CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning

Bill Yuchen Lin[♥] **Wangchunshu Zhou**[♥] **Ming Shen**[♥] **Pei Zhou**[♥]
Chandra Bhagavatula[♠] **Yejin Choi**[♦♠] **Xiang Ren**[♥]
[♥]University of Southern California [♠]Allen Institute for Artificial Intelligence
[♦]Paul G. Allen School of Computer Science & Engineering, University of Washington
{yuchen.lin,xiangren}@usc.edu,{chandrab,yejinc}@allenai.org

Abstract

Recently, large-scale pretrained language models have demonstrated impressive performance on several commonsense-reasoning benchmark datasets. However, building machines with commonsense to compose realistically plausible sentences remains challenging. In this paper, we present a constrained text generation task, CommonGen associated with a benchmark dataset, to explicitly test machines for the ability of *generative commonsense reasoning*. Given a set of common concepts (e.g., {dog, frisbee, catch, throw}); the task is to generate a coherent sentence describing an everyday scenario using these concepts (e.g., "a man throws a frisbee and his dog catches it").

The CommonGen task is challenging because it inherently requires 1) *relational reasoning* with background commonsense knowledge, and 2) *compositional generalization* ability to work on unseen concept combinations. Our dataset, constructed through a combination of crowdsourced and existing caption corpora, consists of 77k commonsense descriptions over 35k unique concept-sets. Experiments show that there is a large gap between state-of-the-art text generation models (e.g., T5) and human performance (31.6% v.s. 63.5% in SPICE metric). Furthermore, we demonstrate that the learned generative commonsense reasoning capability can be transferred to improve downstream tasks such as CommonsenseQA (76.9% to 78.4 in dev accuracy) by generating additional context.

1 Introduction

Commonsense reasoning, the ability to make acceptable and logical assumptions about ordinary scenes in our daily life, has long been acknowledged as a critical bottleneck of artificial intelligence and natural language processing (Davis and Marcus, 2015). Most recent commonsense reasoning challenges, such as CommonsenseQA (Talmor et al., 2019), SocialIQA (Sap et al., 2019b), WinoGrande (Sakaguchi et al., 2019) and HellaSwag (Zellers et al., 2019b), have been framed as *discriminative* tasks – i.e. AI systems are required to *choose* the correct option from a set of choices based on a given context. While significant progress has been made on these discriminative tasks, we argue that commonsense reasoning in text generation poses a distinct complementary challenge. In this paper, we advance machine commonsense towards *generative* reasoning ability.

Figure 1: **An example of the dataset of CommonGen.** GPT-2, UniLM, BART and T5 are large pre-trained text generation models, *fine-tuned* on the proposed task.

Humans acquire the ability to compose sentences by learning to understand and use common concepts that they recognize in their surrounding environment (Tincoff and Jusczyk, 1999). The acquisition of such an ability is regarded as a significant milestone of human development (Moore, 2013). *Can machines acquire such generative commonsense reasoning ability?* To initiate the investigation, we present CommonGen[1] – a novel constrained generation task that requires machines to generate a sentence describing a day-to-day scene using concepts from a given *concept-set*. For ex-

[1] http://inklab.usc.edu/CommonGen/.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1823–1840

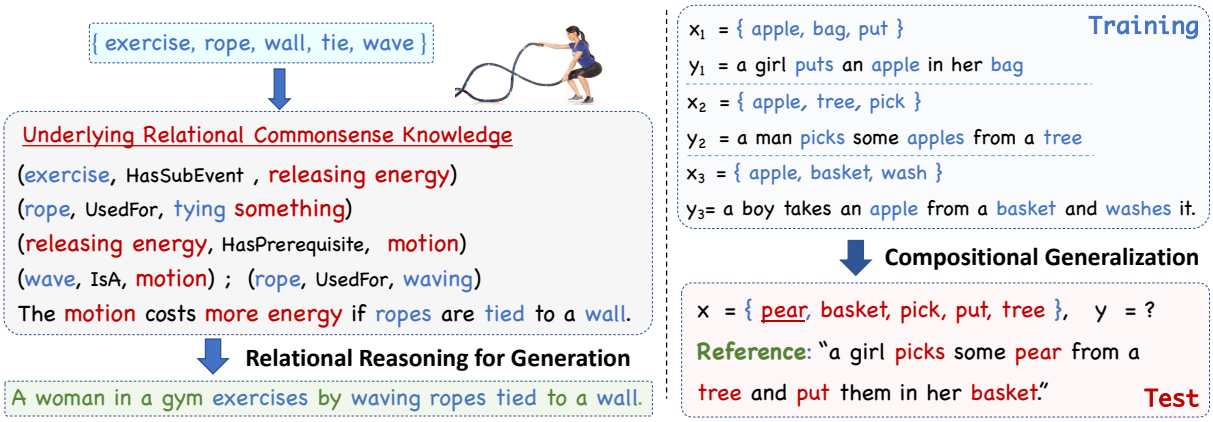

Figure 2: Two **key challenges** of COMMONGEN: *relational reasoning* with underlying commonsense knowledge about given concepts (left), and *compositional generalization* for unseen combinations of concepts (right).

ample, in Figure 1, given a set of concepts: {*dog*, *frisbee*, *catch*, *throw*}, machines are required to generate a sentence such as "a man *throws* a frisbee and his *dog catches* it in the air."

To successfully solve the task, models need to incorporate two key capabilities: a) *relational reasoning*, and b) *compositional generalization*. Grammatically sound sentences may not always be realistic as they might violate our commonsense (e.g., "*a dog throws a frisbee ...*"). In order to compose a plausible sentence that describes an everyday scenario, models need to construct a grammatical sentence while adhering to and reasoning over the commonsense relations between the given concepts. Models additionally need *compositional generalization* ability to infer about unseen concept compounds. This encourages models to reason about a potentially infinite number of novel combinations of familiar concepts – an ability believed to be a limitation of current AI systems (Lake and Baroni, 2017; Keysers et al., 2020).

Therefore, in support of the COMMONGEN task, we present a dataset consisting of 35,141 concept-sets associated with 77,449 sentences. We explicitly design our dataset collection process to capture the key challenges of relational reasoning and compositional generalization described above, through an actively controlled crowd-sourcing process. We establish comprehensive baseline performance for state-of-the-art language generation models with both extensive automatic evaluation and manual comparisons. The best model, based on T5 (Raffel et al., 2019), achieves 31.60% with significant gap compared to human performance of 63.50% in the SPICE metric – demonstrating the difficulty of the task. Our analysis shows that state-of-the-art models struggle at the task, generating implausible sentences – e.g. "dog throws a frisbee ..." , "giving massage to a table", etc. Additionally, we show that successful COMMONGEN models can benefit downstream tasks (e.g., commonsense-centric question answering) via generating useful context as background scenarios. We believe these findings point to interesting future research directions for the community of commonsense reasoning.

2 Task Formulation and Key Challenges

We formulate the proposed COMMONGEN task with mathematical notations and discuss its inherent challenges with concrete examples. The input is an unordered set of k concepts $x = \{c_1, c_2, \ldots, c_k\} \in \mathcal{X}$ (i.e. a concept-set), where each concept $c_i \in \mathcal{C}$ is a common object (noun) or action (verb). We use $\mathcal{X}$ to denote the space of all possible concept-sets and use $\mathcal{C}$ to denote the concept vocabulary (a subset of ConceptNet's unigram concepts). The expected output is a simple, grammatical sentence $y \in \mathcal{Y}$ that describes a common scenario in our daily life, using all given concepts in x (morphological inflections are allowed). A scenario can depict either a static situation or a short series of actions. The COMMONGEN task is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, which maps a concept-set x to a sentence y. The unique challenges of this task come from two aspects:

Relational Reasoning with Commonsense. Expected generative reasoners should prioritize the most plausible scenarios over many other less realistic ones. As shown in Figure 2, models need to recall necessary relational commonsense facts that are relevant to the given concepts, and then reason an optimal composition of them for gener-

ating a desired sentence. In order to complete a scenario, generative commonsense reasoners also need to reasonably associate additional concepts (e.g., 'woman', 'gym') as agents or background environments for completing a coherent scenario.

This not only requires understanding underlying commonsense relations between concepts, but also incrementally composing them towards a globally optimal scenario. The underlying reasoning chains are inherently based on a variety of background knowledge such as spatial relations, object properties, physical rules, temporal event knowledge, social conventions, etc. However, they may not be recorded in any existing knowledge bases.

Compositional Generalization. Humans can compose a sentence to describe a scenario about the concepts they may never seen them co-occurring. For example, in Figure 2, there is a testing concept-set $\hat{x}$ ={*pear*, *basket*, *pick*, *put*, *tree*}. The concept 'pear' never appear in the training data, and 'pick' never co-occurs with 'basket'. We, humans, can generalize from these seen scenarios in the training data and infer that a plausible output: $\hat{y}$ ="*a girl picks some pears from a tree and put them into her basket.*" This compositionally generalization ability via analogy, i.e., to make "infinite use of finite means" (Chomsky, 1965), is challenging for machines. This analogical challenge not only requires inference about similar concepts (e.g., 'apple' → 'pear') but also their latent associations.

3 Dataset Construction and Analysis

Figure 3 illustrates the overall workflow of our data construction for the proposed CommonGen task. We utilize several existing caption corpora for sampling frequent concept-sets (Sec. 3.1) for reflecting common scenarios. We employ AMT crowd workers for collecting human-written sentences (Sec. 3.2) for the development and test set, while we carefully monitor the quality of crowd workers and refine them dynamically. Finally, we present the statistics of the CommonGen dataset, and the analysis on the challenges (Sec. 3.4).

3.1 Collecting Concept-Sets from Captions

It can be unreasonable to present any *arbitrary* set of concepts (e.g., x ={*apple, fold, rope*}) and ask a reasoner to generate a commonsense scenario, since such an arbitrary set of concepts can be too unrelated. Therefore, our concept-sets are supposed to reflect reasonable concept co-occurrences

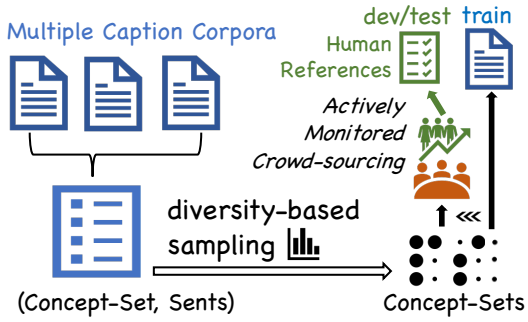

Figure 3: **Dataset construction workflow overview.**

in everyday situations. As web images and video clips capture diverse everyday scenarios, we use their caption text as a natural resource for collecting concept-sets and their corresponding descriptions of commonsense scenarios. More specifically, we collect visually-grounded sentences from several existing caption datasets, including image captioning datasets, such as Flickr30k (Young et al., 2014), MSCOCO (Lin et al., 2014), Conceptual Captions (Sharma et al., 2018), as well as video captioning datasets including LSMDC (Rohrbach et al., 2017), ActivityNet (Krishna et al., 2017), and VATEX (Wang et al., 2019b).

We first conduct part-of-speech tagging over all sentences in the corpora such that words in sentences can be matched to the concept vocabulary of ConceptNet. Then, we compute the sentence frequency of concept-sets consisting of 3~5 concepts. That is, for each combination of three/four/five concepts in the vocabulary, we know how many sentences are in the corpora covering all concepts.

Ideally, we want the selected concept-sets in our dataset to reflect the natural distribution of concept-sets in the real world. At first glance, a reasonable solution may seem to sample from the distribution of the concept-sets based on their frequencies in the source datasets. However, we find that this method leads to a rather unnaturally skewed collection of concept-sets, due to the inherent data biases from the source datasets. We therefore design a function to score a concept-set x based on *scene diversity* and *inverse frequency penalty*. We denote $S(x)$ as the set of unique sentences that contain all given concepts $\{c_1, c_2, \ldots, c_k\}$, and then we have

$$\texttt{score}(x) = |S(x)| \frac{|\bigcup_{s_i \in S(x)} \{w | w \in s_i\}|}{\sum_{s_i \in S(x)} \text{len}(s_i)} \rho(x),$$

where $\rho(x) = \frac{|\mathcal{X}|}{\max_{c_i \in x} |\{x' \mid c_i \in x' \text{ and } x' \in \mathcal{X}\}|}$. The first term in `score` is the number of unique sen-

Statistics	Train	Dev	Test
# Concept-Sets	**32,651**	**993**	**1,497**
-Size = 3	25,020	493	-
-Size = 4	4,240	250	747
-Size = 5	3,391	250	750
# Sentences	67,389	4,018	6,042
per Concept-Set	2.06	4.04	4.04
Average Length	10.54	11.55	13.34
# Unique Concepts	4,697	766	1,248
# Unique Concept-Pairs	59,125	3,926	8,777
# Unique Concept-Triples	50,713	3,766	9,920
% Unseen Concepts	-	6.53%	8.97%
% Unseen Concept-Pairs	-	96.31%	100.00%
% Unseen Concept-Triples	-	99.60%	100.00%

Table 1: The **basic statistics** of the COMMONGEN data. We highlight the ratios of concept compositions that are unseen in training data, which assures the challenge in compositional generalization ability.

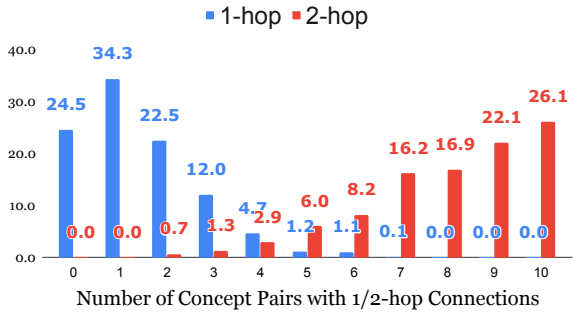

Figure 4: **Connectivity analysis** in 5-size concept-sets in the test set, each of which consists of 10 concept pairs. For example, 12.0 in blue means: there are 12% concept-sets that have 3 concept pairs with one-hop connections on ConceptNet.

tences covering all given concepts in x, and the second term is to represent the diversity of the scenes described in these sentences. Th last term $\rho(x)$ is the penalty of inverse frequency. Specifically, we find the concept in x that has the maximum "set frequency" (i.e., the number of unique concept-sets containing a particular concept), then we take the inverse with the number of all concept-sets for normalization. This penalty based on inverse set-frequency effectively controls the bias towards highly frequent concepts. With the distribution of such scores of concept-sets, we sample our candidate examples for the next steps.

3.2 Crowd-Sourcing References via AMT

In order to ensure the best quality, the references of the evaluation examples are crowdsourced from crowd workers on *Amazon Mechanical Turk*, which amounts to **10,060** references over 2.5k distinct concept-sets. Note that these newly collected references for dev and test examples can ensure that we can do a fair comparisons targeting generalization, considering potential data-leak (i.e., recent pre-trained language models might have seen the caption datasets). Each concept-set was assigned to at least 3 workers. In addition to references about given concept-sets, we also ask the workers to provide rationale sentences to explain what commonsense facts they have used, for ensuring that the described scenarios are common in daily life (example rationales are shown in Fig 9).

We control the quality by actively filtering workers who produced low-quality references, then removing their annotations, and finally re-opening the slots only for quality workers. There were 1,492 accepted workers in total and 171 disqualified workers in the end after the active filtering. There are three criteria for efficiently narrowing down candidates for us to further manually remove out low-quality workers: 1) coverage via part-of-speech tagging, 2) especially high perplexity via GPT-2, and 3) length of the rationales. Meanwhile, we also dynamically replaced the concept-sets that majority of the references do not make sense to ensure the final quality.

3.3 Down-Sampling Training Examples

In order to evaluate the *compositional generalization* ability, we down-sample the remaining candidate concept-sets to construct a distantly supervised training dataset (i.e., using caption sentences as the human references). We explicitly control the overlap of the concept-sets between training examples and dev and test examples. The basic statistics of the final dataset is shown in Table 1. There are on average four sentences for each example in dev and test sets, which provide a richer and more diverse test-bed for automatic and manual evaluation. Table 1 also shows the ratio of *unseen concept compositions* (i.e., concept, concept-pair, and concept-triple) in the dev and test. Notably, all pairs of concepts in every test concept-set are unseen in training data and thus pose a challenge for compositional generalization.

3.4 Analysis of Underlying Common Sense

We here introduce deeper analysis of the dataset by utilizing the largest commonsense knowledge graph (KG), ConceptNet (Speer et al., 2017), as an tool to study connectivity and relation types.

Connectivity Distribution. If the concepts inside a given concept-set is more densely connected

Category	Relations	1-hop	2-hop
Spatial knowledge	AtLocation, LocatedNear	9.40%	39.31%
Object properties	UsedFor,CapableOf,PartOf, ReceivesAction,MadeOf, FormOf, HasProperty,HasA	9.60%	44.04%
Human behaviors	CausesDesire,MotivatedBy, Desires,NotDesires,Manner	4.60%	19.59%
Temporal knowledge	Subevent, Prerequisite, First/Last-Subevent	1.50%	24.03%
General	RelatedTo, Synonym, DistinctFrom, IsA, HasContext,SimilarTo	74.89%	69.65%

Table 2: The **distributions of the relation categories** on one/two-hop connections.

with each other on the KG, then it is likely to be easier to write a scenario about them. In each 5-size concept-set (i.e. a concept-set consists of five concepts), there are 10 unique pairs of concepts, the connections of which we are interested in. As shown in Figure 4, if we look at the one-hop links on the KG, about 60% of the 5-size concept-set have less than one link among all concept-pairs. On the other hand, if we consider two-hop links, then nearly 50% of them are almost fully connected (i.e. each pair of concepts has connections). These two observations together suggest that the COMMONGEN has a reasonable difficulty: the concepts are not too distant or too close, and thus the inputs are neither too difficult nor too trivial.

Relation Distribution. Furthermore, the relation types of such connections can also tell us what kinds of commonsense knowledge are potentially useful for relational reasoning towards generation. We report the frequency of different relation types[2] of the one/two-hop connections among concept-pairs in the dev and test examples in Fig. 8. To better summarize the distributions, we categorize these relations into five major types and present their distribution in Table 2, respectively for one/two-hop connections between concept pairs.

4 Methods

We briefly introduce the baseline methods that are tested on the COMMONGEN task.

Encoder-Decoder Models. Bidirectional RNNs and Transformers (Vaswani et al., 2017) are two most popular architectures for seq2seq learning. We use them with the addition of attention mechanism (Luong et al., 2015) with copying ability (Gu et al., 2016), which are based on an open-source framework OpenNMT-py (Klein et al., 2017). We use `bRNN-CopyNet` and `Trans-CopyNet` denote them respectively. To alleviate the influence from the concept ordering in such sequential learning methods, we randomly permute them multiple times for training and decoding and then get their average performance. To explicitly eliminate the order-sensitivity of inputs, we replace the encoder with a mean pooling-based MLP network (`MeanPooling-CopyNet`).

Non-autoregressive generation. Recent advances (Lee et al., 2018; Stern et al., 2019) in conditional sentence generation have an emerging interest on (edit-based) non-autoregressive generation models, which iteratively refine generated sequences. We assume that these models potentially would have better performance because of their explicit modeling on iterative refinements, and thus study the most recent such model Levenshtein Transformer (`LevenTrans`) by Gu et al. (2019). We also include a recent enhanced version, `ConstLeven` (Susanto et al., 2020), which incorporates lexical constraints in `LevenTrans`.

Pre-trained Language Generation Models. We also employ various pre-trained language generation models, including `GPT-2` (Radford et al., 2019), `UniLM` (Dong et al., 2019), `UniLM-v2` (Bao et al., 2020), `BERT-Gen` (Bao et al., 2020), `BART` (Lewis et al., 2019), and `T5` (Raffel et al., 2019), to tackle this task and test their generative commonsense reasoning ability. We fine-tuned all the above models on our training data with a seq2seq format.

Specifically, to use `GPT-2` for this sequence-to-sequence task, we condition the language model on the format "$c_1\ c_2\ \ldots\ c_k = y$" during fine-tuning, where c_i is a concept in the given concept-set and connects with other concepts with a blank; y is a target sentence. For inference, we sample from the fine-tuned `GPT-2` model after a prompt of "$c_1\ c_2\ \ldots\ c_k =$" with beam search and use the first generated sentence as the output sentence. For `BERT-Gen`, we use the `s2s-ft` package[3] to fine-tune them in a sequence-to-sequence fashion that is similar to the LM objective employed by UniLM.

As for `T5`, the state-of-the-art text-to-text pre-trained model which is pre-trained with a multi-task objective by prepending a task description

[2] Relation definitions are at https://github.com/commonsense/conceptnet5/wiki/Relations.

[3] https://github.com/microsoft/unilm

Model \ Metrics	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	Coverage
bRNN-CopyNet (Gu et al., 2016)	7.61	27.79	10.70	5.70	15.80	4.79	15.00	51.15
Trans-CopyNet	8.78	28.08	11.90	7.10	15.50	4.61	14.60	49.06
MeanPooling-CopyNet	9.66	31.14	10.70	6.10	16.40	5.06	17.20	55.70
LevenTrans. (Gu et al., 2019)	10.58	32.23	19.70	11.60	20.10	7.54	19.00	63.81
ConstLeven. (Susanto et al., 2020)	11.82	33.04	18.90	10.10	24.20	10.51	22.20	94.51
GPT-2 (Radford et al., 2019)	17.18	39.28	30.70	21.10	26.20	12.15	25.90	79.09
BERT-Gen (Bao et al., 2020)	18.05	40.49	30.40	21.10	27.30	12.49	27.30	86.06
UniLM (Dong et al., 2019)	21.48	**43.87**	38.30	27.70	29.70	14.85	30.20	89.19
UniLM-v2 (Bao et al., 2020)	18.24	40.62	31.30	22.10	28.10	13.10	28.10	89.13
BART (Lewis et al., 2019)	**22.23**	41.98	36.30	26.30	**30.90**	13.92	30.60	**97.35**
T5-Base (Raffel et al., 2019)	14.57	34.55	26.00	16.40	23.00	9.16	22.00	76.67
T5-Large (Raffel et al., 2019)	22.01	42.97	**39.00**	**28.60**	30.10	**14.96**	**31.60**	95.29
Human Performance (Upper Bound)	48.88	63.79	48.20	44.90	36.20	43.53	63.50	99.31

Table 3: **Experimental results** of different baseline methods on the COMMONGEN test set. The first group of models are non-pretrained models, while the second group is large pretrained models that we have fine-tuned. The best models are **bold** and second best ones are underlined within each metric. We highlight the metrics that we used in our official leaderboard. (Results on dev set are at Table. 7.)

before the input text, we prepend the input concept set with a simple prompt: "`generate a sentence with:`" and fine-tune the model with the source sentence on the format "*generate a sentence with* c_1 c_2 ... c_k." For decoding, we employ the standard beam search with a beam size of 5 for all compared models. We also report their results with a lexically-constrained decoding method, dynamic beam allocation (DBA) (Post and Vilar, 2018), which do not show improvement over conventional beam searching. [4]

5 Evaluation

We first introduce the automatic evaluation metrics, then present main experimental results with manual analysis, and finally introduce the potential application in transferring CommonGen-trained models for other downstream tasks.

5.1 Metrics

Following other conventional generation tasks, we use several widely-used automatic metrics to automatically assess the performance, such as `BLEU` (Papineni et al., 2002), `ROUGE` (Lin, 2004), `METEOR` (Banerjee and Lavie, 2005), which mainly focus on measuring surface similarities. We report the concept `Coverage`, which is the average percentage of input concepts that are present in lemmatizatized outputs.

In addition, we argue that it is more suitable to use evaluation metrics specially design for captioning task, such as `CIDEr` (Vedantam et al., 2015) and `SPICE` (Anderson et al., 2016). They usually assume system generations and human references use similar concepts, and thus focus on evaluate the associations between mentioned concepts instead of n-gram overlap. For example, the `SPICE` metric uses dependency parse trees as proxy of scene graphs to measure the similarity of scenarios.[5]

To estimate *human performance* within each metric, we treat each reference sentence in dev/test data as a "system prediction" to be compared with all other references, which is equivalent to compute inter-annotator agreement within each metric. Thus, systems that have better generative ability than average crowd-workers should exceed this.

5.2 Experimental Results

Automatic Evaluation. Table 3 presents the experimental results in a variety of metrics. We can see that all fine-tuned pre-trained models (the lower group) outperform non-pretrained models (the upper group) with a significant margin. This is not surprising because their pretraining objectives, including masked language modeling, word ordering, and text infilling which predicts missing words or text spans, are relevant to our task. On the other hand, we find that the key disadvantage of non-pretrained models with CopyNet still falls in the

[4] The used hyper-parameters are reported in the appendix.

[5] We also tried recent metrics such as BERTScore (Zhang et al., 2020b), but we find that they overly focus on lexical semantics instead of dependencies between words, thus resulting low correlation with the manual evaluation results.

	C.Leven	GPT	BERT-G.	UniLM	BART	T5
Hit@1	3.2	21.5	22.3	21.0	26.3	**26.8**
Hit@3	18.2	63.0	59.5	69.0	69.0	**70.3**
Hit@5	51.4	95.5	95.3	96.8	96.3	**97.8**

Table 4: **Manual Evaluation via Pair-wise Comparisons for Ranking.** Numbers are hit rates (%) at top 1/3/5.

Concept-Set: { hand, sink, wash, soap }

[bRNN-CopyNet]: a hand works in the sink .

[MeanPooling-CopyNet]: the hand of a sink being washed up

[ConstLeven]: a hand strikes a sink to wash from his soap.

[GPT-2]: hands washing soap on the sink.

[BERT-Gen]: a woman washes her hands with a sink of soaps.

[UniLM]: hands washing soap in the sink

[BART]: a man is washing his hands in a sink with soap and washing them with hand soap.

[T5]: hand washed with soap in a sink.

1. A girl is washing her hands with soap in the bathroom sink.
2. I will wash each hand thoroughly with soap while at the sink.
3. The child washed his hands in the sink with soap.
4. A woman washes her hands with hand soap in a sink.
5. The girl uses soap to wash her hands at the sink.

Figure 5: A case study with a concept-set {*hand*, *sink*, *wash*, *soap*} for qualitative analysis of machine generations. Human references are collected from AMT.

failure of using all given concepts (i.e., low coverage), which results in worse results.

Among them, UniLM, BART, and T5 performs the best, which may be due to its inherent sequence-to-sequence pre-training framework. We found that BART has the best concept coverage, which is probably due to its comprehensive pre-training tasks that aim to recover text with noise. The results suggest that further modifying pre-trained models is a promising direction for generative commonsense.

Manual Evaluation. We conduct manual evaluation with a focus on *commonsense plausibility* for comparing the 6 best-performing models in Table 4. We ask five graduate students to compare 1,500 pairs of model-generated sentences respectively, for ranking the models within 100 concept-sets that are covered by all the models. The final average ranked results are shown in Table 4 and their inter-annotator agreement is 0.85 in *Kendall's rank correlation coefficient*.

Note that the coverage-weighted hit@1 rate correlates with the SPICE metric the most, i.e., **0.94** in *Spearman's* ρ for model ranks, while METEOR and ROUGE-2 are both 0.88 and BLEU-4 is 0.78.

Case study. Fig. 5 shows the top generations of different models and human references about an input concept-set: {*hand*, *sink*, *soup*, *wash*} (more cases are shown in Fig. 9 in the appendix). We find that non-pretrained seq2seq models (e.g., bRNN, MeanPooling, ConstLeven) can successfully use part of given concepts, while the generated sentences are less meaningful and coherent. On the contrary, the outputs of fine-tuned pre-trained language models are significantly more commonsensical. Most of them use all given concepts in their outputs. ConstLeven tends to make use of frequent patterns to compose a non-sense sentence but uses all concepts. GPT-2 and UniLM incorrectly compose the dependency among *hand*, *wash*, and *soap*. The phrase 'a sink of soaps' in BERT-gen's output makes itself less common. BART and T5 generate relatively reasonable scenarios, but both are not as natural as human references; BART's contains repetitive content while T5's lacks a human agent.

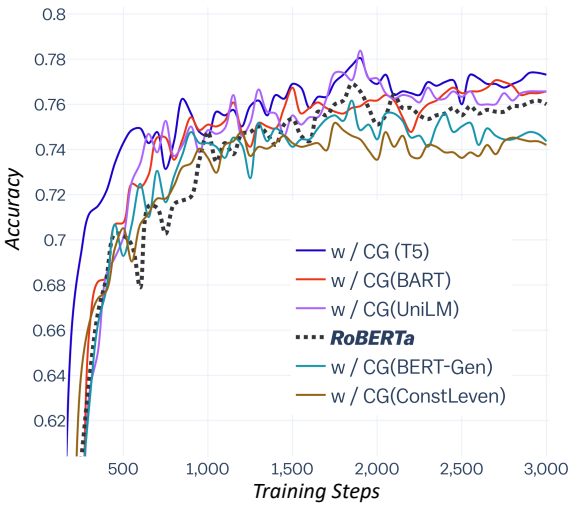

Figure 6: **Learning curve for the transferring study.** We use several trained COMMONGEN (GG) models to generate choice-specific context for the CSQA task. Detailed numbers are shown in Tab. 8 in the appendix.

Influence of Dynamic Beam Allocation. Considering that all tested models decode sentences with beam searching, one may wonder what if we use a decoding method specially designed for constrained decoding. Thus, we employed dynamic beam allocation (DBA) (Post and Vilar, 2018). The results are shown in Table 5. Note that the models are the same as in Table 3 while only the decoding method is changed to DBA. We can see that all methods are negatively impacted by the decoding method. This suggests that for the COMMONGEN task and pre-trained language models, we may need to focus on knowledge-based decoding

Model \ Metrics	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	Coverage
T5-large+DBA	16.8	36.71	27.3	18.7	25.3	8.62	24.3	83.98
T5-base+DBA	15.07	34.82	24.8	16	23.5	9.31	21.3	76.81
GPT-2+DBA	17.56	39.45	29.4	20.6	24.9	10.85	26.8	79.51
BART+DBA	18.15	37.02	28.3	19.1	25.5	9.82	25.1	84.78

Table 5: Experimental results of models with DBA decoding method on the test set.

or re-ranking as future directions.

5.3 Transferring CommonGen Models

One may wonder how fine-tuned COMMONGEN models can benefit commonsense-centric downstream tasks such as Commonsense Question Answering (Talmor et al., 2019) (CSQA) with their generative commonsense reasoning ability. To this end, we use the models trained with the COMMONGEN dataset for generating useful context.

We extract the nouns and verbs in questions and all choices respectively, and combine the concepts of the question q and each choice c_i to build five concept-sets. Then, we use these concept-sets as inputs to a trained COMMONGEN model (e.g., T5) for generating scenario a sentence g_i for each as choice-specific contexts. Finally, we prepend the outputs in front of the questions, i.e., "<s>G: g_i | Q: q </s> C: c_i </s>". Note that the state-of-the-art RoBERTa-based models for CSQA uses the same form without "G: g_i|" in fine-tuning.

We show the learning-efficiency curve in Fig. 6, where y is the accuracy on the official dev set and x is the number of training steps. The details of the experiments are shown in the appendix.

We highlight the performance of original RoBERTa-Large as the baseline. We find that some CommonGen models further improves the performance by a large margin, e.g., 76.9 $\xrightarrow{\text{UniLM}}$ 78.4 and they converge at better accuracy in the end. Note that BERT-gen and ConstLeven cause negative transfer due to the low quality of generated context. Particularly, we find that the context generated by the T5-based CommonGen model (CG-T5) helps speed up training about 2 times, if we look at 550th steps of CG-T5 (74.85%) and 1,250th steps of original RoBERTa (74.77%).

Through manual analysis, we find that the successful COMMONGEN models can generate more reasonable and natural sentence for correct choices while noisy sentences for wrong choices. For example with CG (T5), q="*What do people aim to do at work?*", c_i='complete job' (✓) with g_i="*people work to complete a job aimed at achieving a certain goal.*"; c_j='wear hats' (✗) g_j="*people wearing hats aim their guns at each other while working on a construction site.*" The used question concepts and choice concepts are underlined.

6 Related Work

Commonsense benchmark datasets. There are many emerging datasets for testing machine commonsense from different angles, such as commonsense extraction (Xu et al., 2018; Li et al., 2016), next situation prediction (SWAG (Zellers et al., 2018), CODAH (Chen et al., 2019), HellaSWAG (Zellers et al., 2019b)), cultural and social understanding (Lin et al., 2018; Sap et al., 2019a,b), visual scene comprehension (Zellers et al., 2019a), and general commonsense question answering (Talmor et al., 2019; Huang et al., 2019; Wang et al., 2019a, 2020). However, the success of fine-tuning pre-trained language models for these tasks does not necessarily mean machines can produce novel assumptions in a more open, realistic, generative setting. We see COMMONGEN as a novel, complementary commonsense reasoning benchmark task for advancing machine commonsense in NLG.

Constrained Text Generation. Constrained text generation aims to decode sentences with expected attributes such as sentiment (Luo et al., 2019a; Hu et al., 2017), tense (Hu et al., 2017), template (Zhu et al., 2019; J Kurisinkel and Chen, 2019), style (Fu et al., 2018; Luo et al., 2019b; Li et al., 2018), topics (Feng et al., 2018), etc. Two related scenarios with our task is lexically constrained decoding and word ordering (Zhang and Clark, 2015; Hasler et al., 2018; Dinu et al., 2019; Hokamp and Liu, 2017; Puduppully et al., 2017; Miao et al., 2019). However, they are not easily adopted by the recent pre-trained language models and thus not directly useful for our task. Topical story generation (Fan et al., 2018; Yao et al., 2019) is also a related direction, while it targets generating longer, creative stories around the given topics, making it hard to directly adopt them to our task. Additionally, the

COMMONGEN task brings some more challenges mentioned in Section 2. Prior constrained generation methods cannot address these issues together in a unified model.

Incorporating Commonsense for NLG. There are a few recent works that incorporate commonsense knowledge in language generation tasks such as essay generation (Guan et al., 2019; Yang et al., 2019a), image captioning (Lu et al., 2018), video storytelling (Yang et al., 2019b), and conversational systems (Zhang et al., 2020a). These works suggest that generative commonsense reasoning has a great potential to benefit downstream applications. Our proposed COMMONGEN, to the best of our knowledge, is the very first constrained sentence generation dataset for assessing and conferring generative machine commonsense and we hope it can benefit such applications. Our transferring study in Sec. 5.3 also shows the potential benefits of CommonGen-generated contexts.

7 Conclusion

Our major contribution in this paper are threefold:

- we present COMMONGEN, a novel constrained text generation task for generative commonsense reasoning, with a large dataset;
- we carefully analyze the inherent challenges of the proposed task, i.e., a) relational reasoning with latent commonsense knowledge, and b) compositional generalization.
- our extensive experiments systematically examine recent pre-trained language generation models (e.g., UniLM, BART, T5) on the task , and find that their performance is still far from humans, generating grammatically sound yet realistically implausible sentences.

Our study points to interesting future research directions on modeling commonsense knowledge in language generation process, towards conferring machines with generative commonsense reasoning ability. We hope COMMONGEN would also benefit downstream NLG applications such as conversational systems and storytelling models.

Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007, the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, the Defense Advanced Research Projects Agency with award W911NF-19-20271, and NSF SMA 18-29268. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. We would like to thank all the collaborators in USC INK research lab for their constructive feedback on the work.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. 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. Association for Computational Linguistics.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiulei Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv: Computation and Language*.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. Codah: An adversarially authored question-answer dataset for common sense. *ArXiv*, abs/1904.04365.

Noam Chomsky. 1965. Aspects of the theory of syntax.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58:92–103.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-essay generation with neural networks. In *IJCAI*, pages 4078–4084.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pages 11179–11189.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Litton J Kurisinkel and Nancy Chen. 2019. Set to ordered text: Generating discharge instructions from medical billing codes. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6165–6175, Hong Kong, China. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Brenden M Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In .

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Bill Yuchen Lin, Frank F. Xu, Kenny Zhu, and Seungwon Hwang. 2018. Mining cross-cultural differences and similarities in social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 709–719, Melbourne, Australia. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural baby talk. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7219–7228. IEEE Computer Society.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019a. Towards fine-grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2022, Florence, Italy. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019b. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Chris Moore. 2013. *The development of commonsense psychology*. Psychology Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Ratish Puduppully, Yue Zhang, and Manish Shrivastava. 2017. Transition-based deep input linearization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 643–654, Valencia, Spain. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie description. *International Journal of Computer Vision*, 123(1):94–120.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *ArXiv*, abs/1907.10641.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*.

Raymond Hendy Susanto, Shamil Chollampatt, and Li ling Tan. 2020. Lexically constrained neural machine translation with levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. To appear.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Ruth Tincoff and Peter W Jusczyk. 1999. Some beginnings of word comprehension in 6-month-olds. *Psychological science*, 10(2):172–175.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019a. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy. Association for Computational Linguistics.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019b. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4581–4591.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Frank F. Xu, Bill Yuchen Lin, and Kenny Zhu. 2018. Automatic extraction of commonsense LocatedNear knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 96–101, Melbourne, Australia. Association for Computational Linguistics.

Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019a. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2002–2012, Florence, Italy. Association for Computational Linguistics.

Pengcheng Yang, Fuli Luo, Peng Chen, Lei Li, Zhiyi Yin, Xiaodong He, and Xu Sun. 2019b. Knowledgeable storyteller: a commonsense-driven generative model for visual storytelling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, pages 5356–5362.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6720–6731.

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

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

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020a. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. To appear.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. "bertscore: Evaluating text generation with bert". In *International Conference on Learning Representations*.

Yue Zhang and Stephen Clark. 2015. Discriminative syntax-based word ordering for text generation. *Computational Linguistics*, 41:503–538.

Wanrong Zhu, Zhiting Hu, and Eric P. Xing. 2019. Text infilling. *ArXiv*, abs/1901.00158.

A Supplementary Figures and Tables

We include additional figures and tables that we mentioned in the main content here.

- Figure 8 shows the detailed **distribution of the commonsense relations** between given concepts, the summary of which was shown in Table 2 of the main content.
- Figure 9 presents 4 more **case studies** with human rationales which we asked our crowd workers to provide.
- Figure 7 shows instructions and AMT interface for crowd-sourcing human references.
- Table 7 shows the model performances on the dev set of COMMONGEN, as a reference for future development.
- Table 8 is the full results of the learning curve in Figure 5. We highlight the highest checkpoints and the speed-up by the CG-T5, which are discussed in Section 5.3.

B Experimental Details

Main experiments. We present some implementation details in training and testing the baseline models in Table 6. The detailed instructions for installing dependencies and all necessary training command-lines are shown in the instruction '**readme.md**' files. The number of trainable model parameters are directly induced from either output of the frameworks or the original papers. We show some key hyper-parameters that we manually tuned on top of the development set.

All key hyper-parameters were initialized by the default values as suggested by the original authors of the frameworks. The bound of our manual tuning is done by iterating the magnitudes or the neighboring choices, for example, the learning rates ('lr') of the last seven models are selected from $\{1e-3, \ldots, 1e-4, \ldots, 1e-5\}$. Then, similarly, the batch size (bsz) is first maximized by making full use of the GPU memory. Note that the first three models are implemented with the OpenNMT-py framework [6]. The `LevenTrans`[7], `ConstLeven`[8], and `BART`[9] are adopted by the official authors' release. The

Models	Instruction Files	#Para	Key HPs
bRNN-CopyNet	opennmt_based/README.md	8.12 M	lr=0.2, bsz=128, layers=2, rnn_size=128, dropout=0,
Trans-CopyNet	opennmt_based/README.md	6.25 M	lr=0.2, bsz=128, layers=1, hidden_size=128, dropout=0.1,
MeanPooling-CopyNet	opennmt_based/README.md	7.76 M	global_attention=mlp, lr=0.15, rnn_size=128, bsz=128
LevenTrans.	fairseq_based/README.md	55.4 M	lr=5e-4, warmup-init-lr=1e-7, dropout=0.3, warmup=10k
ConstLeven	const-levt/readme.md	55.4 M	lr=5e-4, warmup-init-lr=1e-7, dropout=0.3, warmup=10k
GPT-2	GPT-2/readme.md	345 M	lr=5e-5, bsz=32*4
BERT-Gen	BERT-based/readme.md	110 M	lr=3e-5, bsz = 32,
UniLM	unilm_based/README.md	340 M	lr=1e-5, bsz = 32
UniLMv2	UniLM_v2/readme.md	110 M	lr=3e-5, bsz = 32
BART	BART/readme.md	400 M	lr=3e-5, warmup= 500, bsz=32
T5-Base	T5/readme.md	220 M	lr=5e-5, bsz = 192
T5-Large	T5/readme.md	770 M	lr=2e-5, bsz = 2*32, warmup_steps=400

Table 6: The paths to the instruction files in our submitted code zip file (under the '*methods/*' folder), and their numbers of parameters and key hyper-parameters.

`BERT-gen`, `UniLM`, `UniLMv2` are all based on their official source code[10]. The `GPT-2` and `T5` are both adopted by the `huggingface` transformers[11] framework (Wolf et al., 2019). All models use beam searching as their decoding algorithms and beam-size are mostly 5, which is selected from $\{5, 10, 20\}$. All our models were trained on Quadro RTX 6000 GPUs. The training time of X-CopyNet and LevenTrans models are less than 12 hours with a single GPU. The second group of models are trained between 12 and 24 hours, expect for T5-large, which we used 3 GPUs and fine-tuned about 48 hours. *Note that all the above methods are self-contained in our submitted code as long as users follow the associated readme instructions.*

Transferring study experiments. We use the same hyper-parameters which are searched over the baseline RoBERTa-Large model for these experiments. The best hyper-parameter[12] of RoBERTa-Large for `CommonsenseQA`[13]:

- batch size = 16, learning rate = 1e-5,
- maximum updates = 3,000 (~5 epochs)
- warmup steps=150, dropout rate=0.1
- weight decay = 0.01, adam_epsilon = 1e-6

We tried 10 random seeds and use the best one (42). Then, we follow the steps described in Sec. 5.3 to run other CG-enhanced models with the

[6] https://github.com/OpenNMT/OpenNMT-py

[7] https://github.com/pytorch/fairseq/blob/master/examples/nonautoregressive_translation/README.md

[8] https://github.com/raymondhs/constrained-levt

[9] https://github.com/pytorch/fairseq/tree/master/examples/bart

[10] https://github.com/microsoft/unilm

[11] https://github.com/huggingface/transformers

[12] We follow the hps selected by 100 trials of tuning in https://github.com/pytorch/fairseq/tree/master/examples/roberta/commonsense_qa.

[13] https://www.tau-nlp.org/commonsenseqa

Model \ Metrics	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	Coverage
bRNN-CopyNet (Gu et al., 2016)	9.23	30.57	13.60	7.80	17.40	6.04	16.90	58.95
Trans-CopyNet	11.08	32.57	17.20	10.60	18.80	7.02	18.00	62.16
MeanPooling-CopyNet	11.36	34.63	14.80	8.90	19.20	7.17	20.20	68.32
LevenTrans. (Gu et al., 2019)	12.22	35.42	23.10	15.00	22.10	8.94	21.40	71.83
ConstLeven. (Susanto et al., 2020)	13.47	35.19	21.30	12.30	25.00	11.06	23.20	96.87
GPT-2 (Radford et al., 2019)	17.74	41.24	32.70	23.30	27.50	13.26	27.60	85.46
BERT-Gen (Bao et al., 2020)	18.73	42.36	33.00	23.70	29.10	13.34	28.70	91.71
UniLM (Dong et al., 2019)	21.68	**45.66**	40.40	30.40	**31.00**	15.72	31.40	92.41
UniLM-v2 (Bao et al., 2020)	19.24	43.01	33.40	24.20	29.20	13.65	29.30	93.57
BART (Lewis et al., 2019)	**22.13**	43.02	37.00	27.50	**31.00**	14.12	30.00	**97.56**
T5-Base (Raffel et al., 2019)	15.33	36.20	28.10	18.00	24.60	9.73	23.40	83.77
T5-Large (Raffel et al., 2019)	21.98	44.41	**40.80**	**30.60**	**31.00**	**15.84**	**31.80**	97.04
Human Performance	48.88	63.79	48.20	44.90	36.20	43.53	63.50	99.31

Table 7: Experimental results of different baseline methods on the COMMONGEN dev set. The first group of models are non-pretrained models, while the second group is large pretrained models that we have fine-tuned. The best models are **bold** and second best ones are underlined within each metric.

same hps. This suggests that further searching for them may have even better performance.

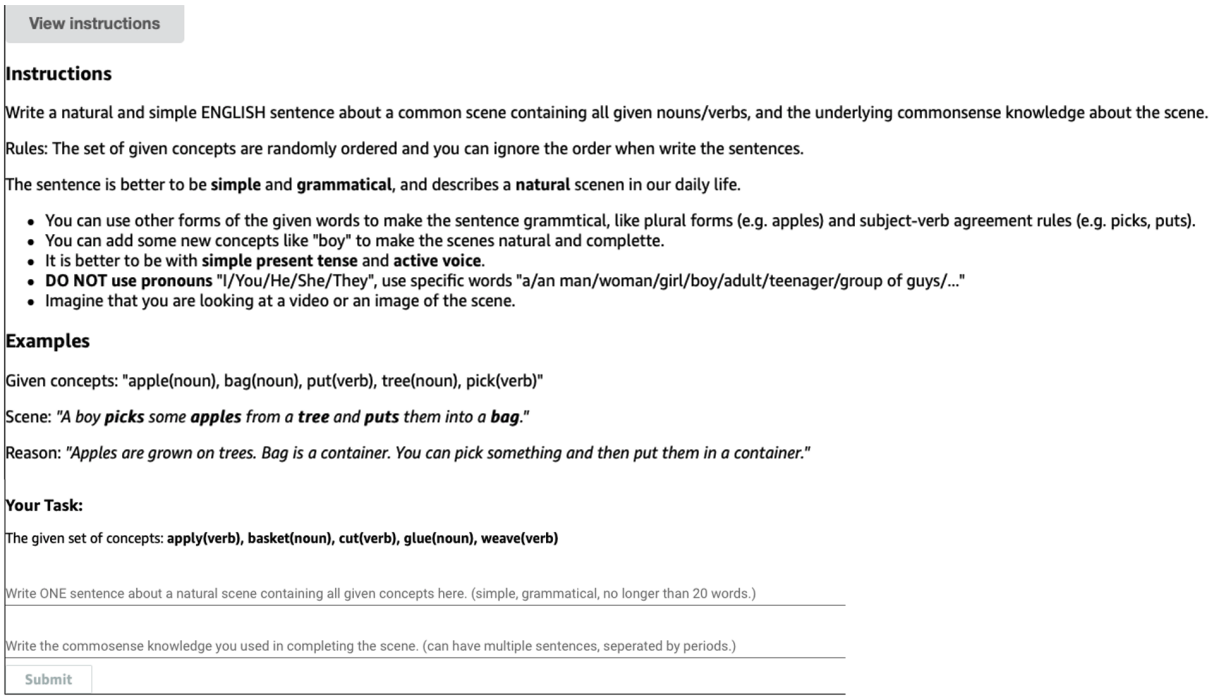

Figure 7: Our annotation interface on the AMT platform. The upper part is the instruction for the annotators and we provide an example for them. Note that we give the part-of-speech hints (from the captain corpora) to boost the speed of annotation, but we do not remove sentences with wrong part-of-speech as long as they also make sense.

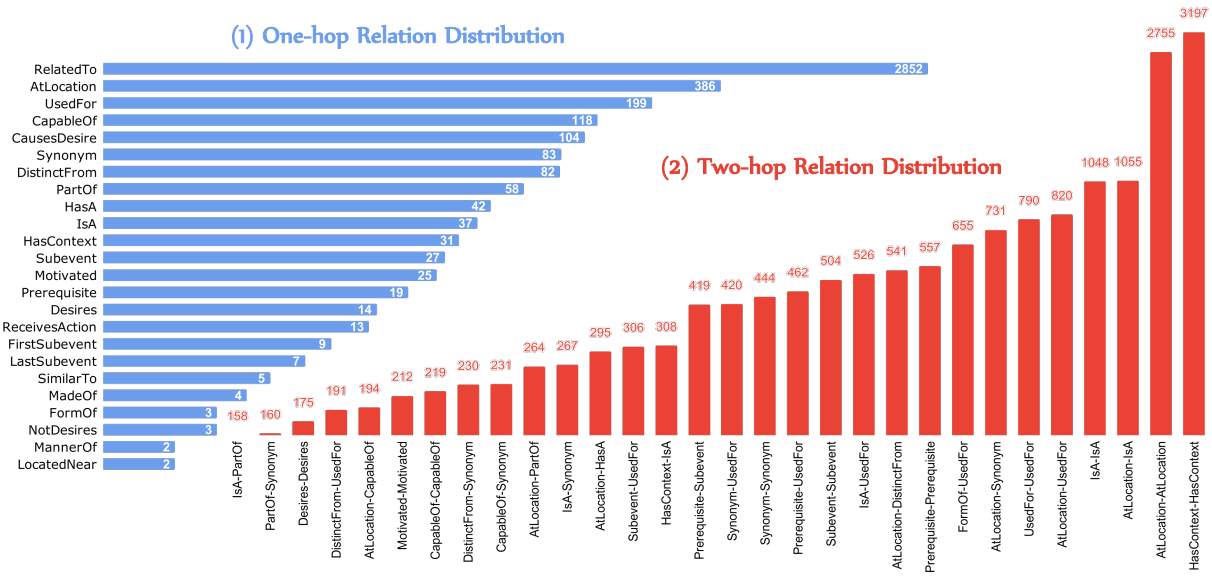

Figure 8: One/two-hop relation frequency in the COMMONGEN dev.&test sets on ConceptNet.

1) [Input concept-set]: { give, lay, massage, table }

[Machine generations]

[bRNN-CpNet]: Lays massage someone table vertical gives on and the water.
[Trans-CpNet]: Massage lays on the kitchen.
[MP-CpNet]: A massage table being calling with an improvisation lay free speaker.
[LevenTrans]: A man chatting at the table.
[GPT-2]: A man gives a massage to a table.
[BERT-Gen]: A woman lays down on a table and gives a massage to a man.
[UniLM]: A woman lays down a massage on a table and gives a massage.
[UniLM-v2]: A woman is laying down and giving a massage on a table.
[BART]: A man lays on a table and gives a massage to a woman laying on the table.
[T5]: Woman lay on a table and gives a massage.

[Human references from AMT]

1. The man lays down on the massage table and the therapist gives him a massage.
[Rationale]: The man must lay down to receive a massage. The therapist is the giver of massages. The table is a massage table.
2. Lay down on the table and the masseuse will give you a neck massage.
[Rationale]: A masseuse is a woman who gives massages professionally. Massages are usually done on tables.
3. The woman gives the man who lays on the table a massage.
[Rationale]: Some massages are done laying down; people like to get massages; tables are used for people to get massages; people lay on tables to get massages.

2) [Input concept-set]: { cow, horse, lasso, ride }

[Machine generations]

[bRNN-CpNet]: Someone lowers his horse from the wall and lasso glass by cows.
[Trans-CpNet]: A horse having lasso in the bridal cows.
[MP-CpNet]: Cow in a lasso getting the ride.
[LevenTrans]: A cow rides through a horse.
[GPT-2]: A horse rides on a lasso.
[BERT-Gen]: A cow rides a lasso on a horse.
[UniLM]: A man rides a horse with a lasso at cows.
[UniLM-v2]: A horse rides a cow with a lasso on it.
[BART]: A man rides a horse and a cow on a bridle with a lasso.
[T5]: Lasso to ride a cow on a horse.

[Human references from AMT]

1. When those men ride a horse for the first time and lasso those cows.
[Rationale]: cowboys ride horses and lasso cows for a living

2. A cowboy can use a lasso to control a horse or cow in order to ride them.
[Rationale]: I understand the words and I can read and write English.

3. The cowboy will lasso the cow while riding on the horse.
[Rationale]: Have seen it.

3) [Input concept-set]: { hand, hold, walk, water }

[Machine generations]

[bRNN-CpNet]: Process of holds at hands under walk on hours.
[Trans-CpNet]: Hands with a walk in the water.
[MP-CpNet]: Walk across the hold to water.
[LevenTrans]: Hand moored at the water.
[GPT-2]: A woman holds a water walker and holds a hand.
[BERT-Gen]: A man walking and holding a hand in water while walking.
[UniLM]: A man holds hands to walk across the water.
[UniLM-v2]: A man is walking and holding a hand in the water.
[BART]: A man walks with a woman holding her hand as they walk through water.
[T5]: Man holds a bottle of water in his hand as he walks along a river.

[Human references from AMT]

1. The couple holds hands as they walk by the water.
[Rationale]:
Couples hold hands when taking walk even by a body of water.

2. The girl is walking holding in her hand a bottle of water.
[Rationale]: I see this reading the words

3. The couple hold hands while they walk by the water.
[Rationale]: People sometimes hold hands. People Like to walk near water.

4) [Input concept-set]: { clean, ladder, squeegee, stand, window }

[Machine generations]

[bRNN-CpNet]: The window stands out a ladder but clean the sun to being squeegee.
[Trans-CpNet]: A brown leather ladder with green eyes.
[MP-CpNet]: Window of the zebra are on a tablecloth.
[LevenTrans]: A man on a a on on the kitchen.
[GPT-2]: Someone grabs a ladder from a window and squeezes it open.
[BERT-Gen]: A woman is cleaning a window with a ladder and a squeegee.
[UniLM]: Someone stands next to a window and stands on a ladder to clean the squeegee.
[UniLM-v2]: A man is standing on a ladder and using a ladder to clean the window.
[BART]: A man with a squeegee and a ladder standing on the ledge of a window is cleaning the window.
[T5]: Squeegee and ladder on a wooden stand to clean windows and windows.

[Human references from AMT]

1. The window cleaner stands on the ladder to clean the window with a squeegee.
[Rationale]: A squeegee is a tool to clean windows. A ladder is something that people use to reach high places.
2. The man clean the window on the ladder stand by using squeegee.
[Rationale]: man need to clean the window by using squeegee on the ladder stand
3. The man stood beside the ladder and cleaned the window with a squeegee.
[Rationale]: people can stand next to ladders. People clean windows. Squeegees are used to clean windows.

Figure 9: Four cases for qualitative analysis of machine generations. References are collected from AMT crowd-workers and they are required to provide rationales. Note that the third one is a positive case showing that some models can successfully generate reasonable scenarios. However, most models perform poorly on the other cases.

Training Steps	RoBERTa-Large	w/CG(BART)	w/CG(T5)	w/CG(UniLM)	w/CG(BERT-Gen)	w/CG(ConstLeven)
50	0.2252	0.1884	0.2506	0.2244	0.2007	0.2162
100	0.3088	0.2703	0.3587	0.3153	0.2924	0.2809
150	0.5053	0.2973	0.5643	0.1851	0.3391	0.3653
200	0.5717	0.4439	0.6650	0.3833	0.5274	0.5324
250	0.6020	0.5242	0.6937	0.5348	0.5839	0.6396
300	0.6388	0.6601	0.7117	0.6323	0.6274	0.6634
350	0.6675	0.6814	0.7150	0.6503	0.6626	0.6740
400	0.6830	0.6830	0.7215	0.6847	0.6781	0.6773
450	0.7027	0.7068	0.7338	0.6921	0.7068	0.6962
500	0.7019	0.7076	0.7428	0.7011	0.6929	0.7052
550	0.6978	0.7248	0.7486	0.7256	0.7068	0.6904
600	0.6790	0.7232	0.7494	0.7338	0.7248	0.7068
650	0.7150	0.7289	0.7428	0.7469	0.7101	0.7117
700	0.7142	0.7453	0.7477	0.7387	0.7305	0.7183
750	0.7027	0.7453	0.7314	0.7527	0.7166	0.7183
800	0.7158	0.7355	0.7437	0.7371	0.7281	0.7240
850	0.7174	0.7445	0.7625	0.7420	0.7379	0.7322
900	0.7191	0.7543	0.7559	0.7502	0.7477	0.7338
950	0.7355	0.7486	0.7477	0.7387	0.7428	0.7404
1000	0.7477	0.7510	0.7461	0.7486	0.7428	0.7363
1050	0.7346	0.7502	0.7568	0.7469	0.7412	0.7297
1100	0.7428	0.7527	0.7551	0.7494	0.7363	0.7420
1150	0.7379	0.7609	0.7576	0.7641	0.7453	0.7437
1200	0.7469	0.7477	0.7502	0.7461	0.7420	0.7477
1250	0.7477	0.7412	0.7592	0.7518	0.7273	0.7371
1300	0.7502	0.7518	0.7617	0.7666	0.7518	0.7412
1350	0.7469	0.7502	0.7551	0.7568	0.7437	0.7404
1400	0.7420	0.7494	0.7641	0.7559	0.7494	0.7428
1450	0.7510	0.7584	0.7625	0.7461	0.7461	0.7461
1500	0.7535	0.7674	0.7690	0.7551	0.7412	0.7428
1550	0.7461	0.7559	0.7674	0.7510	0.7445	0.7412
1600	0.7437	0.7584	0.7584	0.7543	0.7445	0.7420
1650	0.7568	0.7609	0.7633	0.7543	0.7494	0.7428
1700	0.7551	0.7584	0.7633	0.7625	0.7535	0.7396
1750	0.7600	0.7568	0.7699	0.7740	0.7551	**0.7518**
1800	0.7617	0.7559	0.7731	0.7740	0.7527	0.7486
1850	**0.7690**	0.7584	0.7772	0.7707	**0.7617**	0.7461
1900	0.7658	0.7592	**0.7805**	**0.7838**	0.7486	0.7445
1950	0.7584	0.7617	0.7715	0.7715	0.7510	0.7396
2000	0.7510	0.7617	0.7690	0.7715	0.7445	0.7355
2050	0.7551	0.7641	0.7731	0.7649	0.7559	0.7477
2100	0.7641	0.7617	0.7641	0.7625	0.7559	0.7412
2150	0.7584	0.7543	0.7658	0.7641	0.7527	0.7461
2200	0.7584	0.7477	0.7649	0.7633	0.7453	0.7371
2250	0.7551	0.7559	0.7641	0.7609	0.7461	0.7363
2300	0.7535	0.7600	0.7699	0.7674	0.7412	0.7420
2350	0.7551	0.7617	0.7682	0.7625	0.7502	0.7412
2400	0.7559	0.7649	0.7699	0.7625	0.7559	0.7387
2450	0.7584	0.7674	0.7707	0.7658	0.7477	0.7387
2500	0.7551	0.7649	0.7600	0.7633	0.7502	0.7363
2550	0.7592	0.7658	0.7731	0.7658	0.7518	0.7387
2600	0.7559	0.7658	0.7715	0.7600	0.7420	0.7371
2650	0.7576	0.7674	0.7690	0.7600	0.7494	0.7420
2700	0.7568	**0.7707**	0.7690	0.7600	0.7461	0.7379
2750	0.7568	0.7699	0.7674	0.7649	0.7445	0.7437
2800	0.7592	0.7682	0.7690	0.7617	0.7445	0.7453
2850	0.7592	0.7641	0.7707	0.7649	0.7461	0.7445
2900	0.7609	0.7649	0.7740	0.7658	0.7477	0.7437
2950	0.7617	0.7649	0.7740	0.7658	0.7469	0.7437
3000	0.7600	0.7658	0.7731	0.7658	0.7437	0.7420

Table 8: Experimental results of the transferring study on CommonsenseQA dev set.