DiffuCOMET: Contextual Commonsense Knowledge Diffusion

Silin Gao[1], **Mete Ismayilzada**[1], **Mengjie Zhao**[2], **Hiromi Wakaki**[2]
Yuki Mitsufuji[2], **Antoine Bosselut**[1†]
[1]NLP Lab, IC, EPFL, Switzerland, [2]Sony Group Corporation, Tokyo, Japan
[1]`{silin.gao,mahammad.ismayilzada,antoine.bosselut}@epfl.ch`
[2]`{mengjie.zhao,hiromi.wakaki,yuhki.mitsufuji}@sony.com`

Abstract

Inferring contextually-relevant and diverse commonsense to understand narratives remains challenging for knowledge models. In this work, we develop a series of knowledge models, DiffuCOMET, that leverage diffusion to learn to reconstruct the implicit semantic connections between narrative contexts and relevant commonsense knowledge. Across multiple diffusion steps, our method progressively refines a representation of commonsense facts that is anchored to a narrative, producing contextually-relevant and diverse commonsense inferences for an input context. To evaluate DiffuCOMET, we introduce new metrics for commonsense inference that more closely measure knowledge diversity and contextual relevance. Our results on two different benchmarks, $\mathcal{C}om\mathcal{F}act$ and WebNLG+, show that knowledge generated by DiffuCOMET achieves a better trade-off between commonsense diversity, contextual relevance and alignment to known gold references, compared to baseline knowledge models.[1]

1 Introduction

Identifying the commonsense inferences that underlie narratives, such as stories or dialogues (Guan et al., 2019; Zhou et al., 2022), is crucial to understanding those same narratives. For example, to understand why "Hank ... got the shopping bags" in the context in Figure 1, a model would need to infer that (1) Hank was not finished wrapping gifts, and so (2) would need to buy more wrapping paper. However, comprehensively inferring these diverse, yet implicit, commonsense inferences that are relevant to a context remains a challenging task.

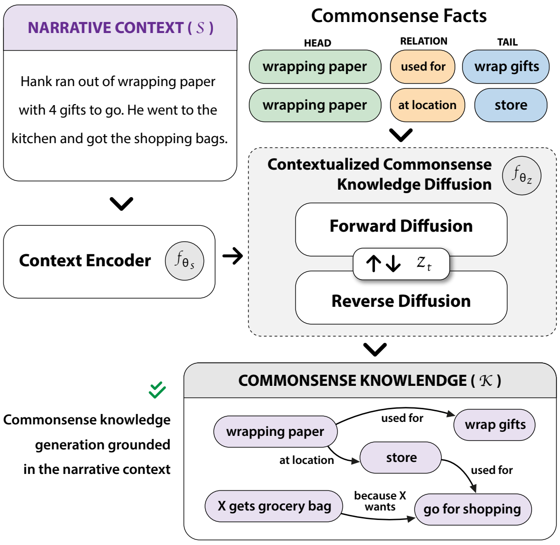

NARRATIVE CONTEXT ($\mathcal{S}$)
Hank ran out of wrapping paper with 4 gifts to go. He went to the kitchen and got the shopping bags.
Context Encoder
f_{θ_s}
Commonsense Facts
HEAD
RELATION
TAIL
wrapping paper
used for
wrap gifts
wrapping paper
at location
store
Contextualized Commonsense Knowledge Diffusion
f_{θ_z}
Forward Diffusion
z_t
Reverse Diffusion
COMMONSENSE KNOWLENDGE ($\mathcal{K}$)
Commonsense knowledge generation grounded in the narrative context
wrapping paper
used for
wrap gifts
at location
store
used for
X gets grocery bag
because X wants
go for shopping

Figure 1: Overview of our diffusion-based contextual commonsense knowledge generation.

Recent methods for identifying contextually-relevant commonsense inferences (Bosselut et al., 2021; Tu et al., 2022; Peng et al., 2022) use knowledge models (Bosselut et al., 2019; West et al., 2022) to generate commonsense facts. While knowledge models have been less brittle than previous retrieval-based methods for commonsense inference, they have two major shortcomings. First, they are trained to verbalize tuples from general commonsense knowledge graphs (Sap et al., 2019; Hwang et al., 2021), leading them to produce valid, but often contextually-irrelevant, commonsense inferences when applied out-of-the-box to real narratives. Second, because they are trained using autoregressive training objectives, they subsequently decode high-likelihood, non-diverse sequences that only identify limited collections of commonsense inferences relevant to an input context.

In this work, we address these challenges of contextual commonsense knowledge generation by developing **Diffu**sion (Ho et al., 2020) **COM**monsEnse **T**ransformer (Bosselut et al., 2019) models. DiffuCOMET models (shown in

[†]Corresponding author.
[1]We release our code to the community at `https://github.com/Silin159/DiffuCOMET`

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4809–4831
August 11-16, 2024 ©2024 Association for Computational Linguistics

Figure 1) uses diffusion-based decoding to generate relevant knowledge embeddings that are constrained to the narrative context. Over multiple iterations of constrained diffusion, our models refine a latent representation of the semantic connections between a context and its relevant facts, ensuring that it generates commonsense knowledge that is more contextually relevant to the narrative. At the same time, by jointly refining multiple fact embeddings during diffusion, DIFFUCOMET also generates more diverse inferences than comparable-size autoregressive knowledge models.

We evaluate DIFFUCOMET models using traditional NLG metrics (*e.g.*, BLEU; Papineni et al., 2002) commonly used for evaluating knowledge models. However, these metrics focus on surface form matching to gold references, and fall short of measuring the diversity of commonsense inferences and their semantic relevance to real narrative contexts. Our second contribution is a novel set of metrics that assess the diversity and contextual relevance of knowledge generated by knowledge models. Using both the traditional evaluation metrics and our new suite, we evaluate our models on a commonsense inference linking benchmark (Gao et al., 2022a) that covers both social and physical knowledge, and a second knowledge generation benchmark that involves extracting RDF triplets from language, WebNLG+ (Ferreira et al., 2020).

Our result show that DIFFUCOMET models generate knowledge that achieves a better balance of diversity and contextual relevance compared to other knowledge models. DIFFUCOMET models also more robustly generalize to generate knowledge for out-of-distribution narratives, and are better at producing novel knowledge tuples that are not in their initial training set. Finally, on our second benchmark, WebNLG+, we verify that our diffusion modeling method also generalizes well to a completely new factual knowledge generation task beyond the commonsense domain.

2 Background: Diffusion Models

Diffusion models learn to construct synthetic data from random noise. They use a forward process to gradually corrupt real data samples with additive noise, and learn a reverse process to recover (or de-noise) the corrupted data samples. Through the de-noising of corrupted data, diffusion models learn to map from a random noise distribution to their target data distribution, which grounds their synthetic data generation.

In this paper, we adopt the DDPM[2] (Ho et al., 2020) formulation of the forward and reverse diffusion processes. Specifically, based on a sample $\mathbf{z}_0$ from a continuous input data distribution $q(\mathbf{z}_0)$, the forward process constructs noisy sample $\mathbf{z}_t$ over a sequence of time steps $t \in \{1, 2, ..., T\}$. In DDPM, $\mathbf{z}_t$ is sampled from a Gaussian distribution conditioned on the previous sample $\mathbf{z}_{t-1}$, given by:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

where β_t is a noise schedule hyperparameter unique to each diffusion step.

In the reverse process, diffusion models learn an inverse distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ to de-noise samples created by the forward process. To more precisely couple the intermediate states of the reverse process with the final de-noised sample $\mathbf{z}_0$, Diffusion-LM (Li et al., 2022) reformulates the task of predicting $\mathbf{z}_{t-1}$ as directly predicting $\mathbf{z}_0$ (based on $\mathbf{z}_t$), and uses a mean-squared error training loss on the $\mathbf{z}_0$ prediction at each time step[3]:

$$\mathcal{L}_{z_0\text{-}mse} = \sum_{t=1}^{T} \mathbb{E}\|\mathbf{z}_0 - f_\theta(\mathbf{z}_t, t)\|^2 \tag{2}$$

where $f_\theta(\mathbf{z}_t, t) = \hat{\mathbf{z}}_0^{t-1}$ denotes the model's learned prediction of $\mathbf{z}_0$ at the reverse stage of step t to $t-1$. To formulate $\hat{\mathbf{z}}_0^{t-1}$ as a refinement of the former reverse stage's output $\hat{\mathbf{z}}_0^t$, Bit-Diffusion (Chen et al., 2022) improves the model function of predicting $\mathbf{z}_0$ with self-conditioning, *i.e.*, $\hat{\mathbf{z}}_0^{t-1} = f_\theta(\hat{\mathbf{z}}_0^t, \mathbf{z}_t, t)$. At inference time, the noisy sample at step t is predicted from $\hat{\mathbf{z}}_0^t$ via the Eq. (1) forward process, denoted as $\hat{\mathbf{z}}_t$ to replace the unknown gold input $\mathbf{z}_t$, while the initial input $\mathbf{z}_T$ is pure Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

3 Contextual Knowledge Diffusion

In this section, we first introduce the task of contextual commonsense knowledge generation, and then propose DIFFUCOMET, our diffusion approach for this task. The overview of our method is presented in Figure 1.

Task Description Given a narrative sample $\mathcal{S}$ as context, *e.g.*, a story snippet or a dialogue, the model needs to generate commonsense inferences

[2] **D**enoising **D**iffusion **P**robabilistic **M**odels

[3] We include more detailed formulation of the reverse diffusion training in Appendix A.

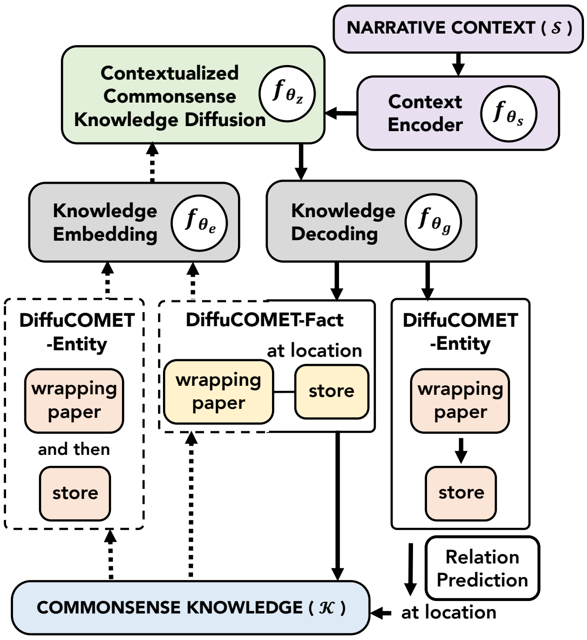


Figure 2: Knowledge diffusion based on facts or entities. Dashed arrows denote the forward process used for constructing gold references at the training phase. Solid arrows denote the reverse process used for generating knowledge with attention to the narrative context.

as a set facts $\mathcal{K} = \{k_1, ..., k_n, ..., k_N\}$, which are relevant for understanding the situation described in the context. Each fact $k_n = (h_n, r_n, a_n)$ is represented as a triple containing a head entity h_n, a tail (attribute) entity a_n, and a relation r_n connecting them, *e.g.*, (*wrapping paper*, *used for*, *wrap gifts*), as shown in Figure 1. We denote the set of unique head entities, relations and tail entities in $\mathcal{K}$ as $\mathcal{H}$, $\mathcal{R}$ and $\mathcal{A}$, respectively.

Contextualization We ground knowledge diffusion on the given context $\mathcal{S}$ by using encoder-decoder cross attention, inspired by SeqDiffuSeq (Yuan et al., 2022). In particular, we use a BART (Lewis et al., 2020) encoder f_{θ_s} to learn the context encoding that represents $\mathcal{S}$ as hidden state $\mathbf{z}_{\mathcal{S}}$:

$$\mathbf{z}_{\mathcal{S}} = f_{\theta_s}(\mathcal{S}) \tag{3}$$

Then, a BART decoder f_{θ_z}, serving as the diffusion module, learns to predict the de-noised data sample $\mathbf{z}_0$. Given the context hidden state $\mathbf{z}_{\mathcal{S}}$ (via cross-attention to the encoder f_{θ_s}), f_{θ_z} makes a prediction of $\mathbf{z}_0$ at time step t-1 (*i.e.*, $\hat{\mathbf{z}}_0^{t-1}$) based on its former prediction $\hat{\mathbf{z}}_0^t$ and time step t's noisy sample $\mathbf{z}_t$:

$$\hat{\mathbf{z}}_0^{t-1} = f_{\theta_z}(\hat{\mathbf{z}}_0^t, \mathbf{z}_t, t|\mathbf{z}_{\mathcal{S}}) \tag{4}$$

Unlike traditional transformer decoders (Vaswani et al., 2017), the diffusion module f_{θ_z} applies a bi-directional self-attention to $\hat{\mathbf{z}}_0^t$ and $\mathbf{z}_t$, since all positions of $\hat{\mathbf{z}}_0^{t-1}$ are decoded simultaneously, *i.e.*, in non-autoregressive manner.[4]

Discrete Knowledge Diffusion We consider two formulations for representing discrete knowledge in continuous embedding spaces for diffusion: **DIFFUCOMET-Fact**, where we learn to reconstruct continuous representations of facts k_n using diffusion, and **DIFFUCOMET-Entity**, where we use separate diffusion processes to reconstruct head h_n and tail a_n representations and then predict the relation between them to complete the fact. We highlight these differences in Figure 2.

For diffusion on the fact-level embedding space (**DIFFUCOMET-Fact**), we first pre-train a BART encoder f_{θ_e} to produce an embedding $\mathbf{e}_n$ of each fact k_n in the knowledge set $\mathcal{K}$ (with embedding size d same as the hidden state size of BART):

$$\mathbf{e}_n = f_{\theta_e}(k_n) \in \mathbb{R}^d \tag{5}$$

where we input the concatenation of each fact's head, relation and tail tokens to the encoder f_{θ_e}, and take the output hidden state of a start token <s> as the embedding of the fact. The initial input $\mathbf{z}_0$ of the forward diffusion process is then sampled from a Gaussian centered on the concatenation of all fact embeddings $\mathbf{e} = [\mathbf{e}_1; \mathbf{e}_2; ...; \mathbf{e}_{|\mathcal{K}|}] \in \mathbb{R}^{d \times |\mathcal{K}|}$, formulated as $q_e(\mathbf{z}_0|\mathbf{e}) = \mathcal{N}(\mathbf{z}_0; \mathbf{e}, \beta_0\mathbf{I})$.

In the reverse process, the diffusion module f_{θ_z} is trained to generate the final output $\hat{\mathbf{z}}_0^0$ (using time step 1's input $\mathbf{z}_1$ and $\hat{\mathbf{z}}_0^1$) as its predicted fact embeddings $\hat{\mathbf{e}}$, *i.e.*, $\hat{\mathbf{e}} = \hat{\mathbf{z}}_0^0 = f_{\theta_z}(\hat{\mathbf{z}}_0^1, \mathbf{z}_1, 1|\mathbf{z}_{\mathcal{S}})$. Finally, we pre-train another BART decoder f_{θ_g} to generate the synthetic fact $\hat{k}_n$ with conditioned on the diffusion module's predicted n-th embedding $\hat{\mathbf{e}}_n = \hat{\mathbf{e}}[:][n]$, $(n = 1, 2, ..., |\mathcal{K}|)$[5]:

$$\hat{k}_n = f_{\theta_g}(\hat{\mathbf{e}}_n) \tag{6}$$

For diffusion on the entity-level embedding space (**DIFFUCOMET-Entity**), we use a pipeline to generate head entities, tail entities and their relations. First, to generate head entities, we use a similar process as in **DIFFUCOMET-Fact**, *i.e.*, pre-train a BART encoder to produce a gold embedding

[4] More implementation details of the diffusion module f_{θ_z} are presented in Appendix B.1.

[5] At inference time, the maximum value of n (number of generated facts) can be arbitrary depending on the user's choice. In Appendix B.2, we introduce how we control the number of facts that our models generate for each context.

of each unique head entity $h_i \in \mathcal{H}$ (for training the diffusion module), and then pre-train a BART decoder to generate synthetic head entities $\hat{h}_i$ from the diffusion module's predicted embeddings. Each predicted head entity $\hat{h}_i$ is then appended to the context (*i.e.*, $\mathcal{S}$ in Eq. 3), expanding the context to $\mathcal{S}_i = [\mathcal{S}, \hat{h}_i]$. A second diffusion module predicts embeddings of synthetic tail entities $\hat{a}_j$ related to $\mathcal{S}_i$ (trained using gold embeddings of tail entities $a_j \in \mathcal{A}$ that possess relations $r_{ij} \in \mathcal{R}$ to the gold head h_i). A final BART model predicts the relation $\hat{r}_{ij}$ between each pair of generated head and tail entities, grounded on the context.

Embedding Module Training We pretrain the embedding modules (f_{θ_e}, f_{θ_g}), which focus on modeling generic knowledge representations independent to the context, before the diffusion modules (f_{θ_s}, f_{θ_z}), which learn the specific mapping from the context to its relevant knowledge. When training the diffusion modules, we freeze the pretrained embedding modules.

To pretrain the fact (or entity) embedding modules, we minimize the decoder's negative log-likelihood of re-constructing facts k (or entity h or a) in the full set of knowledge $\mathcal{K}_{full}$ involved in the whole narrative dataset (or domain), based on its embedding given by encoder f_{θ_e}:

$$\mathcal{L}_{\theta_e,\theta_g} = -\log p_{\theta_g}(k|f_{\theta_e}(k)) \tag{7}$$

Diffusion Module Training We optimize a dual loss to train the diffusion modules. First, we consider the mean-square error loss of the diffusion module's de-noised sample prediction $\hat{\mathbf{z}}_0^t$ at each time step t, compared to the reference sample $\mathbf{z}_0$ (for $t > 0$) and gold embeddings $\mathbf{e}$ (for $t = 0$):

$$\mathcal{L}^{mse}_{\theta_s,\theta_z} = \mathbb{E}\|\mathbf{e} - \hat{\mathbf{z}}_0^0\|^2 + \sum_{t=1}^{T-1} \mathbb{E}\|\mathbf{z}_0 - \hat{\mathbf{z}}_0^t\|^2 \tag{8}$$

We also use an anchor loss (Gao et al., 2022b) to supervise the final fact (or entity) generation. For each time step t, we minimize the negative log-likelihood of the embedding module decoder (with frozen parameters θ_g) generating each fact k_n in knowledge set $\mathcal{K}$, based on the diffusion module's predicted de-noised sample $\hat{\mathbf{z}}_0^t$:

$$\mathcal{L}^{gen}_{\theta_s,\theta_z} = \sum_{t=0}^{T-1} \sum_{n=1}^{|\mathcal{K}|} -\log p_{\theta_g}(k_n|\hat{\mathbf{z}}_0^t[:][n]) \tag{9}$$

where $\hat{\mathbf{z}}_0^t[:][n]$ is the predicted de-noised representation of k_n. The final loss is $\mathcal{L}_{\theta_s,\theta_z} = \mathcal{L}^{mse}_{\theta_s,\theta_z} + \gamma\mathcal{L}^{gen}_{\theta_s,\theta_z}$, where γ is a tunable hyperparameter.

Inference At inference time, the reverse diffusion process is initialized with noise sampled from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, while the embedding module encoder f_{θ_e}, which provides gold diffusion references for training, is not used.

4 Evaluation

Prior work in commonsense knowledge generation (Hwang et al., 2021; Da et al., 2021) evaluated knowledge models using traditional NLG metrics (*e.g.*, BLEU; Papineni et al., 2002) in controlled studies with KGs, where the inputs to the models were head entities and relations and the knowledge model produced tail attributes. In practice, however, knowledge models are used to generate implicit commonsense inferences for natural language contexts (Ismayilzada and Bosselut, 2023), requiring generated inferences to be relevant to a more complex input than a basic KG head entity, and necessitating diverse generated inferences that comprehensively augment the context. However, traditional NLG metrics fall short of measuring these important dimensions because they measure surface form overlap between model outputs and references, which rewards generating facts with similar or duplicated semantics, limiting diversity.

Motivated by these shortcomings, we propose novel evaluation metrics that assess the diversity and contextual relevance of generated knowledge. First, to eliminate the effect of knowledge repetition in generations, we cluster similar facts and treat each fact cluster (instead of each single fact) as a unit piece of knowledge. In particular, we use the DBSCAN (Ester et al., 1996) algorithm to group gold facts $\mathcal{K} = \{k_1, k_2, ..., k_N\}$ and generated facts $\hat{\mathcal{K}} = \{\hat{k}_1, \hat{k}_2, ..., \hat{k}_{\hat{N}}\}$ into clusters $\mathcal{C} = \{c_1, c_2, ..., c_M\}$ and $\hat{\mathcal{C}} = \{\hat{c}_1, \hat{c}_2, ..., \hat{c}_{\hat{M}}\}$, respectively. We test two methods for measuring the similarity of facts for clustering: word-level edit distance (Levenshtein et al., 1966), which measures the difference of two facts' surface-form tokens, and Euclidean distance of Sentence-BERT (Reimers and Gurevych, 2019) embeddings, which measures the semantic difference of two facts. Based on these clusters, we develop three metrics to measure the diversity of generated facts, their contextual relevance, and their alignment to gold references, as shown in Figure 3.

Diversity. To measure the diversity of generated facts (*i.e.*, amount of distinctive knowledge being generated), we count the number of fact clusters

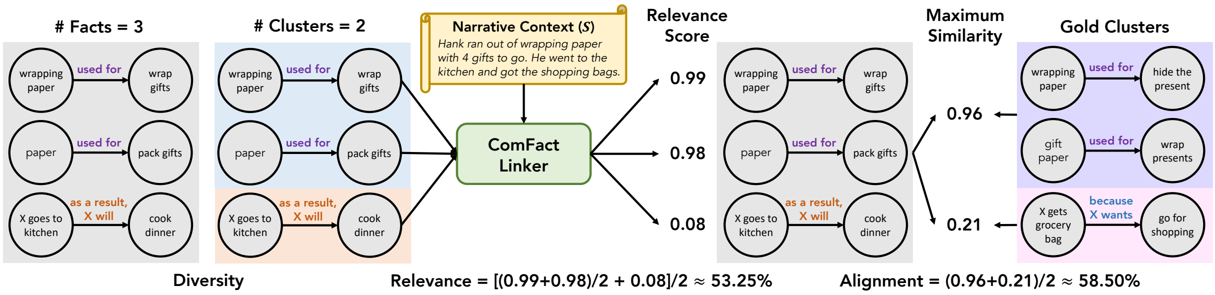


Figure 3: Illustration of clustering-based evaluation metrics for contextual commonsense knowledge generation.

(**# Clusters**), *i.e.*, $\hat{M}$ (or M for gold references). We also report the number of facts (**# Facts**), *i.e.*, $\hat{N}$ (or N for gold references), to compare the number of fact clusters to the number of generated facts produced by the models.

Relevance. We measure the relevance of the fact clusters to the narrative context using a fact linker[6] trained on the $\mathcal{C}om\mathcal{F}act$ dataset (Gao et al., 2022a) that scores the relevance of each fact $\hat{k}_n$ to the context $\mathcal{S}$, denoted as $rel(\hat{k}_n, \mathcal{S}) \in [0, 1]$. The relevance score of a fact cluster $\hat{c}_m$ is defined as the average relevance score of its facts, *i.e.*, $\sum_{\hat{k}_n \in \hat{c}_m} rel(\hat{k}_n, \mathcal{S})/|\hat{c}_m|$. Finally, we measure the average relevance over all fact clusters in $\hat{\mathcal{C}}$:

$$rel(\hat{\mathcal{C}}, \mathcal{S}) = \frac{1}{\hat{M}} \sum_{\hat{c}_m \in \hat{\mathcal{C}}} \frac{1}{|\hat{c}_m|} \sum_{\hat{k}_n \in \hat{c}_m} rel(\hat{k}_n, \mathcal{S}) \tag{10}$$

We note that **Relevance** can be viewed as a *precision* measure for generated facts, which tends to decrease as more facts are generated because irrelevant facts are more likely to be generated.

Alignment measures the average similarity of generated facts to gold fact clusters. Specifically, we define a function $sim(\hat{k}_i, k_j) \in [0, 1]$ to measure the pairwise similarity between a generated fact and a gold reference (using similar distance functions to define clusters above[7]). Using this function, we measure the maximum pairwise similarity of generated facts to references in each gold cluster $c_m \in \mathcal{C}$, which serves as the alignment score to the gold cluster. Finally, we average the alignment scores of generated facts to all gold clusters:

$$sim(\hat{\mathcal{K}}, \mathcal{C}) = \frac{1}{M} \sum_{c_m \in \mathcal{C}} \max_{\substack{\hat{k}_i \in \hat{\mathcal{K}}, \\ k_j \in c_m}} sim(\hat{k}_i, k_j) \tag{11}$$

We note that **Alignment** can be viewed as the generated facts' *recall* of gold fact clusters, which tends to increase as more facts are generated because more facts will be aligned to gold clusters. Given this trade-off between Relevance and Alignment, we also present the harmonic mean of Relevance and Alignment as an overall evaluation of the two dimensions, denoted as **RA-F1**.

5 Experimental Settings

Datasets First, we evaluate our approach on the $\mathcal{C}om\mathcal{F}act$ (Gao et al., 2022a) benchmark, where models need to generate ATOMIC$^{20}_{20}$ (Hwang et al., 2021) social commonsense facts that are relevant to narrative contexts sampled from four diverse corpora: PERSONA-CHAT (Zhang et al., 2018), MuTual (Cui et al., 2020), ROCStories (Mostafazadeh et al., 2016) and CMU MovieSummaries (Bamman et al., 2013). We only use training data from the ROCStories portion of $\mathcal{C}om\mathcal{F}act$, to enable the evaluation of zero-shot generalization on the other three partitions of the dataset. Our fact embedding module is pretrained on the full ATOMIC$^{20}_{20}$ knowledge base, which contains $\sim 972K$ commonsense facts after preprocessing.[8] We also evaluate our approach in a conceptually different setting, the WebNLG+ 2020 (Ferreira et al., 2020) dataset, which consists of RDF (Ora, 1999) facts sampled from the DBpedia (Lehmann et al., 2015) knowledge base, with corresponding natural language texts verbalizations. The task is to generate the sampled RDF facts given their verbalizations. We use ∼13k facts from the training data to pretrain the fact embedding module.

Baselines We train DIFFUCOMET using BART-base and BART-large as pretrained models, and compare with three baselines developed on the same backbones: a) a **Greedy** baseline that is

[6] Fact linking models predict the relevance of knowledge tuples to textual passages (Gao et al., 2022a)

[7] Further details on exact definitions are in Appendix C.1.

[8] More data preprocessing details are in Appendix D.

Model	# Facts	# Clusters	Relevance	Alignment	RA-F1	BLEU	METEOR	ROUGE-L
Greedy-COMET	1.96	1.19	61.42	50.64	55.51	**18.01**	**52.32**	**54.96**
Sampling-COMET	15.00	**8.39**	56.19	**77.97**	65.31	12.69	44.43	45.58
Beam-BART	15.00	4.60	64.35	71.35	67.67	13.11	47.70	46.35
Beam-COMET	15.00	5.09	65.03	71.64	68.18	16.97	47.39	47.19
Grapher	5.08	2.60	**68.29**	40.58	50.91	1.40	23.96	27.21
DIFFUCOMET-Fact	12.88	5.24	65.64	71.65	68.51	15.98	50.06	51.44
DIFFUCOMET-Entity	12.89	5.67	66.39	74.38	**70.16**	17.01	47.61	48.40
Gold	10.55	5.64	80.90	-	-	-	-	-

Table 1: Evaluation results on the ROCStories portion of *ComFact*. Both DIFFUCOMET models presented are developed based on BART-large. Models with suffix "-COMET" and "-BART" are fine-tuned on COMET-BART and BART-large. Presented results of our proposed metrics are based on fact clustering *w.r.t.* embedding Euclidean distance. Best and second-best results (excluding Gold references) are **bolded** and underlined, respectively.

trained to autoregressively generate the concatenation of all relevant facts,[9] b) a **Sampling** baseline that uses nucleus sampling (Holtzman et al., 2019) to generate multiple individual facts in parallel, and c) a Diverse **Beam** search baseline that uses diverse beam search to generate multiple inferences in parallel. We also compare our models trained using BART-large to baselines developed on models of similar scale: d) the aforementioned greedy decoding, sampling and beam search baselines trained from **COMET**-BART (Hwang et al., 2021), a BART-large model further pre-trained on ATOMIC$^{20}_{20}$ for commonsense knowledge completion, and e) **Grapher** (Melnyk et al., 2022), which trains a T5-large (Raffel et al., 2020) model to generate entities (nodes) related to the context, followed by a MLP classifier to predict the relations (edges) between entities.

Metrics We evaluate these methods on our clustering-based metrics described in Section 4. As the clustering algorithm (*i.e.*, DBSCAN) used in our metrics has an adjustable clustering granularity controlled by a distance threshold, we consider a range of distance thresholds and take the average of evaluation results across all thresholds in the range, allowing us to avoid biasing our metrics to a specific distance threshold.[10] For *ComFact*, we also test on the metrics from Hwang et al., 2021 for evaluating commonsense knowledge generation, including **BLEU** (Papineni et al., 2002), **METEOR** (Banerjee and Lavie, 2005) and **ROUGE-L** (Lin, 2004). For evaluation on WebNLG+ 2020, we also report the official metrics for this dataset's challenge (Ferreira et al., 2020), which construct optimal pairings between predicted facts and gold references, and then compute precision, recall, and F1 scores based on the surface-form matching of paired facts. We denote these WebNLG metrics as **Web-Prec.**, **Web-Rec.** and **Web-F1**.[11]

6 Results and Analysis

Table 1 shows evaluation results on the ROCStories portion of the *ComFact* benchmark for our DIFFUCOMET models developed based on BART-large.[12] On our new cluster-based metrics, DIFFUCOMET models demonstrate a better balance between diversity and accuracy in contextual knowledge generation. Specifically, DIFFUCOMET models achieve Relevance and Alignment scores that are both comparable to the best baseline results, contributing to their higher overall RA-F1 measures, while also producing a larger number of distinct knowledge clusters. By contrast, the Greedy, Sampling and Grapher baselines significantly sacrifice one or two dimensions of diversity and quality *w.r.t.* # Clusters, Relevance and Alignment. Beam baselines consistently underperform DIFFUCOMET on cluster metrics.

For evaluation on the traditional NLG metrics, we find that DIFFUCOMET models score higher overall than most baseline models on metrics that check the alignment with gold references, *i.e.*, BLEU, METEOR and ROUGE-L, except for the Greedy decoding baseline, whose higher scores are artificially high because it generates very little knowledge, *i.e.*, only ∼2 facts per context. We also include further comparisons of models with

[9] Facts are concatenated by a special token <*fsep*>.

[10] We include more details of our clustering threshold selection in Appendix C.2.

[11] More details of WebNLG metrics are in Appendix C.3.

[12] Presented results of our metrics are based on fact clustering *w.r.t.* embedding Euclidean distance. Results based on word-level edit distance are included in Appendix F.1, and promote the same conclusions.

Model	**PersonaChat**			**MuTual**			**MovieSummaries**		
	# Facts	**# Clusters**	**RA-F1**	**# Facts**	**# Clusters**	**RA-F1**	**# Facts**	**# Clusters**	**RA-F1**
Beam-BART	15.00	4.04	58.27	15.00	4.45	64.77	15.00	3.70	46.15
Beam-COMET	15.00	4.27	59.04	15.00	4.75	64.82	15.00	4.04	45.88
Grapher	4.53	1.57	41.24	4.50	1.78	54.31	5.34	1.50	41.97
DIFFUCOMET-Fact	10.82	3.89	59.68	10.46	4.80	66.04	8.29	2.89	43.51
DIFFUCOMET-Entity	12.06	**4.48**	**60.50**	11.85	**5.39**	**67.32**	13.50	**5.84**	**50.42**
Gold	8.60	4.28	-	10.80	5.79	-	9.00	4.81	-

Table 2: Evaluation results on out-of-domain contexts from the PersonaChat, MuTual and MovieSummaries portions of $\mathcal{C}om\mathcal{F}act$ benchmark. Models with suffix "-COMET" and "-BART" are fine-tuned on COMET-BART and BART-large. Presented results of our proposed metrics are based on fact clustering *w.r.t.* embedding Euclidean distance. Best and second-best results (excluding Gold references) are **bolded** and underlined, respectively.

Model	**Validity**	**Relevance**
Sampling-COMET	49.45	30.20
Beam-COMET	**74.80**	42.81
DIFFUCOMET-Fact	70.00	48.27
DIFFUCOMET-Entity	74.15	**54.18**
Gold	94.79	82.04

Table 3: Human evaluation results. Best and second-best results (excluding Gold references) are **bolded** and underlined, respectively.

Model	**# Novel Facts**	**# Novel Clusters**
Sampling-COMET	0.26	0.19
Beam-COMET	0.27	0.17
DIFFUCOMET-Fact	**0.30**	0.20
DIFFUCOMET-Entity	**0.30**	**0.24**

Table 4: Novelty of generated knowledge. Best and second-best results are **bolded** and underlined, respectively.

BART-*base* backbones in Appendix F.1, where our models outperform baselines by a larger gap, *i.e.*, $\sim$15% absolute RA-F1 improvement on average.

We also test DIFFUCOMET's ability to generalize to out-of-domain contexts using the other portions of $\mathcal{C}om\mathcal{F}act$ with contexts sampled from PersonaChat, MuTual and MovieSummaries. We report main generalization results to the above three portions in Table 2, and observe similar results where DIFFUCOMET-Entity outperforms baselines by $\sim$5% RA-F1 and produces $\sim$20% more knowledge clusters.[13] Interestingly, DIFFUCOMET-Fact struggles to outperform beam search baselines on the longer context narratives from MovieSummaries, showing that entity-level diffusion is more robust to the shift of narrative contexts, likely due to the more fine-grained multi-step learning of context-to-knowledge mapping.

The results of our automatic evaluation are also supported by our human evaluation. We hire Amazon Mechanical Turk workers[14] to evaluate the validity and contextual relevance of models' generated knowledge on the ROCStories portion of $\mathcal{C}om\mathcal{F}act$. Specifically, given a narrative context and a list of commonsense facts that a model generates about the context, we ask three workers to independently judge whether each fact is valid (*i.e.*, plausible and natural-sounding) and relevant[15] to the context, and take their majority vote as the assessment. In Table 3, we see that DIFFUCOMET models produce *valid* facts at about the same rate as the best baseline, but produce facts that are far more relevant to the narrative context.

Novelty DIFFUCOMET models also produce more novel commonsense inferences. A historical advantage of knowledge models (*e.g.*, COMET) was their ability to generate knowledge beyond the graphs they used for pretraining (Bosselut et al., 2019), making them powerful tools to generate commonsense knowledge for unseen narratives. To test the novelty of generated commonsense knowledge from DIFFUCOMET, we develop a heuristic method that identifies knowledge as *novel* if its maximum pair-wise (Sentence-BERT embedding) cosine similarity to $\mathcal{C}om\mathcal{F}act$ gold references is lower than 0.45. However, as this cut-off would likely cause invalid and irrelevant facts to be considered novel, we only include facts whose relevance score is higher than 0.97.[16] In Table 4, we see that

[13] Full evaluation results on out-of-domain contexts are reported in Appendix F.2.

[14] Details on workers and their payment are in Appendix E

[15] *invalid* facts are automatically labeled *irrelevant*

[16] Thresholds are tuned by a manual check of 100 sampled results to ensure a decent cutoff of novel and relevant facts.

Narrative Context	Jordan decides he must find something to occupy his time. He decides to try playing a game of solitaire. He draws good cards initially, but quits the game before it is over. Jordan shuffles the cards and returns them to a drawer. He is happy that he found something to take up a little time.
Beam -COMET	something, used for, occupy the time solitaire game, used for, play with drawer, used for, store the cards
DIFFUCOMET -Entity	X finds something to do, because X wants, to be entertained solitaire game, used for, get some good time X plays solitaire, but before, X needs to have a deck of cards X draws cards, because X wants, to play cards drawer, used for, put the cards in

Table 5: Comparison of novel knowledge generated by Beam-COMET and DIFFUCOMET-Entity.

DIFFUCOMET models produce more novel facts and clusters compared to baselines.

This pattern is observed in our qualitative analysis too. In Table 5, we provide examples of novel facts generated by Beam-COMET and DIFFUCOMET-Entity for an example narrative. We find that our model produces more diverse novel facts covering both physical entities (*e.g.*, solitaire game) and complex actions (*e.g.*, drawing cards to play solitaire). By comparison, Beam-COMET's novelty is mainly restricted to physical knowledge.[17]

Diffusion Steps To investigate how DIFFUCOMET's multiple rounds of knowledge representation refinement through the diffusion process affect the quality of generated knowledge, we record the performance of our DIFFUCOMET models as a function of diffusion steps conducted during inference. Figure 4 shows how DIFFUCOMET's performance varies when knowledge is generated at earlier time steps.

We find that DIFFUCOMET models gradually produce more facts and more diverse facts (*i.e.*, # Clusters) as the number of diffusion steps increase, indicating that the multiple rounds of diffusion produce a more separable representation capable of representing more facts. While the greater number of facts leads to a slight drop in contextual relevance across the generated facts, a greater corresponding increase in alignment to the gold clusters (as observed by the increase in Alignment and RA-F1) also emerges. On RA-F1, DIFFUCOMET-Fact surpasses Beam-COMET[18] as the diffusion steps increase to larger than 200, and DIFFUCOMET-Entity consistently scores higher and continues benefiting from further diffusion, even after 1000 diffusion steps. These results shows that multi-step refinement of facts via diffusion effectively improves contextual knowledge generation.

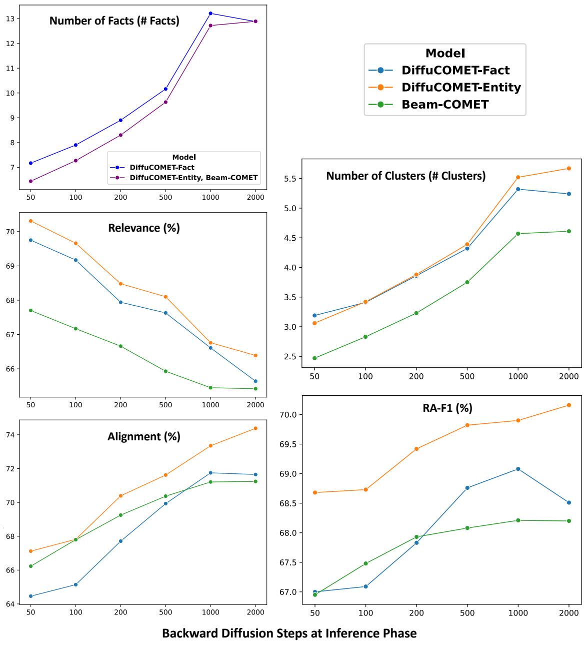


Figure 4: DIFFUCOMET performance at different diffusion steps during inference. Both DIFFUCOMET-Fact and DIFFUCOMET-Entity are developed based on BART-large and tested on the ROCStories portion of $\mathcal{C}om\mathcal{F}act$. Beam-COMET performance is shown as a baseline, with the number of decoded facts set to match DIFFUCOMET-Entity at each diffusion step.

Model	**Web-Prec.**	**Web-Rec.**	**Web-F1**
Beam-BART	73.36	76.27	74.75
Grapher	71.20	73.00	71.90
DIFFUCOMET-Fact	76.30	78.07	77.19
DIFFUCOMET-Entity	**80.68**	**82.89**	**81.74**

Table 6: Results on **WebNLG+ 2020**. Official metrics used for the benchmark challenge are presented.

6.1 WebNLG+ 2020 Benchmark

Finally, to test whether our method generalizes outside the domain of generating commonsense inferences, we present our evaluation results on the WebNLG+ 2020 dataset in Table 6. DIFFUCOMET models achieve better performances on the WebNLG factual knowledge generation task, verified by the official metrics of the benchmark.[19] This results suggests that our diffusion approach to knowledge graph construction could be adapted to other knowledge generation tasks.

[17] We provide more analysis of example generated facts in Appendix F.3.

[18] To make the comparison intuitive, for each test context, we dynamically set the beam size of Beam-COMET to the number of facts generated by DIFFUCOMET-Entity.

[19] We also include the evaluation results on traditional NLG and our proposed clustering-based metrics in Appendix F.4.

7 Related Work

Commonsense Knowledge Grounding To augment NLP systems with commonsense knowledge, various systems for question answering (Zhang et al., 2022; Yasunaga et al., 2021, 2022) and narrative generation (Ji et al., 2020; Zhou et al., 2022) use retrieval methods based on heuristics to link relevant facts from commonsense knowledge graphs (Speer et al., 2017; Sap et al., 2019; Gao et al., 2023). However, these systems typically have low precision when adapted to more general and complex commonsense linking (Hwang et al., 2021; Jiang et al., 2021). Gao et al., 2022a developed commonsense fact linking to improve retrieval precision, but this requires inefficiently traversing all candidate facts to check their contextual relevance.

Due to above limitations of retrieval-based knowledge grounding, one line of research (Bosselut et al., 2021; Tu et al., 2022) uses knowledge models (Bosselut et al., 2019; West et al., 2022) to generate tail inferences from narrative statements. However, these methods often produce irrelevant facts as the knowledge models are pre-trained for context-free knowledge graph completion. Finally, developing new knowledge models to learn contextual commonsense generation turns out to be a promising track of research, while current works are limited to simple physical (Zhou et al., 2022) or RDF-style factual (Melnyk et al., 2022) knowledge. We build new models to address contextual commonsense generation in a more general scope.

Diffusion Models Considerable recent works (Gao et al., 2022b; Lin et al., 2022; Han et al., 2024) have developed methods to improve text generation with diffusion models (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020). However, the potential of diffusion models in text-to-knowledge generation is still under-explored. In this paper, we introduce diffusion models for the task of contextual knowledge generation.

8 Conclusion

In this work, we leverage the power of diffusion models for contextual commonsense knowledge generation, and formulate novel metrics to highlight important dimensions of diversity and contextual relevance for this task. Our diffusion knowledge models, DIFFUCOMET, outperform various autoregressive knowledge models, producing more diverse, novel, and contextually-relevant commonsense knowledge, and achieving better out-of-distribution performance. Finally, our analysis reveals how DIFFUCOMET refines implicit knowledge representations over the course of the diffusion process to produce more relevant and diverse inferences, hinting at our method's potential benefit in other text-to-graph generation tasks.

Limitations

We outline the following limitations in our work. First, narrative samples in our training datasets, *i.e.*, $\mathcal{C}om\mathcal{F}act$ (Gao et al., 2022a) and WebNLG+ 2020 (Ferreira et al., 2020), have short context windows (five sentences at maximum). Therefore, our knowledge models trained on these datasets may have limited inference capacities if applied to longer narratives that involve richer commonsense grounding. Moreover, our models are trained on solely English corpora, and may need additional resources to be adapted to other languages or multilingual settings. Finally, our diffusion modeling method is tested on an encoder-decoder model structure, *i.e.*, BART (Lewis et al., 2020), with maximum model size 406M (BART-large). We leave the feasibility of our method on other model structures, *e.g.* decoder-only GPT (Radford et al., 2019), and larger model scales, to future work.

References

David Bamman, Brendan O'Connor, and Noah A Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361.

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.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 35, pages 4923–4931.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. 2022. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*.

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

Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. 2021. Analyzing commonsense emergence in few-shot knowledge models. *arXiv preprint arXiv:2101.00297*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge discovery and aata mining*, volume 96, pages 226–231.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris Van Der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*.

Silin Gao, Beatriz Borges, Soyoung Oh, Deniz Bayazit, Saya Kanno, Hiromi Wakaki, Yuki Mitsufuji, and Antoine Bosselut. 2023. Peacok: Persona commonsense knowledge for consistent and engaging narratives. *arXiv preprint arXiv:2305.02364*.

Silin Gao, Jena D Hwang, Saya Kanno, Hiromi Wakaki, Yuki Mitsufuji, and Antoine Bosselut. 2022a. Comfact: A benchmark for linking contextual commonsense knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1656–1675.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2022b. Difformer: Empowering diffusion model on embedding space for text generation. *arXiv preprint arXiv:2212.09412*.

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.

Kehang Han, Kathleen Kenealy, Aditya Barua, Noah Fiedel, and Noah Constant. 2024. Transfer learning for text diffusion models. *arXiv preprint arXiv:2401.17181*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Mete Ismayilzada and Antoine Bosselut. 2023. kogito: A commonsense knowledge inference toolkit. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 96–104.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736.

Liwei Jiang, Antoine Bosselut, Chandra Bhagavatula, and Yejin Choi. 2021. "i'm not mad": Commonsense implications of negation and contradiction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4380–4397.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

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

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Weizhu Chen, and Nan Duan. 2022. Genie: Large scale pre-training for text generation with diffusion model. *arXiv preprint arXiv:2212.11685*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Igor Melnyk, Pierre Dognin, and Payel Das. 2022. Knowledge graph generation from text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1610–1622.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Lassila Ora. 1999. Resource description framework (rdf) model and syntax specification. *http://www. w3. org/TR/REC-rdf-syntax/*.

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.

Xiangyu Peng, Siyan Li, Sarah Wiegreffe, and Mark Riedl. 2022. Inferring the reader: Guiding automated story generation with commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7008–7029.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21:5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

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

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Quan Tu, Yanran Li, Jianwei Cui, Bin Wang, Ji-Rong Wen, and Rui Yan. 2022. Misc: A mixed strategy-aware model integrating comet for emotional support conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 308–319.

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

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*.

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

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. In *Proceedings of the 10th International Conference for Learning Representations (ICLR)*.

Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2022. Think before you speak: Explicitly generating implicit commonsense knowledge for response generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1252.

A Backward Diffusion Process

Inverting from the forward diffusion process formulated as Equation (1), the backward diffusion process follows a Gaussian posterior distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)$:

$$\begin{aligned} q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) &= \mathcal{N}(\mathbf{z}_{t-1}; \widetilde{\mu}(\mathbf{z}_t, \mathbf{z}_0), \widetilde{\beta}_t \mathbf{I}) \\ \widetilde{\mu}(\mathbf{z}_t, \mathbf{z}_0) &= \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}\mathbf{z}_0 + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\mathbf{z}_t \\ \widetilde{\beta}_t &= \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t \end{aligned} \tag{12}$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ are weight hyperparameters of the posterior Gaussian defined by the noise schedule β_t. The posterior formulation indicates that only the mean $\widetilde{\mu}$ of $\mathbf{z}_{t-1}$ is correlated to the condition $\mathbf{z}_t$ and $\mathbf{z}_0$. So the training loss for diffusion models, derived from the KL-divergence between gold and learned posterior distributions, is typically defined as a mean-squared error loss on the posterior Gaussian mean:

$$\mathcal{L}_{mse} = \sum_{t=1}^{T} \mathbb{E}\|\widetilde{\mu}(\mathbf{z}_t, \mathbf{z}_0) - \mu_\theta(\mathbf{z}_t, t)\|^2 \tag{13}$$

where model (with parameter θ) learns the function $\mu_\theta(\mathbf{z}_t, t)$ to predict the mean of $\mathbf{z}_{t-1}$. Diffusion-LM (Li et al., 2022) further re-weights the mean-squared error as Eq. 2 to enforce direct prediction of $\mathbf{z}_0$ in every loss term, which is shown to be more efficient at tuning the model to precisely predict the final de-noised sample.

B Model Implementation Details

B.1 Diffusion Module

To conduct the diffusion process defined by Eq.(4) using Transformers (Vaswani et al., 2017), $\mathbf{z}_t$ and $\hat{\mathbf{z}}_0^t$ are first concatenated at the hidden-state dimension and projected by a MLP layer to form their joint representation. The positional encoding layer of Transformers is applied to the time step t (same for every position of self-attention), whose output time step embedding is added to the joint representation of $\mathbf{z}_t$ and $\hat{\mathbf{z}}_0^t$. The decoder f_{θ_z} takes the joint representation (with time step embedding added) as its bi-directional self-attention input, to ground its decoding of refined $\mathbf{z}_0$ prediction $\hat{\mathbf{z}}_0^{t-1}$.

B.2 Number of Generated Facts

To enable our diffusion module (f_{θ_z}) to control the number of facts (or entities) generated for each context, we also pre-train our fact (or entity) embedding module (f_{θ_e} and f_{θ_g}) to learn the representation of a special token k_{end} :=<*eok*>, by adding it as a special fact (or entity) to the pre-training data, which indicates the end of a knowledge set. During the training of diffusion module, k_{end} is appended to the end of knowledge set $\mathcal{K}$, whose embedding and decoding also contributes to the training loss. At inference phase, we post-process our model's generations to keep only the facts that are at positions before k_{end}.

B.3 Noise Schedule

For the noise schedule hyperparameter of diffusion process, we adopt the *sqrt* initialization (Li et al., 2022) to set $\overline{\alpha}_t = 1 - \sqrt{t/T + s}$, where $s = 1e^{-4}$ that sets the initial variance of noise (β_0) to be 0.01. Based on that, we follow SeqDiffuSeq (Yuan et al., 2022) to implement an adaptive noise schedule, which dynamically adjusts $\overline{\alpha}_t$ for each sample position n ($n = 1, 2, ...|\mathcal{K}|$) of the knowledge set $\mathcal{K}$ (the adjusted $\overline{\alpha}_t$ for position n is denoted as $\overline{\alpha}_t^n$), according to the diffusion mean square error (MSE) loss $\mathcal{L}_{\theta_s,\theta_z}^{mse}$ defined in Eq. (8). Specifically, for an adaptive noise schedule update, we first record the MSE loss at each time t and position n as:

$$\mathcal{L}_t^n = \mathbb{E}\|\mathbf{z}_0[:][n] - \hat{\mathbf{z}}_0^t[:][n]\|^2 \tag{14}$$

Then we use a linear interpolation function to update the adjusted noise schedule, formulated as:

$$F_t^n(\mathcal{L}) = \frac{\overline{\alpha}_t^n - \overline{\alpha}_{t-1}^n}{\mathcal{L}_t^n - \mathcal{L}_{t-1}^n}(\mathcal{L} - \mathcal{L}_{t-1}^n) + \overline{\alpha}_{t-1}^n \tag{15}$$

where new loss value $\mathcal{L}_t^{n,new}$ is re-arranged across time step t with equal interval between $\min_t(\mathcal{L}_t^n)$ and $\max_t(\mathcal{L}_t^n)$, which is finally given to the update function to get $\overline{\alpha}_t^{n,new} = F_t^n(\mathcal{L}_t^{n,new})$. The noise schedule is adjusted every 2000 training steps.

B.4 Model Training

For the loss weight hyperparameter γ used to combine mean-square error and anchor losses defined by Eq. (8) and (9), we use $\gamma = 1$ for training our DIFFUCOMET models based on BART-base, while $\gamma = 0.01$ for training our models with BART-large backbone, which achieve the best convergence results, respectively. For training DIFFUCOMET based on BART-large, we also follow Differmer (Gao et al., 2022b) to amplify the standard deviation of diffusion noise by a factor of $A = 4$, *i.e.*, to change the forward process as:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t A^2\mathbf{I}) \quad (16)$$

where $t = 1, 2, ...T$, which effectively avoids model collapse in training. The total diffusion steps T is set to 2000. We use AdamW (Loshchilov and Hutter, 2018) as our training optimizer, with learning rate $1e^{-5}$ and no weight decay. A linear learning rate scheduler is adopted with warm-up steps 2000 and total training steps 150000 and 200000 for models based on BART-base (139M) and BART-large (406M), respectively. We train our base-scale DIFFUCOMET on 4 Tesla V100-SXM2 (32GB) GPUs with batch size set to 4, while for large-scale DIFFUCOMET, we use 4 NVIDIA A100-SXM4 (40GB) GPUs, with batch size set to 2 instead. 15 and 36 hours are required to train base-scale and large-scale DIFFUCOMET models, respectively.

For the pre-training of our fact embedding module with loss described in Eq. (7), we adopt the same hyperparameter setting as training our diffusion module, except for learning rate changed to $2e^{-6}$ and batch size set to 128 and 64 for base-scale and large-scale models, respectively. For pre-training large-scale (*i.e.*, BART-large) fact embedding module, we add a weight decay of 0.01, which leads to better convergence. In DIFFUCOMET-Entity, the two diffusion modules trained for generating contextual relevant head and tail entities share the same pre-trained entity embedding module.

C Evaluation Metrics

C.1 Clustering and Similarity Function

For our evaluation based on fact clustering *w.r.t.* edit distance, we define the similarity function in our Alignment metric as $sim(\hat{k}_i, k_j) = 1 - Edit(\hat{k}_i, k_j)/MaxLen(\hat{k}_i, k_j)$, where $Edit$ denotes the word-level edit distance of two facts, and $MaxLen$ denotes the length of the longer fact of the two, *i.e.*, the maximum possible edit distance for normalization. Our distance measure for clustering also adopts the normalized edit distance, *i.e.*, $Edit/MaxLen$. For evaluation based on fact clustering *w.r.t.* Sentence-BERT embedding, we define the similarity function in our Alignment metric as $sim(\hat{k}_i, k_j) = max(CoS(\hat{k}_i, k_j), 0)$, where CoS denotes the cosine similarity of two facts' Sentence-BERT embeddings. We assume that facts with opposite meanings, *i.e.*, negative similarity, are not considered as aligned with each other, so we cut off the negative values of cosine similarity. While for the distance measure of clustering, we use the Euclidean distance of two facts' embeddings instead, which is typically adopted in DBSCAN (Ester et al., 1996) clustering algorithm.

C.2 Clustering Threshold Selection

For our proposed clustering-based metrics as described in Section 4, we use DBSCAN (Ester et al., 1996) algorithm to group facts into clusters. To avoid bias on a specific clustering granularity, we consider a range of DBSCAN thresholds and take the average evaluation results across all thresholds in the range. We consider a range with equal interval of 0.05, where the number of gold fact clusters significantly changes from near the maximum (*i.e.*, each fact as a cluster) to near the minimum (*i.e.*, all facts grouped into one cluster). Figure 5 shows the number of gold clusters as a function of the DBSCAN clustering threshold, and our selection of threshold ranges (red square) on each dataset.

C.3 WebNLG Metrics

In the evaluation of WebNLG 2020 Challenge (Ferreira et al., 2020), each generated RDF fact (*i.e.*, subject-predicate-object triple) is paired to a gold reference to compute its precision, recall and F1 based on named entity matching (Segura-Bedmar et al., 2013). Three types of matching criterias are considered, including: a) each named entity in generated RDF needs to exactly match an entity in gold reference in order to be counted as true-positive,

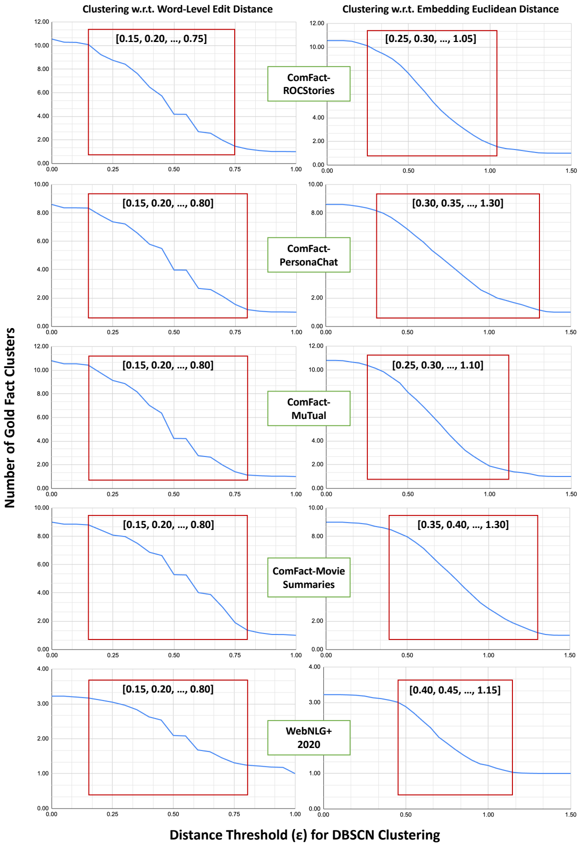


Figure 5: Range selection (red square) of DBSCN clustering thresholds for our proposed metrics.

while its type in the RDF (*i.e.*, whether it is in subject, predicate or object) does not need to match (**Exact Match**), b) each entity in generated RDF only needs to partially match an entity in gold reference, and its type does not matter (**Partial Match**), and c) each named entity in generated RDF needs to exactly match an entity in gold reference, and its type also needs to match (**Strict Match**). For each matching criteria, optimal pairing (with the highest F1 score) between generated facts and gold references is searched by enumerating all possible permutations. We report Strict Match scores in the main body of our paper in Table 6, and include all three kinds of match scores in Table 20.

D Data Preprocessing

The $\mathcal{C}om\mathcal{F}act$ (Gao et al., 2022a) benchmark contains social commonsense knowledge linked from the ATOMIC$^{20}_{20}$ (Hwang et al., 2021) knowledge graph, which contains ∼1.33M facts covering physical entities, daily events and social interactions. ATOMIC$^{20}_{20}$ commonsense relations considered in our experiments are listed in Table 7. We preprocess $\mathcal{C}om\mathcal{F}act$ and ATOMIC$^{20}_{20}$ to filter out facts that have invalid tail entity "none" or contain fillable blank "___", *i.e.*, we do not consider facts with relation "IsFilledBy". After preprocessing, ∼972K facts are involved in the training of our fact embedding and diffusion modules. The original $\mathcal{C}om\mathcal{F}act$ training data in the ROCStories portion only has $\sim 1K$ contexts with gold annotations of relevant facts. Due to the limited supervised data, we augment the training data with $\sim 50K$ additional contexts sampled from the ROCStories corpus, and use a DeBERTa (He et al., 2020) fact linker developed from the $\mathcal{C}om\mathcal{F}act$ benchmark to extract silver annotations of relevant facts from ATOMIC$^{20}_{20}$ to each additional context.

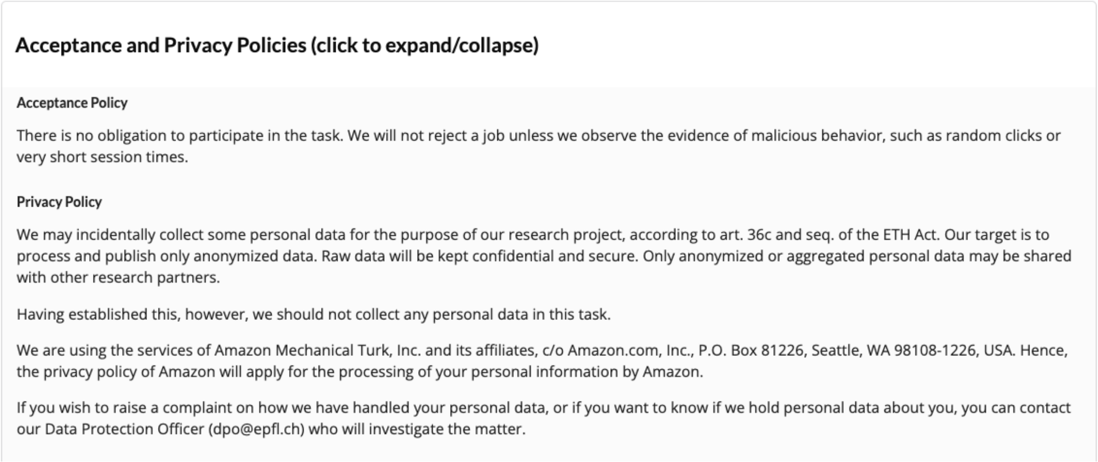
Type	Relation	Relation Description
Physical-Entity	ObjectUse	used for
	AtLocation	located or found at/in/on
	MadeUpOf	made (up) of
	HasProperty	can be characterized by being/having
	CapableOf	is/are capable of
	Desires	desires
	NotDesires	do(es) not desire
Event	IsAfter	happens after
	IsBefore	happens before
	HasSubEvent	includes the event/action
	HinderedBy	can be hindered by
	Causes	causes
	xReason	because
Social-Interaction	xNeed	but before, person X needs
	xAttr	person X is seen as
	xEffect	as a result, person X will
	xReact	as a result, person X feels
	xWant	as a result, person X wants
	xIntent	because person X wants
	oEffect	as a result, others will
	oReact	as a result, others feel
	oWant	as a result, others want

Table 7: Commonsense relations in ATOMIC$^{20}_{20}$ knowledge base that are considered in our experiments on $\mathcal{C}om\mathcal{F}act$ benchmark.

For preprocessing WebNLG+ 2020 (Ferreira et al., 2020) dataset, we follow Grapher (Melnyk et al., 2022) to remove underscores and surrounding quotes appeared in the dataset, and convert non-English characters into their closest available English characters, *e.g.*, "õ" and "å" are mapped to "o" and "a". After preprocessing, We develop our models based on the $\sim 35K$ WebNLG training texts and their linked RDF facts.

E Human Evaluation Details

Our annotator pool for human evaluation contains 58 Amazon Mechanical Turk workers who are located in the USA and have been previously qualified by us for other similar tasks. To prepare the



Acceptance and Privacy Policies (click to expand/collapse)

Acceptance Policy

There is no obligation to participate in the task. We will not reject a job unless we observe the evidence of malicious behavior, such as random clicks or very short session times.

Privacy Policy

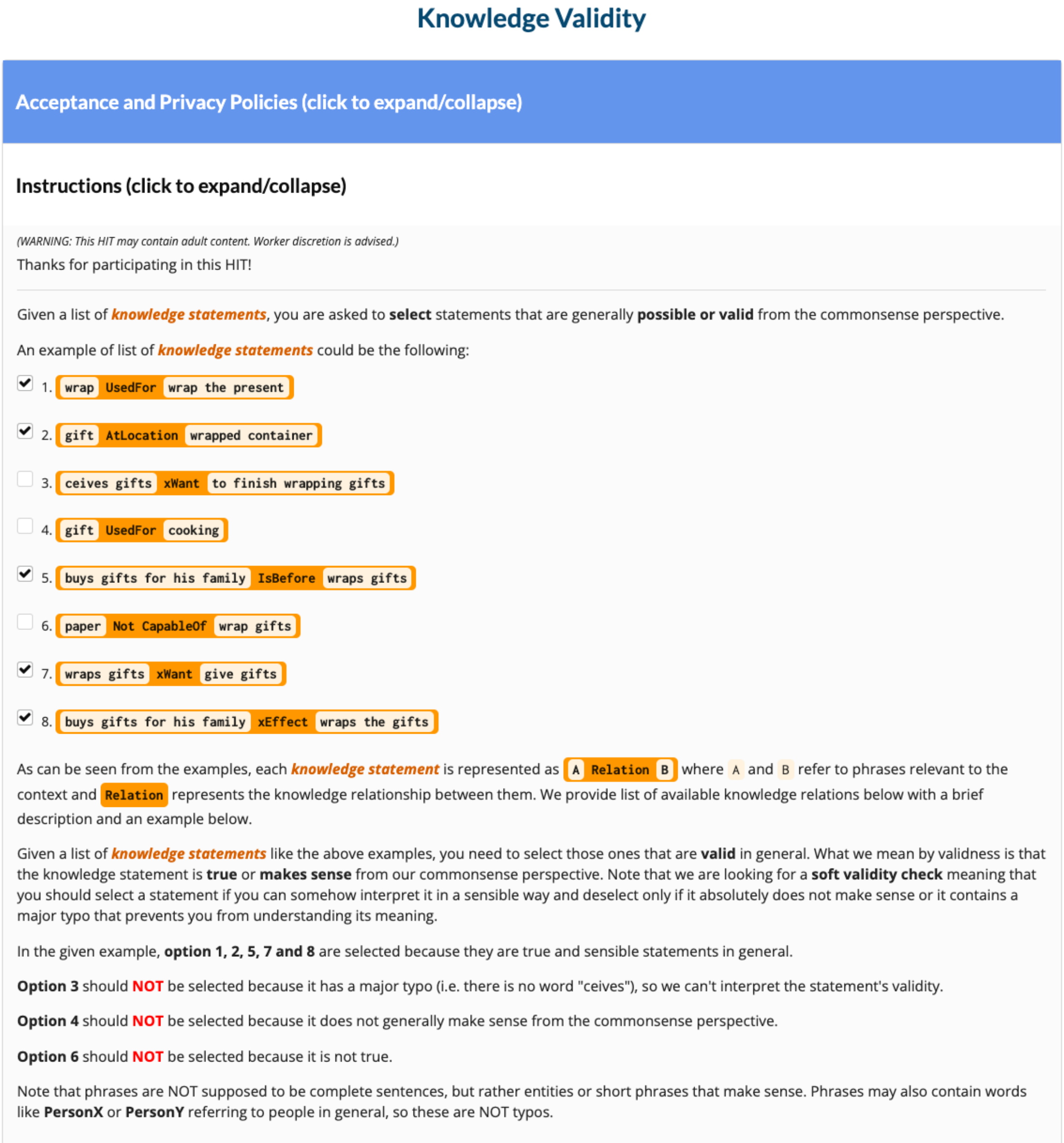
We may incidentally collect some personal data for the purpose of our research project, according to art. 36c and seq. of the ETH Act. Our target is to process and publish only anonymized data. Raw data will be kept confidential and secure. Only anonymized or aggregated personal data may be shared with other research partners.

Having established this, however, we should not collect any personal data in this task.

We are using the services of Amazon Mechanical Turk, Inc. and its affiliates, c/o Amazon.com, Inc., P.O. Box 81226, Seattle, WA 98108-1226, USA. Hence, the privacy policy of Amazon will apply for the processing of your personal information by Amazon.

If you wish to raise a complaint on how we have handled your personal data, or if you want to know if we hold personal data about you, you can contact our Data Protection Officer (dpo@epfl.ch) who will investigate the matter.

Figure 6: Screenshot of Amazon MTurk Acceptance and Privacy Policy



Knowledge Validity

Acceptance and Privacy Policies (click to expand/collapse)

Instructions (click to expand/collapse)

(WARNING: This HIT may contain adult content. Worker discretion is advised.)

Thanks for participating in this HIT!

Given a list of ***knowledge statements***, you are asked to **select** statements that are generally **possible or valid** from the commonsense perspective.

An example of list of ***knowledge statements*** could be the following:

☑ 1. `wrap` `UsedFor` `wrap the present`

☑ 2. `gift` `AtLocation` `wrapped container`

☐ 3. `ceives gifts` `xWant` `to finish wrapping gifts`

☐ 4. `gift` `UsedFor` `cooking`

☑ 5. `buys gifts for his family` `IsBefore` `wraps gifts`

☐ 6. `paper` `Not CapableOf` `wrap gifts`

☑ 7. `wraps gifts` `xWant` `give gifts`

☑ 8. `buys gifts for his family` `xEffect` `wraps the gifts`

As can be seen from the examples, each ***knowledge statement*** is represented as `A` `Relation` `B` where `A` and `B` refer to phrases relevant to the context and `Relation` represents the knowledge relationship between them. We provide list of available knowledge relations below with a brief description and an example below.

Given a list of ***knowledge statements*** like the above examples, you need to select those ones that are **valid** in general. What we mean by validness is that the knowledge statement is **true** or **makes sense** from our commonsense perspective. Note that we are looking for a **soft validity check** meaning that you should select a statement if you can somehow interpret it in a sensible way and deselect only if it absolutely does not make sense or it contains a major typo that prevents you from understanding its meaning.

In the given example, **option 1, 2, 5, 7 and 8** are selected because they are true and sensible statements in general.

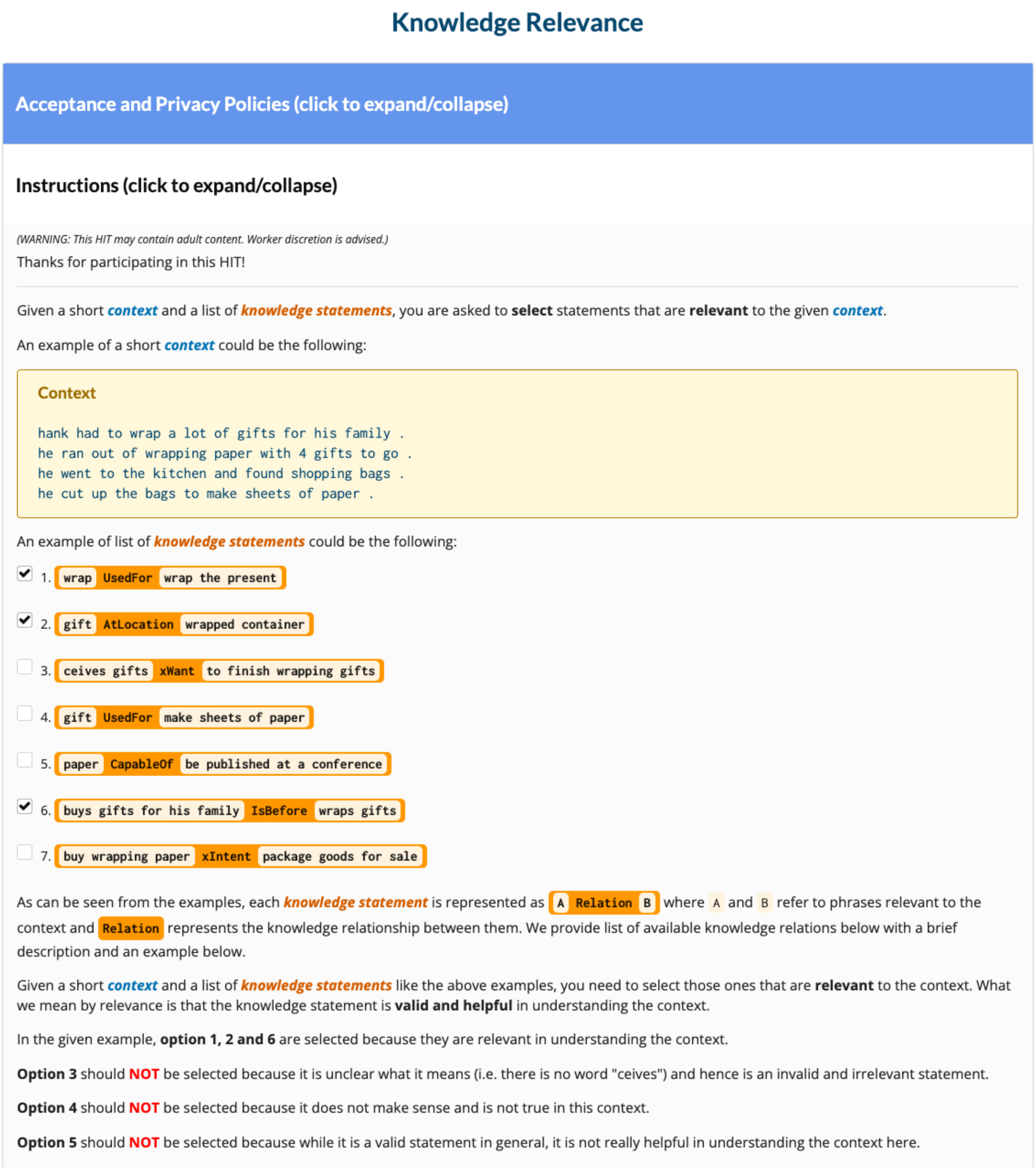
Option 3 should **NOT** be selected because it has a major typo (i.e. there is no word "ceives"), so we can't interpret the statement's validity.

Option 4 should **NOT** be selected because it does not generally make sense from the commonsense perspective.

Option 6 should **NOT** be selected because it is not true.

Note that phrases are NOT supposed to be complete sentences, but rather entities or short phrases that make sense. Phrases may also contain words like **PersonX** or **PersonY** referring to people in general, so these are NOT typos.

Figure 7: Screenshot of Amazon MTurk instructions for knowledge validation task.



Knowledge Relevance

Acceptance and Privacy Policies (click to expand/collapse)

Instructions (click to expand/collapse)

(WARNING: This HIT may contain adult content. Worker discretion is advised.)

Thanks for participating in this HIT!

Given a short ***context*** and a list of ***knowledge statements***, you are asked to **select** statements that are **relevant** to the given ***context***.

An example of a short ***context*** could be the following:

Context

```
hank had to wrap a lot of gifts for his family .
he ran out of wrapping paper with 4 gifts to go .
he went to the kitchen and found shopping bags .
he cut up the bags to make sheets of paper .
```

An example of list of ***knowledge statements*** could be the following:

☑ 1. `wrap` `UsedFor` `wrap the present`

☑ 2. `gift` `AtLocation` `wrapped container`

□ 3. `ceives gifts` `xWant` `to finish wrapping gifts`

□ 4. `gift` `UsedFor` `make sheets of paper`

□ 5. `paper` `CapableOf` `be published at a conference`

☑ 6. `buys gifts for his family` `IsBefore` `wraps gifts`

□ 7. `buy wrapping paper` `xIntent` `package goods for sale`

As can be seen from the examples, each ***knowledge statement*** is represented as `A` `Relation` `B` where `A` and `B` refer to phrases relevant to the context and `Relation` represents the knowledge relationship between them. We provide list of available knowledge relations below with a brief description and an example below.

Given a short ***context*** and a list of ***knowledge statements*** like the above examples, you need to select those ones that are **relevant** to the context. What we mean by relevance is that the knowledge statement is **valid and helpful** in understanding the context.

In the given example, **option 1, 2 and 6** are selected because they are relevant in understanding the context.

Option 3 should **NOT** be selected because it is unclear what it means (i.e. there is no word "ceives") and hence is an invalid and irrelevant statement.

Option 4 should **NOT** be selected because it does not make sense and is not true in this context.

Option 5 should **NOT** be selected because while it is a valid statement in general, it is not really helpful in understanding the context here.

Figure 8: Screenshot of Amazon MTurk instructions for knowledge relevance task.

Backbone	Model	# Facts	Clustering *w.r.t.* Word-Level Edit Distance				Clustering *w.r.t.* Embedding Euclidean Distance			
			# Clusters	Relevance	Alignment	RA-F1	# Clusters	Relevance	Alignment	RA-F1
BART (base)	Greedy	2.48	1.08	32.04	31.98	32.01	1.09	32.11	48.59	38.67
	Sampling-10	10.00	5.59	39.20	46.03	42.34	5.64	38.93	64.51	48.56
	Sampling-15	15.00	7.64	37.18	49.78	42.57	7.82	36.86	68.00	47.81
	Beam-10	10.00	2.63	38.30	44.96	41.36	2.83	38.87	59.58	47.05
	Beam-15	15.00	3.48	41.46	48.04	44.51	3.97	42.88	63.14	51.07
	DIFFUCOMET-Fact	13.40	4.74	59.75	54.07	56.77	5.85	60.32	73.38	66.21
	DIFFUCOMET-Entity	10.08	4.51	62.27	54.61	58.19	5.24	61.77	71.54	66.30
BART (large)	Greedy	2.20	1.38	60.45	36.11	45.21	1.37	60.22	52.31	55.99
	Sampling-10	10.00	6.68	56.09	52.10	54.02	6.40	56.68	73.86	64.14
	Sampling-15	15.00	**8.89**	56.24	55.18	55.70	**8.56**	56.57	76.30	64.97
	Beam-10	10.00	3.32	64.94	50.72	56.96	3.51	64.37	69.14	66.67
	Beam-15	15.00	4.17	64.18	53.66	58.45	4.60	64.35	71.35	67.67
	DIFFUCOMET-Fact	12.88	4.47	65.82	54.18	59.44	5.24	65.64	71.65	68.51
	DIFFUCOMET-Entity	12.89	5.09	67.00	58.22	**62.30**	5.67	66.39	74.38	**70.16**
COMET-BART	Greedy	1.96	1.14	61.27	34.76	44.36	1.19	61.42	50.64	55.51
	Sampling-10	10.00	6.45	56.79	53.36	55.02	6.30	56.60	73.64	64.01
	Sampling-15	15.00	8.52	55.78	**58.99**	57.34	8.39	56.19	**77.97**	65.31
	Beam-10	10.00	3.78	65.62	53.45	58.91	3.89	65.73	70.65	68.10
	Beam-15	15.00	4.78	64.91	54.77	59.41	5.09	65.03	71.64	68.18
T5 (large)	Grapher	5.08	1.75	67.82	33.07	44.46	2.60	68.29	40.58	50.91
-	Gold	10.55	5.64	81.06	-	-	5.64	80.90	-	-

Table 8: Clustering-based evaluation results on the **ROCStories** portion of $\mathcal{C}om\mathcal{F}act$. Best results (excluding Gold references) are in bold. Different numbers after Sampling and Beam denote various sampling numbers or beam search sizes being tested.

Backbone	Model	Distinct-4	BLEU	METEOR	ROUGE-L
BART (base)	Greedy	**99.90**	8.70	40.49	44.43
	Sampling-10	85.29	7.16	37.78	39.20
	Sampling-15	81.57	8.24	38.35	40.13
	Beam-10	50.32	12.25	42.23	43.53
	Beam-15	45.21	11.51	42.04	42.91
	DIFFUCOMET-Fact	57.87	12.09	46.43	47.13
	DIFFUCOMET-Entity	70.02	14.25	43.34	45.08
BART (large)	Greedy	93.01	9.12	43.98	46.26
	Sampling-10	86.33	9.89	43.85	43.69
	Sampling-15	81.56	9.47	43.28	43.15
	Beam-10	47.03	15.02	48.56	48.15
	Beam-15	43.73	13.11	47.70	46.35
	DIFFUCOMET-Fact	52.46	15.98	50.06	51.44
	DIFFUCOMET-Entity	63.49	17.01	47.61	48.40
COMET-BART	Greedy	65.95	18.01	**52.32**	**54.96**
	Sampling-10	83.29	13.35	44.77	45.80
	Sampling-15	79.01	12.69	44.43	45.58
	Beam-10	51.13	**19.89**	50.14	50.48
	Beam-15	47.27	16.97	47.39	47.19
T5 (large)	Grapher	67.83	1.40	23.96	27.21
-	Gold	80.45	-	-	-

Table 9: Evaluation results of natural language generation metrics on the **ROCStories** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

workers for the new tasks of assessing the validity and relevance of knowledge in a given context, we share the instructions with them beforehand and do a small pilot run where we evaluate the quality of the worker annotations and give feedback if needed. We pay each worker $0.10 for each task. Figure 6, 7 and 8 show screenshots of our acceptance/privacy policy and instructions for knowledge validation and relevance tasks. Our data collection protocol follows Amazon Mechanical Turk regulations, and is approved by our organization's human research ethics committee.

F Full Results of Knowledge Generation

F.1 ROCStories

In Table 8 and 9, we present our full evaluation results of contextual commonsense knowledge generation on the ROCStories portion of $\mathcal{C}om\mathcal{F}act$ benchmark. For evaluating Sampling and Beam baseline models, we test two sampling or beam search sizes around the average number of gold facts per context, *i.e.*, 10 and 15 as indicated by the suffix numbers, and adopt the size which achieves better F1 results. On both base and large model scales, DIFFUCOMET models achieve consistently better balance between the diversity (*i.e.*, # Clusters) and accuracy (*i.e.*, RA-F1) of knowledge generation, compared to baseline models that typically perform generation in the autoregressive manner.

F.2 Context Generalization

In this section, we present zero-shot evaluation results of models (trained on the contexts of ROCStories) generalizing to the contexts of other three $\mathcal{C}om\mathcal{F}act$ portions, including PersonaChat (Table 10 and 11), MuTual (Table 12 and 13) and MovieSummaries (Table 14 and 15).

We observe that both DIFFUCOMET models

Backbone	Model	# Facts	Clustering *w.r.t.* Word-Level Edit Distance				Clustering *w.r.t.* Embedding Euclidean Distance			
			# Clusters	Relevance	Alignment	RA-F1	# Clusters	Relevance	Alignment	RA-F1
BART (base)	Greedy	2.62	1.24	33.52	35.64	34.55	1.24	33.57	48.61	39.71
	Sampling-10	10.00	**5.23**	29.35	44.49	35.37	4.63	28.26	58.85	38.18
	Beam-15	15.00	3.65	31.11	47.01	37.44	3.49	30.41	58.90	40.11
	DIFFUCOMET-Fact	13.73	4.97	44.25	52.81	48.15	5.24	44.76	**69.24**	54.37
	DIFFUCOMET-Entity	11.40	4.99	50.39	54.84	52.52	**4.94**	49.36	68.55	57.39
BART (large)	Beam-15	15.00	4.44	53.98	54.13	54.05	4.04	54.07	63.17	58.27
	DIFFUCOMET-Fact	10.82	4.48	**55.44**	55.20	55.32	3.89	**55.02**	65.20	59.68
	DIFFUCOMET-Entity	12.06	4.72	55.08	**57.12**	**56.08**	4.48	54.42	68.11	**60.50**
COMET-BART	Beam-15	15.00	4.86	54.02	54.78	54.40	4.27	53.97	65.15	59.04
T5 (large)	Grapher	4.53	1.68	47.74	30.51	37.23	1.57	47.94	36.18	41.24
-	Gold	8.60	4.76	70.42	-	-	4.28	70.42	-	-

Table 10: Zero-shot clustering-based evaluation results on the **PersonaChat** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

Backbone	Model	Distinct-4	BLEU	METEOR	ROUGE-L
BART (base)	Greedy	**97.83**	8.72	44.44	46.52
	Sampling-10	86.81	4.09	32.95	33.80
	Beam-15	53.05	8.06	37.43	38.62
	DIFFUCOMET-Fact	63.40	5.84	37.53	39.33
	DIFFUCOMET-Entity	73.38	9.04	34.35	36.46
BART (large)	Beam-15	47.64	8.71	41.40	40.44
	DIFFUCOMET-Fact	57.23	8.05	**45.83**	**47.11**
	DIFFUCOMET-Entity	68.54	**11.11**	38.88	40.04
COMET-BART	Beam-15	50.13	10.25	43.47	42.38
T5 (large)	Grapher	52.99	0.68	19.91	22.41
-	Gold	84.96	-	-	-

Table 11: Zero-shot evaluation results of natural language generation metrics on the **PersonaChat** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

generalize well to the contexts of PersonaChat and MuTual, whose generated knowledge possesses comparable diversity (*i.e.*, # Clusters) and better accuracy (*i.e.*, RA-F1) than the strongest baseline model Beam-COMET. More interestingly, we find that DIFFUCOMET-Entity achieves larger points of improvements over baselines on the more challenging MovieSummaries-style contexts, while DIFFUCOMET-Fact struggles to outperform the strongest baseline Beam-COMET, showing that entity-level diffusion is more robust to the shift of narrative contexts, likely due to the more fine-grained multi-step learning of context-to-knowledge mapping.

F.3 Case Study and Knowledge Types

Table 16 showcases the knowledge generation results of DIFFUCOMET models in a narrative context sampled from $\mathcal{C}om\mathcal{F}act$ ROCStories, compared to the sampling and beam search baselines Sample-COMET and Beam-COMET. Facts that are novel (*i.e.*, beyond the coverage of gold references) and relevant to the context are labeled in bold. We find that both DIFFUCOMET-Fact and DIFFUCOMET-Entity can generate facts that are rich in diversity, covering both physical entities (*e.g.*, baseball cap) and social events (*e.g.*, go on vacation). Novel facts generated by DIFFUCOMET models also uncover implicit inter-connections between entities or events in the narrative context, *e.g.*, "vacation" and "family" are associated because "X goes on vacation" to "spend time with family". By contrast, Beam-COMET model mainly generates simple facts about physical entities, and Sample-COMET model generates many facts that are irrelevant to the context, *e.g.*, "field is used for playing baseball".

We also conduct a study on the proportion of different knowledge types that each model generates per context, based on the ROCStories portion of $\mathcal{C}om\mathcal{F}act$ benchmark. In particular, we divide commonsense facts into three types according to their relation groups under ATOMIC$^{20}_{20}$ knowledge scheme, as shown in Table 7, including facts that are centered on physical entities, events and social interactions. Table 17 shows the results of knowledge proportion generated by DIFFUCOMET and baseline models, with gold references. Compared to Sampling-COMET and Beam-COMET baselines, DIFFUCOMET models generate a larger proportion of facts that reveal complex event or social inter-connections. The proportion of social-based facts generated by DIFFUCOMET even significantly surpasses the gold references. All above results imply that diffusion models have the potential to uncover more in-depth and implicit commonsense inferences from narrative contexts, which may not be easily extracted from existing knowledge bases.

F.4 WebNLG+ 2020

We present our full evaluation results on the WebNLG+ 2020 benchmark in Table 18, 19 and 20. For evaluating Sampling and Beam baselines,

Backbone	Model	# Facts	Clustering *w.r.t.* Word-Level Edit Distance				Clustering *w.r.t.* Embedding Euclidean Distance			
			# Clusters	Relevance	Alignment	RA-F1	# Clusters	Relevance	Alignment	RA-F1
BART (base)	Greedy	2.50	1.16	35.85	33.63	34.70	1.19	36.10	48.53	41.40
	Sampling-10	10.00	**5.68**	43.77	45.76	44.74	**5.88**	43.98	63.75	52.05
	Beam-15	15.00	3.55	41.27	49.72	45.10	4.08	42.27	63.17	50.65
	DIFFUCOMET-Fact	13.11	4.54	57.64	52.25	54.81	5.57	57.51	70.10	63.18
	DIFFUCOMET-Entity	10.63	4.60	60.08	54.65	57.24	5.27	59.08	68.88	63.60
BART (large)	Beam-15	15.00	3.92	64.19	51.75	57.30	4.45	62.41	67.31	64.77
	DIFFUCOMET-Fact	10.46	4.33	64.74	54.51	59.19	4.80	64.13	68.07	66.04
	DIFFUCOMET-Entity	11.85	4.70	64.39	**55.91**	**59.85**	5.39	63.82	**71.22**	**67.32**
COMET-BART	Beam-15	15.00	4.52	61.88	54.04	57.69	4.75	60.56	69.72	64.82
T5 (large)	Grapher	4.50	1.70	**73.30**	32.74	45.26	1.78	**73.33**	43.13	54.31
-	Gold	10.80	5.58	74.63	-	-	5.79	74.77	-	-

Table 12: Zero-shot clustering-based evaluation results on the **MuTual** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

Backbone	Model	Distinct-4	BLEU	METEOR	ROUGE-L
BART (base)	Greedy	**97.47**	**14.05**	**49.89**	50.78
	Sampling-10	86.42	5.61	37.11	38.60
	Beam-15	49.31	11.57	45.47	45.51
	DIFFUCOMET-Fact	60.66	8.71	44.23	46.11
	DIFFUCOMET-Entity	70.94	11.08	40.28	42.15
BART (large)	Beam-15	43.91	11.37	46.86	46.75
	DIFFUCOMET-Fact	52.00	12.33	49.50	**50.97**
	DIFFUCOMET-Entity	66.12	12.68	45.11	45.73
COMET-BART	Beam-15	47.40	12.40	49.12	48.57
T5 (large)	Grapher	51.30	1.96	24.70	29.36
-	Gold	80.99	-	-	-

Table 13: Zero-shot evaluation results of natural language generation metrics on the **MuTual** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

we set both sampling and beam search sizes as 5, which is around the average number of gold facts per context. Consistent with the evaluation results on $\mathcal{C}om\mathcal{F}act$, DIFFUCOMET models keep achieving better performances on the WebNLG task of factual knowledge generation, implying that our method of diffusion-based contextual knowledge generation can generalize well to knowledge beyond commonsense.

F.5 Comparison of Fact and Entity Diffusion

For the comparison in between our two diffusion models, DIFFUCOMET-Entity in general outperforms DIFFUCOMET-Fact on our proposed metrics, which may benefit from more fine-grained multi-step learning of knowledge construction in pipeline. However, DIFFUCOMET-Fact is computational cheaper, *i.e.*, only requires a single step of fact diffusion instead of two steps of (head and tail) entity diffusion and a relation prediction.

Backbone	Model	# Facts	Clustering *w.r.t.* Word-Level Edit Distance				Clustering *w.r.t.* Embedding Euclidean Distance			
			# Clusters	Relevance	Alignment	RA-F1	# Clusters	Relevance	Alignment	RA-F1
BART (base)	Greedy	2.59	1.12	33.50	25.28	28.82	1.11	33.38	37.95	35.52
	Sampling-10	10.00	4.90	26.61	33.31	29.59	4.24	24.96	50.26	33.36
	Beam-15	15.00	3.54	30.45	36.07	33.02	3.17	29.13	49.63	36.71
	DiffuCOMET-Fact	14.61	6.02	35.97	38.76	37.31	5.49	36.29	59.83	45.18
	DiffuCOMET-Entity	15.82	**6.31**	39.86	39.93	39.89	5.76	39.57	57.55	46.90
BART (large)	Beam-15	15.00	4.46	42.92	32.79	37.18	3.70	42.52	50.46	46.15
	DiffuCOMET-Fact	8.29	3.01	41.50	30.47	35.14	2.89	40.82	46.59	43.51
	DiffuCOMET-Entity	13.50	6.28	44.08	**40.46**	**42.19**	**5.84**	42.70	**61.56**	**50.42**
Comet-BART	Beam-15	15.00	5.06	41.97	34.63	37.95	4.04	41.54	51.24	45.88
T5 (large)	Grapher	5.34	1.83	**54.09**	23.74	33.00	1.50	**54.12**	34.27	41.97
-	Gold	9.00	5.64	58.55	-	-	4.81	58.37	-	-

Table 14: Zero-shot clustering-based evaluation results on the **MovieSummaries** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

Backbone	Model	Distinct-4	BLEU	METEOR	ROUGE-L
BART (base)	Greedy	**95.18**	5.14	34.24	36.53
	Sampling-10	90.99	2.52	24.56	28.08
	Beam-15	53.65	4.82	28.33	31.56
	DiffuCOMET-Fact	63.93	2.27	27.09	30.08
	DiffuCOMET-Entity	67.24	2.68	24.14	26.95
BART (large)	Beam-15	47.57	4.89	31.41	33.19
	DiffuCOMET-Fact	43.68	**5.26**	**34.55**	**38.80**
	DiffuCOMET-Entity	67.13	3.83	26.63	29.32
Comet-BART	Beam-15	50.29	5.18	31.36	33.39
T5 (large)	Grapher	42.76	0.46	18.24	21.28
-	Gold	87.39	-	-	-

Table 15: Zero-shot evaluation results of natural language generation metrics on the **MovieSummaries** portion of $\mathcal{C}om\mathcal{F}act$. Notations are same as Table 8.

Narrative Context	Dustin loved to wear his baseball cap everywhere he went.
	On vacation his family visited the windy city of Chicago.
	Dustin's baseball cap blew off his head and into the street.
	His dad waited until it was safe before getting Dustin's cap.
	He loved his baseball cap even though it was a little dirty.
Gold	cap, used for, to wear on head
	cap, used for, wear on their heads
	head cap, used for, put on head
	vacation, used for, have fun on
	vacation, used for, fun
	vacation, used for, relax out of work and school
	family, is capable of, plan to go on vacation
	X takes a family trip, because X wants, to go on vacation
	X visits the city, X is seen as, traveling
	dad, can be characterized by being, one of human's parents
Sample -COMET	baseball cap, used for, protect the head
	baseball cap, used for, protect your head while playing baseball
	baseball cap, used for, wearing over head
	X ops for baseball, but before X needs, to find a baseball
	X's favorite baseball, because X wants to, enjoy the sport
	baseball, used for, sport as a mascot
	vacation, used for, have fun on
	X chases the wind, because X wants, to walk around
	port, used for, get vacation
	field, used for, playing baseball
	cap, used for, keep their head up
	cap, used for, protection from wind
	cap, used for, protect head while traveling
	cap, used for, wear around head
	jersey, used for, wear while playing
Beam -COMET	baseball cap, used for, wear while playing baseball
	baseball cap, used for, wear on their head
	baseball cap, used for, wear on the head
	baseball, used for, playing baseball with friends
	baseball, used for, playing baseball with family
	sport cap, used for, wear while playing
	Chicago, can be characterized by having, many streets
	Chicago, can be characterized by having, many cities
	Chicago, can be characterized by having, many neighborhoods
	cap, used for, wear on head while playing baseball
	cap, used for, wear to the game with
	cap, used for, protect head from wind
	cap, used for, protect head from wind blows
	cap, used for, keep the cap on
	cap, used for, keep the cap clean
DIFFUCOMET -Fact	baseball cap, used for, to put on
	baseball cap, used for, to keep baseball cap on head
	baseball cap, used for, wear
	baseball cap, used for, to play baseball with
	city, used for, live in
	vacation, used for, relax
	X takes a family trip, but before X needs, to spend time with family
	X takes a family trip, because X wants, to enjoy family time
	X goes on vacation, because X wants, to spend time with family
	X is on vacation, because X wants, to spend time with family
	dad, can be characterized by being, one of human's parents
	dad's car, used for, to be safe
	safe, used for, safe to wear
DIFFUCOMET -Entity	cap, used for, wear on head
	cap, used for, wear on the head
	baseball cap, used for, look professional
	baseball cap, used for, to play baseball with
	X is wearing cap, but before X needs, have a cap
	X is wearing cap, but before X needs, put on a cap
	go on vacation, includes the action, take family to beach
	go on vacation, includes the action, go somewhere nice
	vacation, used for, enjoy your time off
	X goes on vacation, because X wants, to spend time with family
	dad, can be characterized by being, one of human's parents
	safe, used for, keeping things safe

Table 16: Examples of contextual knowledge generation. Novel and contextually relevant facts are in bold. Model notations are same as Table 1.

Model	**Physical**	**Event**	**Social**
Sampling-COMET	46.17	4.01	49.82
Beam-COMET	60.72	2.36	36.92
DIFFUCOMET-Fact	41.00	4.66	54.34
DIFFUCOMET-Entity	35.75	4.53	59.72
Gold	43.54	7.32	49.14

Table 17: Proportion (%) of different types of knowledge generation on the ROCStories portion of $\mathcal{C}om\mathcal{F}act$. "Physical", "Event" and "Social" denote facts with relation types belonging to physical-entity, event and social-interaction, respectively, as shown in Table 7. Model notations are same as Table 1.

Backbone	Model	# Facts	Clustering *w.r.t.* Word-Level Edit Distance				Clustering *w.r.t.* Embedding Euclidean Distance			
			# Clusters	Relevance	Alignment	RA-F1	# Clusters	Relevance	Alignment	RA-F1
BART (large)	Greedy	1.69	0.88	83.16	50.26	62.65	0.88	83.16	71.71	77.01
	Sampling-5	5.00	2.09	81.10	71.25	75.86	1.73	80.89	83.93	82.38
	Beam-5	5.00	2.12	82.70	72.69	77.37	1.64	82.50	85.78	84.11
	DiffuCOMET-Fact	2.56	1.69	84.39	74.12	78.92	1.51	84.38	86.18	85.27
	DiffuCOMET-Entity	2.71	1.82	**87.86**	**78.46**	**82.89**	1.57	**87.76**	**88.59**	**88.17**
Comet-BART	Greedy	1.61	0.96	83.33	54.34	65.78	0.95	83.33	77.23	80.16
	Sampling-5	5.00	2.09	80.89	72.77	76.62	**1.76**	80.72	84.21	82.43
	Beam-5	5.00	**2.15**	82.11	72.94	77.25	1.70	81.94	85.94	83.89
T5 (large)	Grapher	2.10	1.39	83.48	70.66	76.54	1.29	83.46	82.21	82.83
-	Gold	3.22	2.27	96.43	-	-	1.91	96.43	-	-

Table 18: Clustering-based evaluation results on the **WebNLG+ 2020** benchmark. Notations are same as Table 8.

Backbone	Model	Distinct-4	BLEU	METEOR	ROUGE-L
BART (large)	Greedy	87.29	81.12	84.57	84.92
	Sampling-5	48.24	74.22	81.71	81.19
	Beam-5	45.58	75.01	81.78	80.51
	DiffuCOMET-Fact	81.02	80.43	83.23	84.30
	DiffuCOMET-Entity	82.20	**83.04**	**89.88**	**89.72**
Comet-BART	Greedy	**93.17**	81.43	84.95	85.34
	Sampling-5	47.36	75.44	81.84	81.85
	Beam-5	46.17	73.56	80.93	79.46
T5 (large)	Grapher	89.95	76.17	79.61	80.89
-	Gold	82.05	-	-	-

Table 19: Evaluation results of natural language generation metrics on the **WebNLG+ 2020** benchmark. Notations are same as Table 8.

Backbone	Model	Exact Match			Partial Match			Strict Match		
		Web-Prec.	**Web-Rec.**	**Web-F1**	**Web-Prec.**	**Web-Rec.**	**Web-F1**	**Web-Prec.**	**Web-Rec.**	**Web-F1**
BART (large)	Greedy	50.42	52.79	51.51	53.76	56.84	55.20	50.14	52.53	51.25
	Sampling-5	73.65	76.73	75.11	79.57	83.89	81.66	72.37	75.45	73.83
	Beam-5	75.32	78.39	76.76	81.32	85.72	83.38	73.36	76.27	74.75
	DIFFUCOMET-Fact	76.59	78.35	77.47	79.17	81.52	80.35	76.30	78.07	77.19
	DIFFUCOMET-Entity	**80.80**	**82.97**	**81.84**	**83.72**	**86.48**	**85.07**	**80.68**	**82.89**	**81.74**
COMET-BART	Greedy	52.55	54.82	53.62	55.99	58.95	57.39	52.30	54.59	53.37
	Sampling-5	74.96	77.87	76.33	80.31	84.41	82.18	73.77	76.67	75.15
	Beam-5	75.95	78.88	77.03	81.66	85.84	83.15	73.80	76.61	74.85
T5 (large)	Grapher	71.50	73.30	72.20	74.10	76.50	75.00	71.20	73.00	71.90

Table 20: Evaluation results on official metrics provided by the **WebNLG+ 2020** benchmark challenge. We present the results of Grapher as reported in its paper. Notations are same as Table 8.