7	71.5
MT–MT(50)	8.02	34.5	19.2	10.8	6.3	20.3	81.8
MT–FT(50)	<b>5.11</b>	43.5	26.3	16.5	10.6	20.0	83.1
FT–MT(ComVE, 50)	8.18	33.6	18.8	10.9	6.2	20.8	82.1
FT–FT(50)	<b>5.13</b>	<b>44.4</b>	27.5	17.5	10.7	<b>21.2</b>	<b>83.6</b>
Heuristic baseline	n/a	40.8	25.8	17.2	12.0	18.7	81.4

distributed almost equally across Yes, Weak Yes, and Weak No scores, slightly in favor of FT–FT. We select FT–FT for the method scalability experiments in Section 3.4, because it outperforms all other methods.

In terms of ComVE test accuracy, FT–MT and FT–FT perform the best out of the four main models, which, like in the WinoGrande results, suggests that the parent and child tasks need separate training regimes in order to preserve the classification accuracy on the child tasks.

Similar to the WinoGrande models, amongst all four main approaches, the multi-task (MT) models that include the parent task outperform their fine-tuning (FT) counterparts in terms of preserving the accuracy and perplexity on the parent task (e-SNLI). The FT–MT model performs the worst out of the proposed models in terms of e-SNLI perplexity (2.53 vs. 2.27 for the second-worst) and accuracy (75.9% vs. 87.6% for the second-worst). This suggests that the NLE information from the parent task has been overwritten to a larger extent. This can also explain why the FT–MT model produces the lowest NLE score (36.5) and the highest percentage of No (52.2%) scores out of all four main models.

For qualitative results, see Appendix C, where six NLEs from all models are presented. Upon inspection, the correct model-generated NLEs are one of two types. The first is a repetition of the correct statement in the cases when it explains why the nonsensical sentence is wrong, e.g., from T5b(ComVE)–ft(50):

**Statement 1:** The fire will burn you if you touch it.

**Statement 2:** The fire will cool you if you touch it.

**Answer:** Statement 2 does not make sense. **Explanation:** The fire will burn you if you touch it.

The second type are negations of the nonsensical statement. E.g., from MT–MT:

**Statement 1:** He inserts his thumb into the car ignition switch.

**Statement 2:** He inserts the keys into the car ignition switch.

**Answer:** Statement 1 does not make sense. **Explanation:** One cannot insert his thumb into the car ignition switch.

### 3.3 Automatic Evaluation

In Table 5, we additionally evaluate all compared models on the full ComVE test set (1,000 examples with three NLEs per example) via automatic metrics following Kayser et al. [2021]. Automatic evaluation provides additional insights to the human evaluation, since the human evaluation was done on ten times fewer examples, and, furthermore, the test set has three NLEs per example. We choose not to evaluate the models w.r.t. WinoGrande, since the NLE test set contains only 100 examples, for which we already have a human evaluation (Table 3), which is the gold standard for NLE evaluation. We report the ComVE test perplexity, along with BLEU [Papineni et al., 2002], METEOR [Banerjee and Lavie, 2005], and BERTScore [Zhang et al., 2020b]. We select METEOR and BERTScore because they have been shown to have the best correlation with human judgment of NLEs across

Table 6: Performance of models with WG (upper half) and ComVE (lower half) as child tasks. The columns Yes, Weak Yes, Weak No, and No present the number (with percentage) of NLE validity scores given by the human annotators. Note that only correctly classified examples are included in these scores. The final column shows the inter-annotator agreement measured by Fleiss’ kappa.

WG Models	Acc @100	Yes	Weak Yes	Weak No	No	NLE score	Fleiss’ kappa
T5-Large: MT-FT(50)	68	57 ( <b>27.9%</b> )	56 ( <b>27.5%</b> )	24 (11.8%)	67 (32.8%)	<b>50.2</b>	0.09
MT-FT(0)	64	8 (4.2%)	15 (7.8%)	11 (5.7%)	158 (82.3%)	11.3	0.2
MT-FT(25)	64	36 (18.8%)	49 ( <b>25.5%</b> )	25 (13.0%)	82 (42.7%)	40.1	0.17
MT-FT(50)	63	50 ( <b>26.5%</b> )	33 (17.5%)	34 (18.0%)	72 (38.1%)	<b>44.1</b>	0.12
MT-FT(100)	63	40 (21.2%)	42 (22.2%)	27 (14.3%)	80 (42.3%)	40.7	0.06
ComVE Models	Acc @100	Yes	Weak Yes	Weak No	No	NLE score	Fleiss’ kappa
T5-Large: FT-FT(50)	87	76 (29.1%)	56 ( <b>21.5%</b> )	9 (3.4%)	120 (46.0%)	44.6	0.17
FT-FT(0)	83	27 (10.8%)	48 (19.3%)	18 (7.2%)	156 (62.7%)	26.1	0.01
FT-FT(25)	82	72 (29.3%)	53 ( <b>21.5%</b> )	10 (4.1%)	111 (45.1%)	45.0	0.12
FT-FT(50)	79	79 ( <b>33.3%</b> )	37 (15.6%)	17 (7.2%)	104 (43.9%)	46.1	0.17
FT-FT(100)	81	80 ( <b>32.9%</b> )	48 (19.8%)	8 (3.3%)	107 (44.0%)	47.2	0.31
FT-FT(200)	79	81 ( <b>34.2%</b> )	49 (20.7%)	10 (4.2%)	97 (40.9%)	<b>49.4</b>	0.27

several datasets [Kayser et al., 2021]. We also report on BLEU because it gives insights into the lower-level (n-gram) features of the NLEs.

In terms of test perplexity, as expected from the human evaluation, the MT-FT and FT-FT models outperform all other models in the comparison, except for the heuristic baseline for which there is no definition of perplexity. Although the MT-MT and MT-FT models are similar in terms of human evaluation, they differ significantly on test perplexity (8.02 vs. 5.11). This is not surprising if we factor in the similar comparison between FT-MT and FT-FT (8.18 vs. 5.13), which suggests that the test perplexity has more to do with the extent to which the model has been fit to the test distribution. In this case, the ComVE test perplexity results only let us conclude that fine-tuning on the 50 NLEs produces a better fit to the test distribution. This is confirmed by the results on BLEU, where fine-tuning on the 50 NLEs produces much more low-level features (B-1, B-2, B-3, and B-4) that match with the test dataset.

In terms of BLEU, METEOR, and BERTScore, FT-FT outperforms all three other main models. The best-performing baseline, T5b(ComVE)-ft(50), outperforms FT-FT in terms of BLEU score but is similar in terms of METEOR and BERTScore. This suggests that it produces NLEs that are closer to the test NLEs in terms of low-level features (unigram, bigram, trigram, and four-grams). This can be explained by the fact that many training NLEs resemble one of the two statements, because often the correct statement is a trivial NLE of itself. Upon manual inspection, we have noticed that the baseline T5b(ComVE)-ft(50) does produce more trivial NLEs (that repeat part of the input statements). The BERTScore and METEOR results on the full test dataset confirm that the FT-FT model performs significantly better than the heuristic baseline.

### 3.4 Method Scalability

After selecting the best training approaches from the previous experiments, we investigate the model performance over various explanation dataset sizes, and by training with a larger LM (T5-Large).

First, we investigate the performance of the best models as we increase the size of the NLE training dataset on the child task. For WG, we train the best model on up to 100 NLEs (as many as we have). For ComVE, we train the best model on up to 200 NLEs, since ComVE has a vastly larger training set with NLEs (10,000 examples). For WG, the results in Table 6 show that the performance improves up to 50 training NLEs, but the NLE score is worse with 100 NLEs than with 50 (40.7 vs 44.1). However, this drop of performance is not statistically significant with a p-value of 0.21. Nevertheless, this could suggest that for larger training datasets, the model is transitioning from relying on the inductive bias of e-SNLI to relying mostly on the information from the child task NLEs. For ComVE, the results in Table 6 show a clear trend of performance improvement from 0 to 200 training instances. This trend is also consistent across Yes and No scores, which suggests that the FT-FT model scales well with the training data. An investigation of high-resource settings falls beyond the scope of this work.



Second, we train the best models by using T5-Large instead of T5-Base, to verify if larger models can lead to better NLE transfer learning. For WG, Table 6 shows that the T5-Large model outperforms T5-Base in terms of NLE score (50.2 vs. 44.1), but it is not statistically significant with p-value of 0.06. The four categories show that while T5-Large is relatively close to T5-Base in terms of Yes, Weak No and No scores, it outperforms it significantly on the Weak Yes score (27.5% vs. 17.5%) with a p-value of 0.01. Furthermore, T5-Large obtains positive scores (Yes or Weak Yes) in most cases (55.4%), which proves that large models can obtain convincing explanations on this task. For ComVE, the results in Table 6 show that T5-Large underperforms T5-Base in terms of the NLE score (44.6 vs. 46.1), but it is not statistically significant, with a p-value of 0.38. As for WG, the two models have similar performance in terms of Yes, Weak No and No scores, but T5-Large significantly outperforms T5-Base in terms of Weak Yes score (21.5% vs 15.6%) with a p-value of 0.03. In conclusion, increasing the LM size from T5-Base to T5-Large does not lead to a significant improvement in the overall NLE quality (NLE score) for either tasks, but significantly improves the number of plausible NLEs (Weak Yes).

## 4 Related Work

There are three main focuses in NLE generation for natural language processing: perceived quality improvement Camburu et al. [2018], Narang et al. [2020], Valentino et al. [2020], NLE faithfulness Kumar and Talukdar [2020], Liu et al. [2019], Latcinnik and Berant [2020], and transfer of NLEs. In-domain transfer of NLEs, between datasets of the same task, has been done by Camburu et al. [2018], Kumar and Talukdar [2020], and Narang et al. [2020]. Narang et al. [2020] additionally consider zero-shot out-of-domain transfer of NLEs, while Erliksson et al. [2021] extends their work by showing that few-shot out-of-domain transfer of NLEs works when evaluated on automatic metrics.

In the more general area of natural language generation, few-shot learning is a growing topic Chen et al. [2020], especially in dialog generation Peng et al. [2020], Shalyminov et al. [2019]. These approaches, however, cannot be directly applied to transfer learning of NLEs, which is a dual task of predicting both the label and generating an explanation.

NLE evaluation can be done in one of two ways: automatic evaluation and human evaluation. The use of automatic metrics from the machine translation domain to the NLE evaluation domain has been investigated by Kayser et al. [2021]. They conclude that the automatic evaluation metrics only weakly correlate with human judgment of NLE quality, but that there are significantly better alternatives to the BLEU score [Papineni et al., 2002], such as METEOR [Banerjee and Lavie, 2005] and BERTScore [Zhang et al., 2020b].

The ERASER benchmark [DeYoung et al., 2020] includes a number of NLE generation tasks, along with metrics for automatic evaluation of NLEs. Apart from the standardization of NLE evaluation, the ERASER benchmark proposes metrics for measuring NLE faithfulness. Jacovi and Goldberg [2020] additionally propose a better definition of faithful NLEs and procedures for evaluation. Although faithfulness is important for NLE generation, it falls beyond the scope of the current work, which is perceived NLE quality.

For the task of resolving hard cases of pronoun resolution, there is the WinoWhy Zhang et al. [2020a] diagnostic dataset. It is based on the Winograd Schema Challenge dataset [Levesque et al., 2012], and is phrased as a zero-shot NLE classification task. We decided not to use it because we are interested in measuring NLE generation rather than classification of predefined NLEs.

## 5 Summary and Outlook

In this work, we introduced and compared four methods for few-shot out-of-domain transfer of NLEs, and showed that out-of-domain transfer learning can significantly help with NLE generation. Amongst the four proposed NLE transfer learning methods, we found that the most convincing NLEs are generated by the methods that provide separate training regimes for the child task and its few training NLEs. We have also concluded that the methods that obtain the best downstream accuracy are those for which the parent and child tasks are on a different training regime. Finally, we investigated how the best methods scale in terms of model size and NLE training data size. This method comparison provides a solid foundation for future research into methods for out-of-domain transfer of explanations.

## References

- S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W05-0909>.
- S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL <https://www.aclweb.org/anthology/D15-1075>.
- O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom. e-SNLI: Natural language inference with natural language explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc., 2018. URL <http://papers.nips.cc/paper/8163-e-snli-natural-language-inference-with-natural-language-explanations.pdf>.
- O.-M. Camburu, B. Shillingford, P. Minervini, T. Lukasiewicz, and P. Blunsom. Make up your mind! Adversarial generation of inconsistent natural language explanations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4157–4165, July 2020. doi: 10.18653/v1/2020.acl-main.382. URL <https://www.aclweb.org/anthology/2020.acl-main.382>.
- Z. Chen, H. Eavani, W. Chen, Y. Liu, and W. Y. Wang. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.18. URL <https://www.aclweb.org/anthology/2020.acl-main.18>.
- I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d’Alché Buc, editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6. doi: 10.1007/11736790\_9.
- J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL <https://www.aclweb.org/anthology/2020.acl-main.408>.
- K. F. Erlíksson, A. Arpteg, M. Matskin, and A. H. Payberah. Cross-domain transfer of generative explanations using text-to-text models. In E. Métais, F. Mezziane, H. Horacek, and E. Kapetanios, editors, *Natural Language Processing and Information Systems*, pages 76–89, Cham, 2021. Springer International Publishing. ISBN 978-3-030-80599-9.
- J. Fleiss et al. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- L. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9908 of *LNCS*, pages 3–19, 10 2016. ISBN 978-3-319-46492-3. doi: 10.1007/978-3-319-46493-0\_1.
- J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL <https://www.aclweb.org/anthology/P18-1031>.

- 411 A. Jacovi and Y. Goldberg. Towards faithfully interpretable NLP systems: How should we de-  
412 fine and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association*  
413 *for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Compu-  
414 tational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL [https://www.aclweb.org/](https://www.aclweb.org/anthology/2020.acl-main.386)  
415 [anthology/2020.acl-main.386](https://www.aclweb.org/anthology/2020.acl-main.386).
- 416 M. Kayser, O.-M. Camburu, L. Salewski, C. Emde, V. Do, Z. Akata, and T. Lukasiewicz. e-ViL: A  
417 dataset and benchmark for natural language explanations in vision-language tasks. *Computing*  
418 *Research Repository*, arXiv:2105.03761, 2021.
- 419 J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata. Textual explanations for self-driving  
420 vehicles. *Lecture Notes in Computer Science*, page 577–593, 2018. ISSN 1611-3349. doi:  
421 10.1007/978-3-030-01216-8\_35. URL [http://dx.doi.org/10.1007/978-3-030-01216-8\\_](http://dx.doi.org/10.1007/978-3-030-01216-8_35)  
422 35.
- 423 S. Kumar and P. Talukdar. NILE : Natural language inference with faithful natural language explana-  
424 tions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,  
425 pages 8730–8742, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/  
426 v1/2020.acl-main.771. URL <https://www.aclweb.org/anthology/2020.acl-main.771>.
- 427 J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*,  
428 33(1):159–174, 1977. ISSN 0006341X, 15410420. URL [http://www.jstor.org/stable/](http://www.jstor.org/stable/2529310)  
429 2529310.
- 430 V. Latcinnik and J. Berant. Explaining question answering models through text generation. *Computing*  
431 *Research Repository*, arXiv:2004.05569, 2020.
- 432 H. J. Levesque, E. Davis, and L. Morgenstern. The Winograd Schema Challenge. In *Proceedings*  
433 *of the 13th International Conference on Principles of Knowledge Representation and Reasoning*,  
434 pages 552–561. AAAI Press, 2012. ISBN 9781577355601. URL [https://dl.acm.org/doi/](https://dl.acm.org/doi/10.5555/3031843.3031909)  
435 10.5555/3031843.3031909.
- 436 W. Ling, D. Yogatama, C. Dyer, and P. Blunsom. Program induction by rationale generation:  
437 Learning to solve and explain algebraic word problems. In *Proceedings of the Annual Meeting of*  
438 *the Association for Computational Linguistics (ACL)*, pages 158–167, July 2017. doi: 10.18653/  
439 v1/P17-1015. URL <https://www.aclweb.org/anthology/P17-1015>.
- 440 H. Liu, Q. Yin, and W. Y. Wang. Towards explainable NLP: A generative explanation framework for  
441 text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational*  
442 *Linguistics*, pages 5570–5581, Florence, Italy, July 2019. Association for Computational Linguis-  
443 tics. doi: 10.18653/v1/P19-1560. URL <https://www.aclweb.org/anthology/P19-1560>.
- 444 T. Lombrozo. The structure and function of explanations. In *Trends in Cognitive Sciences*, volume 10,  
445 pages 464–70. Cell Press, 11 2006. doi: 10.1016/j.tics.2006.08.004.
- 446 T. Lombrozo. Explanation and abductive inference. In *Oxford Handbook of Thinking and Reasoning*,  
447 pages 260–276. Oxford University Press, 01 2012. doi: 10.1093/oxfordhb/9780199734689.013.  
448 0014.
- 449 I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th Inter-*  
450 *national Conference on Learning Representations*, New Orleans, LA, USA, 2019. OpenReview.net.  
451 URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- 452 B. P. Majumder, O.-M. Camburu, T. Lukasiewicz, and J. McAuley. Rationale-inspired natural  
453 language explanations with commonsense. *Computing Research Repository*, arXiv:2106.13876,  
454 2021.
- 455 S. Narang, C. Raffel, K. Lee, A. Roberts, N. Fiedel, and K. Malkan. WT5?! Training text-to-text  
456 models to explain their predictions. *Computing Research Repository*, arXiv:2004.14546, 2020.
- 457 K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of  
458 machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computa-*  
459 *tional Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for  
460 Computational Linguistics. doi: 10.3115/1073083.1073135. URL [https://www.aclweb.org/](https://www.aclweb.org/anthology/P02-1040)  
461 [anthology/P02-1040](https://www.aclweb.org/anthology/P02-1040).

- 462 D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach.  
463 Multimodal explanations: Justifying decisions and pointing to the evidence. In *2018 IEEE/CVF*  
464 *Conference on Computer Vision and Pattern Recognition*, pages 8779–8788, 2018. doi: 10.1109/  
465 CVPR.2018.00915.
- 466 B. Peng, C. Zhu, C. Li, X. Li, J. Li, M. Zeng, and J. Gao. Few-shot natural language gener-  
467 ation for task-oriented dialog. In *Findings of the Association for Computational Linguistics:*  
468 *EMNLP 2020*, pages 172–182, Online, Nov. 2020. Association for Computational Linguistics.  
469 doi: 10.18653/v1/2020.findings-emnlp.17. URL [https://www.aclweb.org/anthology/2020.](https://www.aclweb.org/anthology/2020.findings-emnlp.17)  
470 [findings-emnlp.17](https://www.aclweb.org/anthology/2020.findings-emnlp.17).
- 471 C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu.  
472 Exploring the limits of transfer learning with a unified text-to-text transformer. *Computing Research*  
473 *Repository*, arXiv:1910.10683, 2019.
- 474 N. F. Rajani, B. McCann, C. Xiong, and R. Socher. Explain yourself! leveraging language models  
475 for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association*  
476 *for Computational Linguistics*, pages 4932–4942, Florence, Italy, July 2019. Association for  
477 Computational Linguistics. doi: 10.18653/v1/P19-1487. URL [https://www.aclweb.org/](https://www.aclweb.org/anthology/P19-1487)  
478 [anthology/P19-1487](https://www.aclweb.org/anthology/P19-1487).
- 479 S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *5th International*  
480 *Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon,  
481 France, 2017. OpenReview.net. URL <https://openreview.net/forum?id=rJY0-Kc1l>.
- 482 K. Sakaguchi, R. Le Bras, C. Bhagavatula, and Y. Choi. WinoGrande: An adversarial winograd  
483 schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):  
484 8732–8740, Apr. 2020. doi: 10.1609/aaai.v34i05.6399. URL [https://ojs.aaai.org/index.](https://ojs.aaai.org/index.php/AAAI/article/view/6399)  
485 [php/AAAI/article/view/6399](https://ojs.aaai.org/index.php/AAAI/article/view/6399).
- 486 I. Shalyminov, S. Lee, A. Eshghi, and O. Lemon. Few-shot dialogue generation without annotated data:  
487 A transfer learning approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and*  
488 *Dialogue*, pages 32–39, Stockholm, Sweden, Sept. 2019. Association for Computational Linguistics.  
489 doi: 10.18653/v1/W19-5904. URL <https://www.aclweb.org/anthology/W19-5904>.
- 490 S. Thrun. Is learning the n-th thing any easier than learning the first? In *Ad-*  
491 *vances in Neural Information Processing Systems*, volume 8, pages 640–646.  
492 MIT Press, 1996. URL [https://proceedings.neurips.cc/paper/1995/file/](https://proceedings.neurips.cc/paper/1995/file/bdb106a0560c4e46ccc488ef010af787-Paper.pdf)  
493 [bdb106a0560c4e46ccc488ef010af787-Paper.pdf](https://proceedings.neurips.cc/paper/1995/file/bdb106a0560c4e46ccc488ef010af787-Paper.pdf).
- 494 M. Valentino, M. Thayaparan, and A. Freitas. Explainable natural language reasoning via conceptual  
495 unification. *Computing Research Repository*, arXiv:2009.14539, 2020.
- 496 C. Wang, S. Liang, Y. Jin, Y. Wang, X. Zhu, and Y. Zhang. SemEval-2020 task 4: Commonsense vali-  
497 dation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages  
498 307–321, Barcelona (online), Dec. 2020. International Committee for Computational Linguistics.  
499 URL <https://www.aclweb.org/anthology/2020.semeval-1.39>.
- 500 K. K. Yuen and W. J. Dixon. The approximate behaviour and performance of the two-sample trimmed  
501 t. *Biometrika*, 60(2):369–374, 1973. ISSN 00063444. URL [http://www.jstor.org/stable/](http://www.jstor.org/stable/2334550)  
502 [2334550](http://www.jstor.org/stable/2334550).
- 503 H. Zhang, X. Zhao, and Y. Song. WinoWhy: A deep diagnosis of essential commonsense knowledge  
504 for answering Winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the*  
505 *Association for Computational Linguistics*, pages 5736–5745, Online, July 2020a. Association for  
506 Computational Linguistics. doi: 10.18653/v1/2020.acl-main.508. URL [https://www.aclweb.](https://www.aclweb.org/anthology/2020.acl-main.508)  
507 [org/anthology/2020.acl-main.508](https://www.aclweb.org/anthology/2020.acl-main.508).
- 508 T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: Evaluating text generation  
509 with BERT. In *Proceedings of the 8th International Conference on Learning Representations*,  
510 Addis Ababa, Ethiopia, 2020b. OpenReview.net. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=SkeHuCVFDr)  
511 [SkeHuCVFDr](https://openreview.net/forum?id=SkeHuCVFDr).

## Checklist

### 1. For all authors...

- (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 5.
- (b) Did you describe the limitations of your work? [Yes] See Section 3.
- (c) Did you discuss any potential negative societal impacts of your work? [N/A] the paper has only positive societal impacts
- (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] the paper is ethically sound

### 2. If you are including theoretical results...

- (a) Did you state the full set of assumptions of all theoretical results? [N/A] no theoretical results
- (b) Did you include complete proofs of all theoretical results? [N/A] no theoretical results

### 3. If you ran experiments...

- (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] the code will be made public on acceptance
- (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] in the appendix
- (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] only one seed was used
- (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] we used miscellaneous hardware

### 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

- (a) If your work uses existing assets, did you cite the creators? [Yes] all datasets were cited
- (b) Did you mention the license of the assets? [N/A] all datasets are public
- (c) Did you include any new assets either in the supplemental material or as a URL? [No] the new data will be made public on acceptance
- (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] the data is publicly available
- (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] it contains no such information

### 5. If you used crowdsourcing or conducted research with human subjects...

- (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] we have shared screenshots in the appendix
- (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] there are no risks
- (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] but only the price per 10 NLE pairs.

Table 7: Best hyperparameters for all trained models, along with the corresponding criterion used for model selection, and the best dev result value w.r.t. that criterion. \*–subject to the dev accuracy being large enough (>75%).

Models	Num epochs	Learning rate	Criterion	Best value
T5b(eSNLI)	3	3e-4	e-SNLI dev NLE ppl	2.192
T5–Large(e-SNLI)	–	–	same as T5b	–
T5b(e-SNLI, SNLI)	3	3e-4	e-SNLI dev NLE ppl	2.199
<b>WG Models</b>				
T5b(eSNLI, WG)	5	1e-4	WG-dev acc	83.2%
T5–Large(e-SNLI, WG)	–	–	same as T5b	–
T5b(eSNLI)–ft(WG)	7	3e-4	WG-dev acc	81.0%
T5b(WG)	5	1e-4	WG-dev acc	85.1%
T5b(WG)–ft(50)	21	3e-4	WG dev NLE ppl	4.665
T5b(WG, 50)	5	1e-4	WG dev NLE ppl	4.945
WT5–FT	11	3e-4	WG-dev acc	80.8%
WT5–MT	5	1e-4	WG-dev acc	83.4%
MT–MT(50)	3	3e-5	WG dev NLE ppl	4.815
FT–MT(WG, 50)	5	1e-4	WG dev NLE ppl	5.419
MT–FT(50)	10	3e-4	WG dev NLE ppl	4.401
FT–FT(50)	17	3e-4	WG dev NLE ppl	5.022
T5–Large: MT–FT(50)	7	3e-4	WG dev NLE ppl	3.974
MT–FT(25)	10	3e-4	WG dev NLE ppl	4.684
MT–FT(100)	7	3e-4	WG dev NLE ppl	4.212
<b>ComVE Models</b>				
T5b(e-SNLI, ComVE)	3	3e-4	ComVE dev acc	82.8%
T5b(e-SNLI)–ft(ComVE)	7	3e-4	ComVE dev acc	86.8%
T5–Large(e-SNLI)–ft(ComVE)	7	3e-4	ComVE dev acc	89.4%
T5b(ComVE)	5	3e-4	ComVE dev acc	88.4%
T5b(ComVE)–ft(50)	13	3e-4	ComVE dev NLE ppl	5.170
T5b(ComVE, 50)	5	1e-4	ComVE dev NLE ppl*	9.294
WT5–FT	10	3e-4	ComVE dev acc	87.0%
WT5–MT	5	1e-4	ComVE dev acc	84.4%
MT–MT(50)	5	1e-4	ComVE dev NLE ppl	7.886
FT–MT(50)	1	1e-3	ComVE dev NLE ppl	7.970
MT–FT(50)	5	1e-3	ComVE dev NLE ppl	4.958
FT–FT(50)	5	1e-3	ComVE dev NLE ppl	5.002
T5–Large: FT–FT(50)	7	1e-3	ComVE dev NLE ppl	4.654
FT–FT(25)	7	1e-3	ComVE dev NLE ppl	5.274
FT–FT(100)	3	1e-3	ComVE dev NLE ppl	4.865
FT–FT(200)	3	1e-3	ComVE dev NLE ppl	4.688

## A Hyperparameters

For all models, we fix the batch size to 16 and do a grid search over the learning rate values and the number of training epochs. For all WG models, we search over the learning rate values of 3e-4, 1e-4, and 3e-5, whereas for ComVE we search over 1e-3, 3e-4, 1e-4, and 3e-5. For e-SNLI, we train on 1, 2, 3, and 5 epochs. For WG, we train on 1, 2, 3, 5, 7, 9, and 11 epochs, and for ComVE, we train on 1, 2, 3, 5, 7, 10, and 13 epochs. When few-shot fine-tuning with NLEs, we train on 1, 2, 3, 5, 7, 10, 13, 17, 21, and 26 epochs. Multi-task learning always uses the hyperparameter range of the larger dataset.

The best hyperparameters (via grid search) for each model training stage are presented in Table 7. The table also contains the hyperparameter selection criterion and the dev results w.r.t. that criterion. Note that the WG-dev accuracy in Table 7 is much higher than the corresponding WG-test accuracy in Table 3 because WG-dev is sampled from the training dataset of WG, whereas WG-test is the original WG development set, which is filtered to increase its difficulty Sakaguchi et al. [2020].

564 **B Data Collection Forms**

565 Below are full-page screenshots of the data collection forms that were used for WinoGrande (Figure 1)  
566 and ComVE (Figure 2).

567 **C Examples of Model-Generated NLEs**

568 In the tables below are the answers and explanations for each task (WG and ComVE) and for all 8  
569 compared models on the first 6 examples (out of the 100) that were evaluated.

Figure 1: WinoGrande data collection template. There are two explanations per task.

Instructions

### Overview

Thank you for participating in this HIT

This HIT contains 10 **independent** tasks.

### Task Description

1. First, you will be shown a sentence with a gap denoted by an underscore (\_).
2. You will then be provided with **two** options to fill the gap "\_" in the sentence, and you will have to choose the correct one.
3. You will then be shown two explanations that each, separately, tries to justify this answer. **Note that the explanations are independent of each other and their order is meaningless!**
4. For each of the explanations, we ask **two evaluation questions**:
  - Given the statement, is this a **valid and satisfactory** explanation to justify the selected option for filling the gap?
  - If any, what are the shortcomings of the explanation?

### Tips

- Minor grammatical and style errors should be ignored (e.g. case sensitivity, missing periods, a missing pronoun etc.).
- A valid and satisfactory explanation should be logical, sufficient, and should not contain irrelevant arguments.
- An explanation that just repeats or restates the statement is NOT a valid explanation.
- A good approach to evaluating explanations is the following: Before looking at the explanations, think of an explanation yourself and then anchor your assessments based on that.

**Quality checks and known answers are placed throughout the questionnaire!**

Examples (click to expand/collapse)

Questionnaire

----- TASK 1 -----

**Fill the gap:** Lawrence planned to steal the valuable painting from Michael, because \_ wanted to own something beautiful.

Options: ☐ Lawrence ☐ Michael

**Explanation #1:** A valuable painting is a thing of beauty. Lawrence wants to steal the valuable painting from Michael, so Lawrence wants to own this thing of beauty.

a) Given the above schema, is this a valid and satisfactory explanation to justify the selected option?

☐ Yes ☐ Weak Yes ☐ Weak No ☐ No

b) What are the shortcomings of the explanation?

☐ Does **not** make sense ☐ Insufficient justification ☐ Irrelevant to the task ☐ Too trivial ☐ None

**Explanation #2:** Lawrence wanted something beautiful, so he planned to steal the painting.



Figure 2: ComVE data collection template. There are two explanations per task.

Instructions

### Overview

Thank you for participating in this HIT

This HIT contains 10 **independent** tasks.

### Task Description

1. First, you will be shown two statements in random order. One of them makes sense, and the other does not.
2. You have to choose which of the two statements does **not** make sense.
3. You will then be shown two explanations that each try to justify this answer. **Note that the explanations are independent of each other and their order is meaningless!**
4. For each of the explanations, we ask **two evaluation questions**:
  - Given the selected statement, is this a **valid and satisfactory** explanation of why this statement does not make sense?
  - If any, what are the shortcomings of the explanation?

### Tips

- Minor grammatical and style errors should be ignored (e.g. case sensitivity, missing periods, a missing pronoun etc.).
- A valid and satisfactory explanation should be logical, sufficient, and should not contain irrelevant arguments.
- An explanation that just repeats or restates the statements is NOT a valid explanation.
- A good approach to evaluating explanations is the following: Before looking at the explanations, think of an explanation yourself and then anchor your assessments based on that.

**Quality checks and known answers are placed throughout the questionnaire!**

Examples (click to expand/collapse)

Questionnaire

----- TASK 1 -----

**Select the statement that does not make sense:**

**Statement 1:** He moved a city to his belongings.

**Statement 2:** He moved his belongings to a new city.

Options: ☒ Statement 1 ☐ Statement 2

**Explanation #1:** A city is too big to fit into whatever belongings the person has.

a) Given the above statements, is this a valid and satisfactory explanation of the selected option?

☒ Yes ☐ Weak Yes ☐ Weak No ☐ No

b) What are the shortcomings of the explanation?

☐ Does **not** make sense ☐ Insufficient justification ☐ Irrelevant to the task ☐ Too trivial ☒ None

**Explanation #2:** There are plenty of options of places to go in a city.

<b>Schema:</b> The ___ car was a pleasure to drive but the red car was not as the blue car was very expensive. <b>Options:</b> blue, red. <b>Ground-truth:</b> blue.	
<b>Model</b>	<b>Output</b>
T5b(WG)–ft(50)	<b>Prediction:</b> red. <b>NLE:</b> The blue car is expensive and the blue car is more expensive.
T5b(WG, 50)	<b>Prediction:</b> red. <b>NLE:</b> The blue car is more expensive than the red car, hence it is more expensive to drive.
WT5-FT	<b>Prediction:</b> red. <b>NLE:</b> The car is described in more detail than in the second sentence. Blue
WT5-MT	<b>Prediction:</b> blue. <b>NLE:</b> Red is a different color than blue.
MT-MT(50)	<b>Prediction:</b> red. <b>NLE:</b> The car was a pleasure to drive but the blue car was not as the blue car was very expensive.
MT-FT(50)	<b>Prediction:</b> red. <b>NLE:</b> The red car is more expensive than the blue car.
FT-MT(WG, 50)	<b>Prediction:</b> blue. <b>NLE:</b> The blue car is a pleasure to drive and the red car is very expensive.
FT-FT(50)	<b>Prediction:</b> blue. <b>NLE:</b> The car is a pleasure to drive but the red car is red.

<b>Schema:</b> Logan ended the one-side relationship they were having with Robert, which made ___ feel relieved. <b>Options:</b> Logan, Robert. <b>Ground-truth:</b> Logan.	
<b>Model</b>	<b>Output</b>
T5b(WG)–ft(50)	<b>Prediction:</b> Robert. <b>NLE:</b> Logan ended the one-side relationship they were having with Robert, which made Logan feel relieved.
T5b(WG, 50)	<b>Prediction:</b> Logan. <b>NLE:</b> Robert has a one-sided relationship with Logan, which makes him feel relieved.
WT5-FT	<b>Prediction:</b> Logan. <b>NLE:</b> The Logan
WT5-MT	<b>Prediction:</b> Logan. <b>NLE:</b> Robert
MT-MT(50)	<b>Prediction:</b> Robert. <b>NLE:</b> Logan ended the one-side relationship they were having with Robert, which made Logan feel relieved
MT-FT(50)	<b>Prediction:</b> Logan. <b>NLE:</b> The fact that Logan ended the one-sided relationship with Robert makes him feel relieved.
FT-MT(WG, 50)	<b>Prediction:</b> Robert. <b>NLE:</b> Logan’s feelings are relieved because he ended the relationship with Robert.
FT-FT(50)	<b>Prediction:</b> Logan. <b>NLE:</b> Logan ending a one-side relationship makes Logan feel relieved.

<b>Schema:</b> Fiction books were interesting and easy to read for Logan but not Brett because ___ enjoyed real tales. <b>Options:</b> Logan, Brett. <b>Ground-truth:</b> Brett.	
<b>Model</b>	<b>Output</b>
T5b(WG)–ft(50)	<b>Prediction:</b> Brett. <b>NLE:</b> Fiction books were interesting and easy to read for Logan, but Brett enjoyed real tales more than fiction books.
T5b(WG, 50)	<b>Prediction:</b> Brett. <b>NLE:</b> Fiction books were interesting and easy to read for Logan, while real tales were more difficult to read for Brett.
WT5-FT	<b>Prediction:</b> Logan. <b>NLE:</b> There is nothing to indicate that he is enjoying real stories or that he has been referred to as ‘Boy’
WT5-MT	<b>Prediction:</b> Logan. <b>NLE:</b> The Logan may not be Logan. Brett may not be Brett. Logan may not be Logan. Brett may not be Brett. Logan may not be Logan. Brett may not be Brett. Loga [...]
MT-MT(50)	<b>Prediction:</b> Brett. <b>NLE:</b> Fiction books were interesting and easy to read for Logan but not Brett because Brett enjoyed real tales
MT-FT(50)	<b>Prediction:</b> Brett. <b>NLE:</b> Fiction books are not real tales.
FT-MT(WG, 50)	<b>Prediction:</b> Logan. <b>NLE:</b> Fiction books are interesting and easy to read for Logan, so Logan is more likely to enjoy real tales than Brett.
FT-FT(50)	<b>Prediction:</b> Logan. <b>NLE:</b> Fiction books are not real tales.



<b>Statement 1:</b> He inserts his thumb into the car ignition switch. <b>Statement 2:</b> He inserts the keys into the car ignition switch. <b>Nonsensical statement:</b> Statement 1.	
<b>Model</b>	<b>Output</b>
T5b(ComVE)–ft(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> A thumb is not inserted into the car ignition switch.
T5b(ComVE, 50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> It's not a car ignition switch, it's a car ignition switch, it's a car ignition switch, it's a car ignition switch, it's a car ignition switch, it's a car ignition switch, [...]
WT5-FT	<b>Prediction:</b> Statement 1. <b>NLE:</b> SIDEWALK INSTALLS THE KEYS INTO THE CAR INFUSION SHUT.
WT5-MT	<b>Prediction:</b> Statement 1. <b>NLE:</b> the keys are not the same as the thumb
MT-MT(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> One cannot insert his thumb into the car ignition switch.
MT-FT(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> One cannot insert his thumb into the car ignition switch.
FT-MT(ComVE, 50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> You can't use your thumb to insert keys into the car ignition switch.
FT-FT(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> A thumb is not used to insert keys into a car ignition switch

<b>Statement 1:</b> She was met with an accident and was rushed to the hospital <b>Statement 2:</b> She was met with an accident and was rushed to the library <b>Nonsensical statement:</b> Statement 2.	
<b>Model</b>	<b>Output</b>
T5b(ComVE)–ft(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The library is closed to the public
T5b(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> She was met with an accident and was rushed to the library
WT5-FT	<b>Prediction:</b> Statement 2. <b>NLE:</b> She was met with an accident and was rushed to the hospital 1 explanation:
WT5-MT	<b>Prediction:</b> Statement 2. <b>NLE:</b> .
MT-MT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The library is not the only place where a person can have an accident.
MT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> A library is not a hospital.
FT-MT(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> A library is not a hospital.
FT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> A library is not a hospital.

<b>Statement 1:</b> She came off sailing across the pacific ocean <b>Statement 2:</b> She came off walking across the pacific ocean <b>Nonsensical statement:</b> Statement 2.	
<b>Model</b>	<b>Output</b>
T5b(ComVE)–ft(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> She came off sailing across the pacific ocean
T5b(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> She came off sailing across the pacific ocean Sailing across the pacific ocean Sailing across the pacific ocean Sailing across the pacific ocean Sailing across the pacific ocean Sailing across the pac [...]
WT5-FT	<b>Prediction:</b> Statement 2. <b>NLE:</b> She came off sailing across the pacific ocean Sentence 1:
WT5-MT	<b>Prediction:</b> Statement 2. <b>NLE:</b> .
MT-MT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> A person can't be sailing and walking at the same time.
MT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> sailing is not the same as walking
FT-MT(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> Sailing and walking are two different things.
FT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> sailing is not the same as walking

<b>Statement 1:</b> Leena daily eats two bananas which are good for health. <b>Statement 2:</b> Leena eats almost 1000 bananas daily as his morning breakfast. <b>Nonsensical statement:</b> Statement 2.	
<b>Model</b>	<b>Output</b>
T5b(ComVE)–ft(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> Leena eats two bananas a day which are good for health.
T5b(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> Bananas are good for health and you can't eat them in the morning.
WT5-FT	<b>Prediction:</b> Statement 2. <b>NLE:</b> Leena daily eats two bananas which are good for health 1 explanation:
WT5-MT	<b>Prediction:</b> Statement 2. <b>NLE:</b> .
MT-MT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> Leena daily eats two bananas which are good for health. is same as Leena eats almost 1000 bananas daily as his morning breakfast
MT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The bananas are good for health.
FT-MT(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The bananas are not good for health, they are good for health.
FT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The bananas are good for health.

<b>Statement 1:</b> The fire will burn you if you touch it. <b>Statement 2:</b> The fire will cool you if you touch it. <b>Nonsensical statement:</b> Statement 2.	
<b>Model</b>	<b>Output</b>
T5b(ComVE)–ft(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The fire will burn you if you touch it.
T5b(ComVE, 50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The fire will burn you if you touch it and will cool you if you touch it
WT5-FT	<b>Prediction:</b> Statement 2. <b>NLE:</b> The fire will burn you if you touch it.
WT5-MT	<b>Prediction:</b> Statement 2. <b>NLE:</b> the fire will burn you if you touch it.
MT-MT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The fire will burn you is not the same as the fire that will cool you.
MT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The fire will burn you if you touch it
FT-MT(ComVE, 50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Fire can't burn you
FT-FT(50)	<b>Prediction:</b> Statement 2. <b>NLE:</b> The fire will burn you.

<b>Statement 1:</b> Salsa is for hair <b>Statement 2:</b> Brushes are for hair <b>Nonsensical statement:</b> Statement 1.	
<b>Model</b>	<b>Output</b>
T5b(ComVE)–ft(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Salsa is for hair
T5b(ComVE, 50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Salsa is for hair
WT5-FT	<b>Prediction:</b> Statement 1. <b>NLE:</b> SENTENCE 1:
WT5-MT	<b>Prediction:</b> Statement 1. <b>NLE:</b> :
MT-MT(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Brushes are for hair is not the same as a Salsa
MT-FT(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Salsa is for hair
FT-MT(ComVE, 50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Brushes are for hair is a rephrasing of brushes are for hair.
FT-FT(50)	<b>Prediction:</b> Statement 1. <b>NLE:</b> Salsa is for hair