Grounded Keys-to-Text Generation: Towards Factual Open-Ended Generation

Faeze Brahman[♠♡*] **Baolin Peng**[◇] **Michel Galley**[◇]
Sudha Rao[◇] **Bill Dolan**[◇] **Snigdha Chaturvedi**[♣] **Jianfeng Gao**[◇]
[♠]Allen Institute for Artificial Intelligence
[♡]Paul G. Allen School of Computer Science & Engineering, University of Washington
[◇]Microsoft Research, [♣]UNC Chapel Hill
faezeb@allenai.org

Abstract

Large pre-trained language models have recently enabled open-ended generation frameworks (e.g., prompt-to-text NLG) to tackle a variety of tasks going beyond the traditional data-to-text generation. While this framework is more general, it is under-specified and often leads to a lack of controllability restricting their real-world usage. We propose a new *grounded keys-to-text* generation task: the task is to generate a factual description about an entity given a set of guiding keys, and grounding passages. To address this task, we introduce a new dataset, called ENTDEGEN. Inspired by recent QA-based evaluation measures, we propose an automatic metric, MAFE, for factual correctness of generated descriptions. Our ENTDESCRIPTOR model is equipped with strong rankers to fetch helpful passages and generate entity descriptions. Experimental result shows a good correlation (60.14) between our proposed metric and human judgments of factuality. Our rankers significantly improved the factual correctness of generated descriptions (15.95% and 34.51% relative gains in recall and precision). Finally, our ablation study highlights the benefit of combining *keys* and *groundings*.

1 Introduction

Converting information to text (McKeown, 1985) has been a cornerstone of NLG research with the goal of improving the accessibility of knowledge to general users. It has found many applications such as generating sport commentaries (Wiseman et al., 2017), weather forecast (Konstas and Lapata, 2012), biographical text (Lebret et al., 2016), and dialogue response generation (Wen et al., 2015, 2016). The problem has traditionally been formulated as data-to-text generation, to generate an output given structured input such as graph, tables or key-value pairs. However, this formulation is overspecified and does not cover other *open-ended* scenarios in real-world. Recent advances in large pre-trained language models (PLMs), as well as the general knowledge represented in them, have made it possible to formulate the problem as *prompt-to-text* or *outline-to-text* (Rashkin et al., 2020) generation. This offers the prospect of making NLG more broadly applicable, as such models allow input to be more parsimonious or ill-defined. However, issues such as lack of controllability and hallucination have lessened the practical applicability of this setting in real-world scenarios.

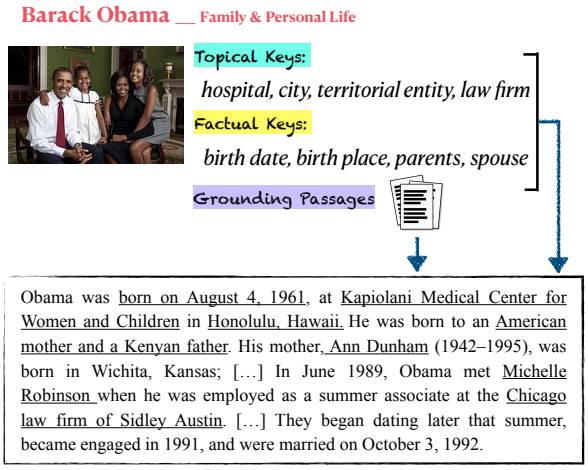

Figure 1: An example from ENTDEGEN dataset. Given a set of topical and factual keys, along with multiple grounding passages, the task is to generate an entity description. Corresponding knowledge are underlined.

To overcome these issues, we propose a new task, *grounded keys-to-text* generation, where given a wishlist of *keys* (without the values) about an entity[1] and a set of short *grounding passages* as a source knowledge, the goal is to generate a factual description. An example is shown in Fig. 1, where the task is to generate a paragraph about "Barack Obama", in particular about his family and personal life. Potential *factual keys* in this example are "birth

[*]Work done while first author was interning at MSR.

[1]We consider a broad definition of "entity" which includes person, place, event, species, etc.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 7426 - 7442

date, birthplace, parents, spouse, children", etc. The task also enables a finer-grained control over the types of entities to be included in the output via *topical keys* such as "hospital, city, law firm" for the example in Fig. 1. Finally, pertinent information about the entities needs to be fetched from a set of candidate *grounding passages*. These passages can be obtained via internet search. Our task differs from similar existing tasks, such as data-to-text generation (Koncel-Kedziorski et al., 2019; Chen et al., 2021) in that, we presume keys but not values are given. This covers more open-ended scenarios in the real-world where knowledge about entities are not available in detailed structured format, is constantly changing and so have to be fetched on the fly. Moreover, this formulation offers control to the user over the generated text.

To facilitate research on grounded keys-to-text generation task, we introduce a large-scale and challenging dataset, called ENTDEGEN, with about 375K instances. The *grounded, factual* and *long-form* nature of the task, brings a new challenge, i.e., generating paragraph-level text which is faithful to one or more grounding passages based the provided guiding keys. To address this challenge, we propose ENTDESCRIPTOR equipped with strong ranker to help the model focus on passages that are both relevant to the keys and complementary. We propose two rankers. Our contrastive dense ranker is based on embedding-based retrieval systems trained in a contrastive framework. Our autoregressive ranker generates a sequence of passage indices autoregressivly by modeling the probability of each passage conditioned on previously generated passages. This ranker is shown to achieve the strongest performance by modeling the joint probability of passages.

The factual aspect of generation also calls for a new evaluation metric. Inspired by recent fact-based evaluation for summarization, we propose an automatic metric, called MAFE, to evaluate different aspects of grounded text quality, including relevance and consistency. Our contributions are:[2]

- We introduce a new controllable entity description generation task which requires aggregating knowledge from multiple grounding passages efficiently.
- To address this task, we also present a new dataset, called ENTDEGEN.
- We propose two ranking methods, contrastive dense and autoregressive, to select a sequence of useful passages for the model to ground in.
- We propose an evaluation metric to evaluate factual consistency in our proposed task, which highly correlates with human judgments of factuality.

[2]Data and code available at: https://github.com/fabrahman/Grounded_Keys2Text

2 Task: Grounded keys-to-text Generation

Given an entity e, title t, a set of factual $\mathcal{K}^f = \{k_1^f, k_2^f, ..., k_m^f\}$ and topical $\mathcal{K}^t = \{k_1^t, k_2^t, ..., k_m^t\}$ keys, and grounding passages $\mathcal{P} = \{p_1, p_2, ..., p_N\}$, the goal is to generate a text (description) with respect to the provided keys.

3 Dataset: ENTDEGEN

Our dataset collection strategy is based on Wikipedia and motivated by the WIKITABLET dataset (Chen et al., 2021). Each Wikipedia article $\mathcal{A}_w$ is composed of multiple sections $\mathcal{S} = \{s_1, s_2, ..., s_n\}$. The title of the Wikipedia article is the entity e whose description is to be generated and the text in each section forms a reference (gold) description, r. For example, an article about a football player may contain sections about "Introduction", "Early Life", "Club Career", and "International Career", each forming a separate instance in our dataset. We perform the following steps for each section s_i in an article to obtain: factual keys, topical keys, and grounding passages.

Factual Keys. Factual keys seek specific knowledge about an entity of interest.For obtaining factual keys, we align key-value pairs in infobox and Wikidata with each section s_i. For this, we took a distant-supervision approach to estimate the alignment score of each key-value pair with the section using semantic similarity and lexical precision. For semantic similarity, we compute the precision component of BERT-Score (Zhang et al., 2020b) between the section text and the concatenation of key-value pair (`key + value`). A high value indicates that the key-value pair is semantically relevant to that section. We also measure the ROUGE-L precision score (Lin, 2004) between the section text with respect to the concatenation of key-value pair. For each instance in our dataset, we select keys whose key-value BERT-Score is greater than 0.82,

	Train	Dev	Test
Instances	267,453	2,500	2,497
Avg. Output Len. (token)	116.37	125.56	124.10
Avg. Passage Len. (token)	59.98	60.13	60.25
Avg. Top. Keys	4.14	4.15	4.30
Avg. Fac. Keys	6.04	6.13	6.17

Table 1: ENTDEGEN Dataset Statistics.

and ROUGE-L score is greater than 0.25.[3][4]

Topical Keys. Topical keys are not tied to specific aspects of the entity of interest, but give hints on the type of other entities to be included in the output. For obtaining topical keys, we first find all hyperlinked articles $\mathcal{A}_h$ appearing in the section. We then use the value of the "instance of" or "subclass of" tuple in the Wikidata table of $\mathcal{A}_h$ as the set of topical keys for section s_i. For example, the hyperlinked *Kapiolani Medical Center for Women and Children* in Fig. 1 is an instance of *hospital* according to its Wikidata table. So it will be turned into a topical key *hospital*. Both types of keys help the model to generate an output that satisfies the user's need.

Grounding Passages. For obtaining grounding passages, we use the documents in the WikiSum dataset (Liu et al., 2018). The documents are citations in the Wikipedia article obtained by CommonCrawl or web pages returned by Google Search. Each instance in our data has 40 grounding passages. Note that our dataset is distantly supervised, and these passages may not always contain all the facts regarding the keys. To enhance the quality of our dataset, we filter out entities for which the average Bert-Score recall of key-value pairs against the grounding passages is lower than 0.82.[5]

Basic statistics of ENTDEGEN are provided in Table 1.[6] Fig. 4 in Appendix depicts the diversity of ENTDEGEN entity domains. We associate each entity in our dataset with a domain such as Person, Place, Organization, Event, etc. using the DBPedia knowledge-base (Lehmann et al., 2015). See Appendix B for an assessment of dataset quality.

[3]These threshold values are selected empirically from BERT-Score $\in \{0.80, 0.82, 0.84, 0.86\}$ and ROUGE-L $\in \{0.20, 0.25, 0.30\}$ based on the goodness of the alignment.

[4]These metrics are computed using HuggingFace dataset library: https://github.com/huggingface/datasets.

[5]Similarly, this value is chosen empirically from BERT-Score $\in \{0.80, 0.82, 0.84, 0.86\}$.

[6]Our dataset creation pipeline is generic and can be applied to other encyclopedic knowledge sources.

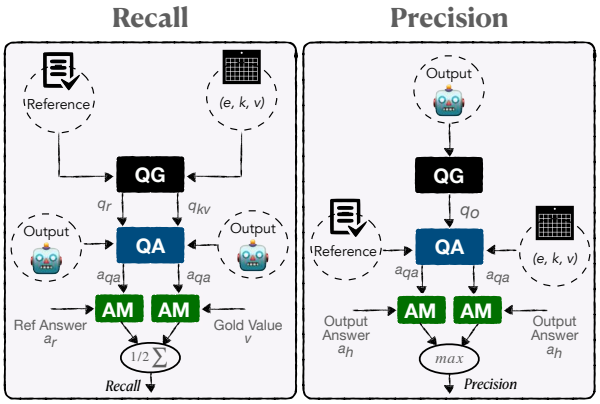

Figure 2: MAFE Evaluation Framework consisting of Question Generation (QG), Question Answering (QA) and Answer Matching (AM) components.

4 MAFE: Multi-Aspect Factuality Evaluation

Our proposed task is to generate a factual description. Hence, it is crucial to evaluate the factuality of the generated texts. Inspired by recent fact-based evaluation in abstractive summarization (Scialom et al., 2019; Durmus et al., 2020; Wang et al., 2020), we propose to assess the factuality of generation through question answering (QA). We evaluate factuality of a generated description h with respect to (i) factual triples *(e, k, v)* which are constructed from the entity e, each factual key k and its value v,[7] and (ii) reference (gold) description r. In our QA based Multi-Aspect Factualy Evaluation (MAFE), questions are generated from spans in the reference and factual triples (recall), or the generated output (precision), and are automatically answered using the output, or reference-factual triples. Then, the similarity between the predicted answer and the gold answer is used to compute recall and precision. Our evaluation framework is illustrated in Fig. 2 which accounts for both relevance (recall) and consistency (precision):

Recall ($h \rightarrow r$) evaluates the generated output h on recalling information from the factual triples *(e, k, v)* AND reference r. For this, we generate questions that have gold answers in factual triples and reference using a Question Generation (QG) module, and obtain answers to these questions from generated output h using a Question Answering (QA) module. We define recall as the average scores of these answers when compared to the gold answers (computed by an Answer Matching (AM) module).

[7]This resembles the *(s, r, o)* in the Knowledge Bases, e.g., `(Barack Obama, place of birth, Hawaii)`.

Precision ($r \rightarrow h$) measures the amount of information contained in the generated output h that is consistent with factual triples *(e, k, v)* OR reference r. For this, we generate questions from output and obtain answers from factual triples and reference. We define precision as the maximum score between answers predicted from factual triples and reference.[8]

Next, we describe the 3 modules of the evaluation framework.

4.1 Question Generation

Given a sentence s containing an answer span a (marked by special tokens), we train a QG model to generate a question q (which is answerable by a), modeling $P_{qg}(q|s, a)$. For evaluating a generated output, we gather a set of answer spans a by extracting all name entities and noun phrases from each sentence s (of reference or output) using spaCy[9] . For generating questions from factual triples, we linearize them by concatenating their constituent elements and consider the value v as the answer a. For example, we form "*Barack Obama place of birth hawaii*" from `(Barack Obama, place of birth, Hawaii)`. Following Durmus et al. (2020), our QG model is a BART model fine-tuned on (s, a, q) triples annotated by Demszky et al. (2018). Although the QG model is trained on natural language sentences, we found it transferring reasonably well on relational triple data because of their simple format.

4.2 Question Answering

Given a question q, and a context c, the QA model gives the probability of an answer a, modeling $P_{qa}(a|q, c)$. For evaluating a generated output, given a question q generated by the QG model from the reference and factual triples, or the output, the QA model answers it using the output, or reference and factual triples (as context c), respectively. For answering questions using factual triples as context, we concatenate all the linearized triples into a single text. Our QA model is an ALBERT-XL model (Lan et al., 2020) fine-tuned on SQuAD2.0 (Rajpurkar et al., 2018), with F1 score of 87.9% on SQuAD2.0. SQuAD2.0 support identifying *unanswerable* questions, which is crucial as not all answers are found in a given context.

[8] We use *maximum* because a fact contained in the model's output should either be precise w.r.t knowledge triples or the reference, but not necessarily both.

[9] https://spacy.io/

4.3 Answer Matching

The common approach to assess the answers given by a QA model (compared to gold answers) is to use F1-score, which is based on exact matching of n-grams. We argue that is problematic in our case when correct answers are lexically different. For example, Sport:*"professional wrestling"* can be realized as *"She is a wrestler [...]"*. The F1-score does not capture these lexically varied but correct answers. Therefore, we propose using an NLI model to compare the similarity of two answers. Given the generated question q from the reference, to compare the reference (gold) answer and the predicted answer, we concatenate each answer with the question separately to form the premise and hypothesis for the NLI model. For example, for the question *"What sport did Mr. Kenny Jay play?"*, we pass the following to the NLI model:

> `Premise:` *What sport did Mr. Kenny Jay play? professional wrestling*
> `Hypothesis:` *What sport did Mr. Kenny Jay play? wrestler*

We give the predicted answer a score of 1 if the NLI model predicts entailment, and a score of 0 if it predicts contradiction. For neutral, we compute the BERTScore (Zhang et al., 2020b) comparing the contextualized representations of the two answers. For the NLI model, we use RoBERTa (Liu et al., 2019) fine-tuned on MNLI (Williams et al., 2018), with an accuracy of 90% on MNLI. We included examples of comparison between NLI and F1 score in Table 9.

5 ENTDESCRIPTOR

The ENTDESCRIPTOR model needs to fetch relevant passages on the fly to generate a factual description. For this, we equip our ENTDESCRIPTOR model with a *Passage Ranker* (§5.1). Given the entity, keys and a set of ranked passages, the *Descriptor Generator* (§5.2) then generates an entity description.

5.1 Passage Ranker

Each instance in our dataset is accompanied by a set of candidate grounding passages. However, not all passages contain useful knowledge about certain aspects of an entity, i.e., the provided *factual* and *topical keys*. We, therefore, introduce a ranking stage where we rank the grounding passages $\mathcal{P}$ given the entity, title, and a set of keys as query q.

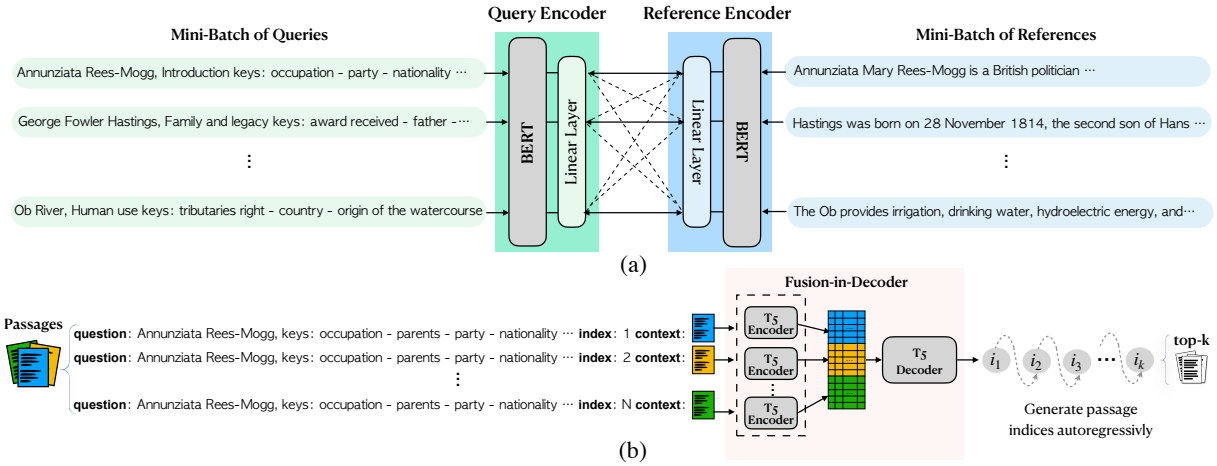

Figure 3: Two proposed passage rankers: (a) Contrastive Dense (b) Autoregressive.

The ranker outputs top-k passages $\{p_1, ..., p_k\} \subset \mathcal{P}$, which are then used to ground the Descriptor Generator. Below, we describe two baseline rankers, namely ROUGE-2 and tf-idf rankers, and two proposed rankers, namely contrastive dense and autoregressive rankers.

ROUGE-2 (*oracle*). This ranker ranks passages according to their ROUGE-2 recall against the reference. This is akin to *oracle* ranking as we use information in the reference to do the ranking.

Tf-idf. This ranker ranks passages using their tf-idf score following Liu et al. (2018).

Contrastive Dense. This ranker learns and then compares dense representations of queries and passages using contrastive training. We train a dense ranker (shown in Fig. 3(a)) which is inspired by recent embedding-based retrieval systems such as REALM (Guu et al., 2020), and DPR (Karpukhin et al., 2020). We follow a distant supervision approach (Jernite, 2020) by using the reference descriptions r instead of the gold passages as supervision signal. For each instance, we form a query q_i by concatenating the entity, title and the set of keys. We thus construct a dataset of (q_i, r_i) pairs and use a bi-encoder architecture to project queries and references to 128-d embedding space. We use a contrastive framework with in-batch negatives where the idea is to push encoded vector of a query closer to its corresponding reference vector, but away from other reference vectors in the batch. Formally, we optimize the following Cross-Entropy loss with in-batch negatives:

$$\mathcal{L} = -\sum_{(q_i,r_i)\in mB} \log \frac{\exp(\mathbf{q_i}.\mathbf{r_i})}{\sum_{r_j \in mB} \exp(\mathbf{q_i}.\mathbf{r_j})} \quad (1)$$

where $\mathbf{q_i}$ and $\mathbf{r_i}$ are encoded query and reference vectors, and mB denotes the mini-Batch. We use mini-batches of 1024, and initialize the encoders with distilled-BERT (Turc et al., 2019; Devlin et al., 2019). Two projection layers are then learned for queries and references. Once the ranker is trained, we use the reference encoder to encode each grounding passage p_i and score them based on their dot product similarity w.r.t vector representation of query $\mathbf{q_i}$. We then use the top-k passages as input to Descriptor Generator.

Autoregressive. In the previous ranker, passages are scored independently according to their relevance to the input query q. However, an ideal ranker should select relevant yet diverse passages. To achieve this goal, we develop an autoregressive ranker with an encoder-decoder architecture (shown in Fig. 3(b)) where the encoder process the entire set of passages $\mathcal{P}$, and the decoder generates a sequence of k passage indices. The autoregressive nature enables modeling the joint probability of passages $P(p_1, ..., p_N|q)$. Similar text-to-index framework showed promising results for *sentence ordering* (Basu Roy Chowdhury et al., 2021) and *multi-answer retrieval* (Min et al., 2021). To enable encoding the entire set of passages (in our case 40), we use the Fusion-in-Decoder (FiD) architecture following Izacard and Grave (2021).

The FiD architecture takes the input query (concatenation of entity, title and set of keys) as well as each individual passage independently as inputs to its encoders. The query is concatenated with each passage and its positional index using special tokens: `question: [Entity]` e `[Title]` t `[Keys]` $\mathcal{K}^f$ + $\mathcal{K}^t$ `index:` i `context:` p_i. The encoders output representations $\mathbf{h_i} \in \mathbb{R}^{L\times d}$ for each individual passage p_i, where L, and d are

input length and hidden dimension, respectively. The concatenation of encoders' representations $\mathbf{H} = [\mathbf{h_1}; ...; \mathbf{h_N}] \in \mathbb{R}^{L.N \times d}$ is passed to decoder which in turn generates a sequence of passage indices $\{i_1, ...i_k\}$. The corresponding top-*k* passages $\{p_{i_1}, ..., p_{i_k}\}$ are then used to ground the Descriptor Generator. All encoders and the decoder are initialized with T5 (Raffel et al., 2020). This ranker is trained using the *silver* sequence of passage indices obtained by ROUGE-2 *(oracle)* ranker.

5.2 Descriptor Generator

Extractive. We build an extractive baseline using QG and QA models. For this, we convert an entity name and each factual key in our input into a natural language question using a seq2seq model. We then use a strong extractive QA model, namely ALBERT-XL fine-tuned on SQuAD2.0, to answer these questions using all grounding passages as the context (Each grounding passage is passed separately). Finally, we concatenate all sentences from the groundings which contained the most confident answers as our final output.

Abstractive. We build strong abstractive baselines by fine-tuning several transformer-based PLMs. This include encoder-decoder models, namely **BART-Large** (Lewis et al., 2020a), **T5-large** (Raffel et al., 2020), and **PEGASUS** (Zhang et al., 2020a). All baselines take inputs in the form of `[Entity]` e `[Title]` t `[Keys]` $\mathcal{K}^f$ + $\mathcal{K}^t$ `[docs]` P^k and generate the entity description. Note that during training our generation models, we use the top-10 grounding passages obtained by *oracle* ROUGE-2 ranker.[10] See Appendix A for details.

6 Experiments

6.1 Automatic Evaluation

Beside our proposed MAFE metric, we use several widely used automatic metrics like **BLEU** (Papineni et al., 2002), and **ROUGE-L** (Lin, 2004). However, recent works (Dhingra et al., 2019) have raised concerns on the usage of these metrics for automatically constructed data-to-text dataset as they fail to consider divergent reference texts. We also use **PARENT** (Dhingra et al., 2019)[11] that considers similarity of generation to both data (in our case factual triples) and the reference. Lastly, we use **BERTScore** (Zhang et al., 2020b) which computes alignment between BERT representations of reference and generated output.

Metric	BLEU	R-L	PAR	BERT-S	MAFE
Corr.	48.00	56.75	40.80	56.76	**60.14**

Table 2: Paragraph-level Pearson correlation coefficient between automatic metrics and human judgement of factuality. All correlations are significant at $p < 0.001$.

6.2 Evaluation of MAFE

We propose a metric for evaluating factuality, MAFE. To evaluate MAFE as a metric, we compute its correlation with human judgments of factuality. We take a random set of 297 BART-L generated outputs using different rankers. The instances include diverse set of entity domains (see Fig. 5 in Appendix). We collect human judgments of factuality on this subset using Amazon Mechanical Turk (AMT). Three annotators judged the recall-oriented and precision-oriented factuality of each generated paragraph. For evaluating recall-oriented factuality, we present each sentence of the reference one at a time and ask annotators how well the sentence is supported by the content in the generated paragraph. The annotators have to choose from a Likert scale of 1-5 (1 being very badly supported, 5 being very well supported).[12] We also present each factual triple one at a time and ask annotators if it is supported by the content in the generated paragraph.[13] For evaluating precision-oriented factuality, we switch references and factual triples with generated paragraphs, i.e., we show the generated paragraphs one sentence at a time and ask how well the sentence is supported by the reference and all factual triples. We then average scores across all sentences. See Appendix C.2 for details and screenshots of annotation layout.

To account for recall and precision oriented values, we measure correlations between human judgment F1 ($2\frac{rec.prec}{rec+prec}$) with MAFE-F1 and other automatic metrics. According to Table 2, MAFE shows a higher correlation with the human judgment than other metrics. Hence, we include MAFE in our experiments to gauge the factuality of generations.

[10] Using the passages obtained from other rankers during training degrades the performance.

[11] We use the co-occurrence version which is recommended when paraphrasing is involved between data and text.

[12] We find the Likert scale to be more suitable than binary decision because each sentence might contain multiple facts.

[13] Factual triples are presented in the form of *e (k; v)*. E.g. *Henry Stanton (placeofburial; West Point)* which is read as "The place of burial of Henry Stanton is West Point."

Model +Ranker	BLEU	R-L	PAR	BERTS	MAFE	
					R	P
Extractive	2.94	19.85	14.22	82.44	19.01	18.91
T5-Large	5.41	15.09	20.58	84.25	17.65	17.38
+Tf-idf	5.50	26.70	21.42	84.53	17.68	18.35
+Contrastive Dense	6.89	28.14	22.44	84.92	19.00	19.66
+Autoregressive	7.24	28.64	23.45	84.96	**19.48**	19.97
+Rouge2 (oracle)	8.56	30.84	25.97	85.41	20.99	22.01
BART-L	7.03	30.82	23.57	86.10	17.43	23.13
+Tf-idf	6.97	31.14	23.96	86.23	17.45	24.02
+Contrastive Dense	8.25	32.46	24.67	86.48	18.55	26.00
+Autoregressive	**8.69**	**32.97**	**25.40**	**86.58**	19.11	**26.71**
+Rouge2 (oracle)	9.82	34.87	27.25	87.01	20.28	29.32
PEGASUS	6.49	27.11	22.88	83.65	15.34	22.72
+Tf-idf	6.34	27.25	22.68	83.74	14.79	23.72
+Contrastive Dense	7.99	28.94	24.10	84.43	16.40	24.70
+Autoregressive	8.55	29.75	25.38	84.54	17.16	25.34
+Rouge2 (oracle)	10.05	31.71	27.72	84.97	18.07	26.73

Table 3: BLEU, ROUGE-L, PARENT, BERTScore, and MAFE scores for different unranked models, as well with adding different rankers. Models consistently perform better when using autoregressive ranker.

Ranker	Recall@5	Recall@10
Tf-idf	32.02	42.35
Contrastive Dense	36.62	45.90
Autoregressive	**44.67**	**52.08**

Table 4: Recall@k (%) for different rankers w.r.t *oracle* ranking.

6.3 Results

Performance of Different Baselines. Table 3 reports the performance of different baselines for the task of entity description generation. According to the results, Extractive performs poorly compared to other abstractive baselines. This is mainly because it lacks the narrative flow required for a coherent output. Comparing all abstractive baselines, when they are given *oracle* groundings (defined in §5.1), shows that BART outperforms T5 and PEGASUS in general on all n-gram overlap-based, PARENT, as well as BERTScore metrics. Uni/bi-gram overlap (R-1,R-2) are reported in Table 8.

When comparing baselines with respect to factuality using our MAFE metric, we see that BART in general generates paragraphs that are significantly more consistent (precise) with respect to factual triples and reference. Whereas, T5 is slightly better at content-selection (measured by recall).

Performance of Different Rankers. We now investigate the effect of different rankers on generation performance. For this, we compare baselines using different rankers (see Table 3). All models perform better when they are given top-k ranked groundings than their *Unranked* baselines. For all generation models, the proposed contrastive and autoregressive rankers significantly outperform the tf-idf baseline ranker. This is because tf-idf ranker only finds passages that feature sparse words from the input query and fails to capture semantic similarities. Moreover, by predicting a sequence of passages each conditioned on the previously selected passages in the autoregressive ranker, the generation model gains further improvements over the strong contrastive dense ranker. We also compare Recall@k for different rankers w.r.t the *oracle* ranking in Table 4. The score indicates the proportion of *oracle* passages (obtained y ROUGE-2 method) that is found in the top-k predicted passages by any of the rankers. We find that autoregressive outperforms the other two rankers.

6.4 Human Evaluation

Here, we evaluate factuality and faithfulness of generated descriptions on AMT.

Factuality ($r \leftrightarrow h$). We evaluate the factuality of generated paragraphs using human annotators. We randomly sample 100 datapoints from the test set and evaluate paragraphs generated by BART-L using four rankers: tf-idf, contrastive dense, autoregressive and ROUGE-2 *(oracle)* (a total of 400 generation examples). We ask 3 judges from AMT to evaluate the recall-oriented and precision-oriented factual correctness of each sample generation. We use the same annotation layout described for evaluating MAFE metric (correlation analysis; §6.1). More details can be found in Appendix C.2.

Table 5 shows that human annotators consistently rate the factuality of paragraphs generated using autoregressive ranker higher than those generated using contrastive dense ranker and lower than *Oracle* ranker. The result is consistent with our proposed metric as well.

Faithfulness ($P^k \rightarrow h$). We also evaluate whether the generated outputs are faithful to the top-k grounding passages.[14] For this, we randomly sample 100 data points from the test set and ask 3 annotators from AMT to evaluate the faithful-

[14]Here, we are evaluating faithfulness wrt input groundings. Thus, we use the same set of groundings (by fixing the ranker to be autoregressive) and evaluate different underlying LM.

Ranker	Recall	Precision
Tf-idf	48.95	43.36
Contrastive Dense	51.98	57.50
Autoregressive	**56.76**	**58.41**
Rouge2 (oracle)	58.92	62.00

Table 5: Human evaluation of factuality (recall- and precision-oriented in %) for BART-L generated paragraphs using different rankers.

	T5-Large	BART-L	PEGASUS
Human Rating	**4.17**	3.53	3.83

Table 6: Human evaluation of faithfulness of different baselines w.r.t grounding passages. Scores are on a scale of 1 (very poor) to 5 (very high).

ness of generated outputs using different baselines on a scale of 1-5. Following our previous annotation layout, we show one sentence at a time and then average scores across all sentences. Table 6 shows that T5 generates more faithful paragraphs compared to other baselines.

6.5 Ablation Studies

Here, we discuss different ablations of our task where we remove/add certain information from/to the input and investigate its effect on the performance. We experiment with settings where there are *no groundings*, *no keys*, *no factual keys*, *no topical keys*, *values w/o groundings*, and *values w/ groundings*.

Table 7 shows the results for the BART-L baseline with the autoregressive ranker. As expected, the model performance degrades the most w.r.t all metrics when the grounding passages are removed from the input. This setting is similar to the prompt-to-text generation, where the model mostly relies on its parametric knowledge and is prone to hallucination. Removing all the keys from the input is detrimental in recalling important information, as shown from the MAFE-R score. We also observe that ablating factual keys hurts the relevance of the generated paragraph (i.e., Recall) w.r.t its reference more, whereas ablating topical keys hurts the n-gram overlapping metric (R-L). This is because factual keys are essential to make a good content selection and be rewarded by MAFE metric, whereas topical keys mostly appear verbatim in the output. Lastly, having the gold values for the corresponding keys without the grounding passages cannot beat the performance with the original inputs. In particular, although the model can recover more information (i.e. better recall), not being grounded causes it to generate less consistent information (i.e. lower precision). This is in line with our previous findings where passages play an important role in achieving good performance. When accompanied with groundings, the model achieves the best performance, emphasizing the importance of grounding.

Ablated Inputs	R-L	BERT-S	MAFE	
			Recall	Precision
Grounding Passages				
no groundings	25.44	84.80	8.87	13.01
Keys				
no keys	30.34	85.95	17.99	27.12
no factual keys	**31.92**	**86.35**	18.21	27.14
no topical keys	31.67	86.33	**19.13**	**28.15**
Orig. Task Input	32.97	86.58	19.11	26.71
Values & Grounding Passages				
values w/o groundings	*28.82*	*85.90*	*19.30*	*25.97*
values w/ groundings	*33.61*	*86.70*	*21.77*	*29.40*

Table 7: Ablation study: Best results are in bold. The gray section is when values are assumed to be at hand, and akin to *oracle* experiment.

7 Related Work

Natural Language Generation. Several data-to-text problems have been proposed with various input formats like Knowledge Graphs (Koncel-Kedziorski et al., 2019; Cheng et al., 2020), Abstract Meaning Representations (Flanigan et al., 2016; Ribeiro et al., 2019), tables and tree structured semantic frames (Bao et al., 2018; Chen et al., 2020; Parikh et al., 2020; Chen et al., 2021; Nan et al., 2021), and Resource Description Framework (Gardent et al., 2017).

Towards a more controlled generation task, ToTTo (Parikh et al., 2020) was introduced for an open-domain table-to-text generation where only some of the cells are selected as the input. However, ToTTo and most existing datasets such as WIKIBIO (Lebret et al., 2016) and LogicNLG (Chen et al., 2020) focus on generating single sentences. Although generating long-form text is becoming

a new frontier for NLP research (Roy et al., 2021; Brahman et al., 2021), not many datasets and tasks have been proposed to explore this new direction. Available datasets such as ROTOWIRE (Wiseman et al., 2017) or MLB (Puduppully et al., 2019) are either small-scale or on single domain (e.g., Sports). Unlike prior works, we propose a long-form grounded keys-to-text generation task that covers multiple domains and categories, including people, location, organization, event, etc.

Recently, Chen et al. (2021) presented the WIKITABLET dataset for long-form text generation from multiple tables and meta data. However, this setting is overpecified because knowledge about entities may not always be available in structured format and may get updated in real-time. In a more natural setting, our ENTDEGEN dataset uses factual and topical keys as guidance but still leaves a considerable amount of content selection from grounding passages to be done by the model.

There has been several work on open-ended NLG (e.g., prompt-to-text or outline-to-text) (Fan et al., 2018; Xu et al., 2018; Yao et al., 2019; Rashkin et al., 2020; Brahman et al., 2020). Our task is also closely related to query-focused multi-document summarization (Xu and Lapata, 2020, 2022) which relies on retrieval-style methods for estimating the relevance between queries and text. Additionally, our task setup can benefit from evaluation methods in summarization domain.

Factual Consistency Evaluation. Evaluating factual consistency of machine-generated outputs has gained growing attention in recent years. New approaches have been proposed mainly for tasks like abstractive summarization and machine translation (Zhang et al., 2020b; Sellam et al., 2020; Durmus et al., 2020). Some of these metrics are QA based and have been used to measure common information between documents/reference and summaries (Eyal et al., 2019; Scialom et al., 2019; Wang et al., 2020). Our proposed metric, MAFE, is inspired by these works.

8 Conclusion

We present a practical task of grounded *keys-to-text* generation and construct a large-scale dataset ENTDEGEN to facilitate research on this task. Experiments show the effectiveness of the proposed rankers to fetch relevant information required to generate a factual description. The human evaluation shows that ENTDEGEN poses a challenge to state-of-the-art models in terms of achieving human-level factuality in long-form generation. Our proposed dataset and task can also foster further research in the recently emerging retrieval augmented generations models (Lewis et al., 2020b; Zhang et al., 2021; Shuster et al., 2021) – where the retriever and generator components are trained end-to-end.

Limitations

One of the limitations of our work is the reliance on a strong retriever/ranker. A weak retriever may result in generating text that are less factual and thus less thrust-worthy. While we proposed efficient and simple methods for training the retriever, these require large GPUs. Additionally, as the retrieved passages get longer the quality of text generation may degrades due to known issues with encoding longer sequences.

Acknowledgments

We thanks our anonymous reviewers, Felix Faltings, and members of the DL and NLP group at Microsoft Research for their constructive feedback. This work was supported in part by NSF grant IIS-2047232.

References

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, M. Zhou, and T. Zhao. 2018. Table-to-text: Describing table region with natural language. In *AAAI*.

Somnath Basu Roy Chowdhury, Faeze Brahman, and Snigdha Chaturvedi. 2021. Is everything in order? a simple way to order sentences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10769–10779, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "Let Your Characters Tell Their Story": A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Faeze Brahman, Alexandru Petrusca, and Snigdha Chaturvedi. 2020. Cue me in: Content-inducing approaches to interactive story generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference*

on Natural Language Processing, pages 588–597, Suzhou, China. Association for Computational Linguistics.

Mingda Chen, Sam Wiseman, and Kevin Gimpel. 2021. WikiTableT: A large-scale data-to-text dataset for generating Wikipedia article sections. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209, Online. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Liying Cheng, Dekun Wu, Lidong Bing, Yan Zhang, Zhanming Jie, Wei Lu, and Luo Si. 2020. ENT-DESC: Entity description generation by exploring knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1187–1197, Online. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *ArXiv*, abs/1809.02922.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*.

Yacine Jernite. 2020. Explain anything like i'm five: A model for open domain long form question answering.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2012. Unsupervised concept-to-text generation with hypergraphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 752–761, Montréal, Canada. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, D. Kontokostas, Pablo N. Mendes, Sebastian Hellmann, M. Morsey, Patrick van Kleef, S. Auer, and C. Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

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

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *CoRR*, abs/1801.10198.

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

Kathleen McKeown. 1985. *Text Generation*. Studies in Natural Language Processing. Cambridge University Press.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6997–7008, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

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.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. *CoRR*, abs/1909.00352.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse

attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proc. of EMNLP-IJCNLP*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and J. Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *ArXiv*, abs/2104.07567.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. In *Proc. of EMNLP*, pages 2153–2162, Austin, Texas. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark.

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

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315.

Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3632–3645, Online. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. 2022. Document Summarization with Latent Queries. *Transactions of the Association for Computational Linguistics*, 10:623–638.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

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

Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2021. Joint retrieval and generation training for grounded text generation. *arXiv preprint arXiv:2105.06597*.

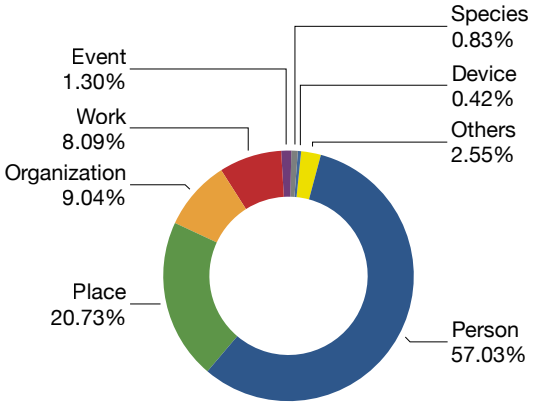

Figure 4: ENTDEGEN Entity Domain Distribution.

Model	Ranker	R-1	R-2	R-L
Extractive	n/a	22.37	6.28	19.85
T5-Large	Unranked	28.74	10.31	26.8
	Tf-idf	29.49	10.54	26.70
	Contrastive Dense	30.92	12.19	28.14
	Autoregressive	31.45	12.59	28.64
	Rouge2 (oracle)	33.75	14.84	30.84
BART-L	Unranked	33.52	14.39	30.82
	Tf-idf	34.00	14.54	31.14
	Contrastive Dense	35.26	16.04	32.46
	Autoregressive	**35.80**	**16.62**	**32.97**
	Rouge2 (oracle)	37.72	18.57	34.87
PEGASUS	Unranked	29.23	12.46	27.11
	Tf-idf	29.43	12.38	27.25
	Contrastive Dense	31.21	14.16	28.94
	Autoregressive	32.03	14.90	29.75
	Rouge2 (oracle)	34.01	17.03	31.71

Table 8: ROUGE scores (Lin, 2004).

A Implementation Details

Baselines. Top-10 grounding passages were used to train and test all baselines. We use the Transformer library (Wolf et al., 2019). Each baseline was trained for 3 epochs with effective batch size of 8, and initial learning rate of 5e-6 for T5 and BART, and 1e-4 for PEGASUS. We use the maximum input length of 512 tokens. During inference, we use beam search decoding with 5 beams, and repetition penalty of 1.2. Note that we use the BART-L model finetuned on XSUM dataset as our initial weights. Similarly, we use google's PEGASUS model finetuned on XSUM. The experiments are conducted in PyTorch framework using Quadro RTX 6000 GPU.

Rankers. The contrastive dense ranker was trained for 10 epochs with 2e-4 learning rate. The autoregressive ranker was trained for total of 30,000 steps with learning rate and weight decay of 1e-5 and 0.01, respectively. Rankers were trained using 4x Nvidia V100 GPU machines, each with 32G memory.

Question Generation in MAFE. The question generation module (QG) in MAFE evaluation metric, generates questions using beam search decoding with beam size of 10.

B Dataset Quality Assessment

We conducted a human evaluation on Amazon Mechanical Turk to assess the quality of our automatically constructed dataset. In this experiment, we randomly sample 100 examples from the test set. For each example, we ask 3 annotators to read the reference description carefully and answer whether each of the factual key and value pair is stated in the description or can be implied by the description. We then take the majority vote between the annotations. The result shows that 74% of reference descriptions contain information about more than half of the key-value pairs, with Fleiss' Kappa of 0.53 showing moderate agreement.

C Experimental Results

C.1 Automatic Evaluation

We report all ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap) and ROUGE-L (longest matching sequence) scores in Table 8.

C.2 Human Evaluations

For all the human evaluations, we restricted the pool of workers to those who were located in the US, or CA, and had a 95% approval rate for at least 1,000 previous annotations. Additionally, to further ensure the quality of annotations, we only hired master turkers, i.e., high performing turkers who have demonstrated excellence across a wide range of tasks and are awarded Masters Qualification. We also designed our setup to avoid annotator fatigue by asking them to read each paragraph only once and continuously answer several questions about it. We use a pay rate of $15 per hour approximately based on our estimation of time needed to complete the task.

We depict our annotation layouts for evaluating precision-oriented and recall-oriented (both w.r.t reference and factual triples) factuality in Fig-

Figure 5: EntDeGen Entity Domain Distribution of Correlation Analysis Subset.

ure 6, 7, and 8. Likert scale of 1-5 and binary scores (supported/not supported) are used when evaluating recall w.r.t references, and factual triples, respectively. These scores are then normalized and averaged to obtain the final recall-oriented score.

Answers	**F1**	**NLI**
gold: "saxophone" predicted: "saxophonist"	0.0	1.0
gold:"an american lawyer" predcited:"an american politician"	0.66	0.0
gold: "st frideswide 's priory" predicted: "priory of st frideswide"	0.75	1.0
gold: "december 30 , 1995" predicted: "december 31 , 1995"	0.75	0.0
gold: "the united kingdom" predicted: "united kingdom"	0.8	1.0
gold: "his son, malcom" predicted: "malcom"	0.4	1.0
gold: "species survival plans" predicted: "captive breeding programs"	0.0	0.89
gold: "rio de janeiro" predicted: "rio de janeiro , brazil"	0.74	1.0
gold: "liberal party" predicted: "conservative party"	0.5	0.0

Table 9: Examples of comparison between F1 and NLI scores.

Figure 6: An illustration of human evaluation of precision-oriented factuality. Generated paragraphs are presented one sentence at a time and are evaluated on how well they are supported by the references.

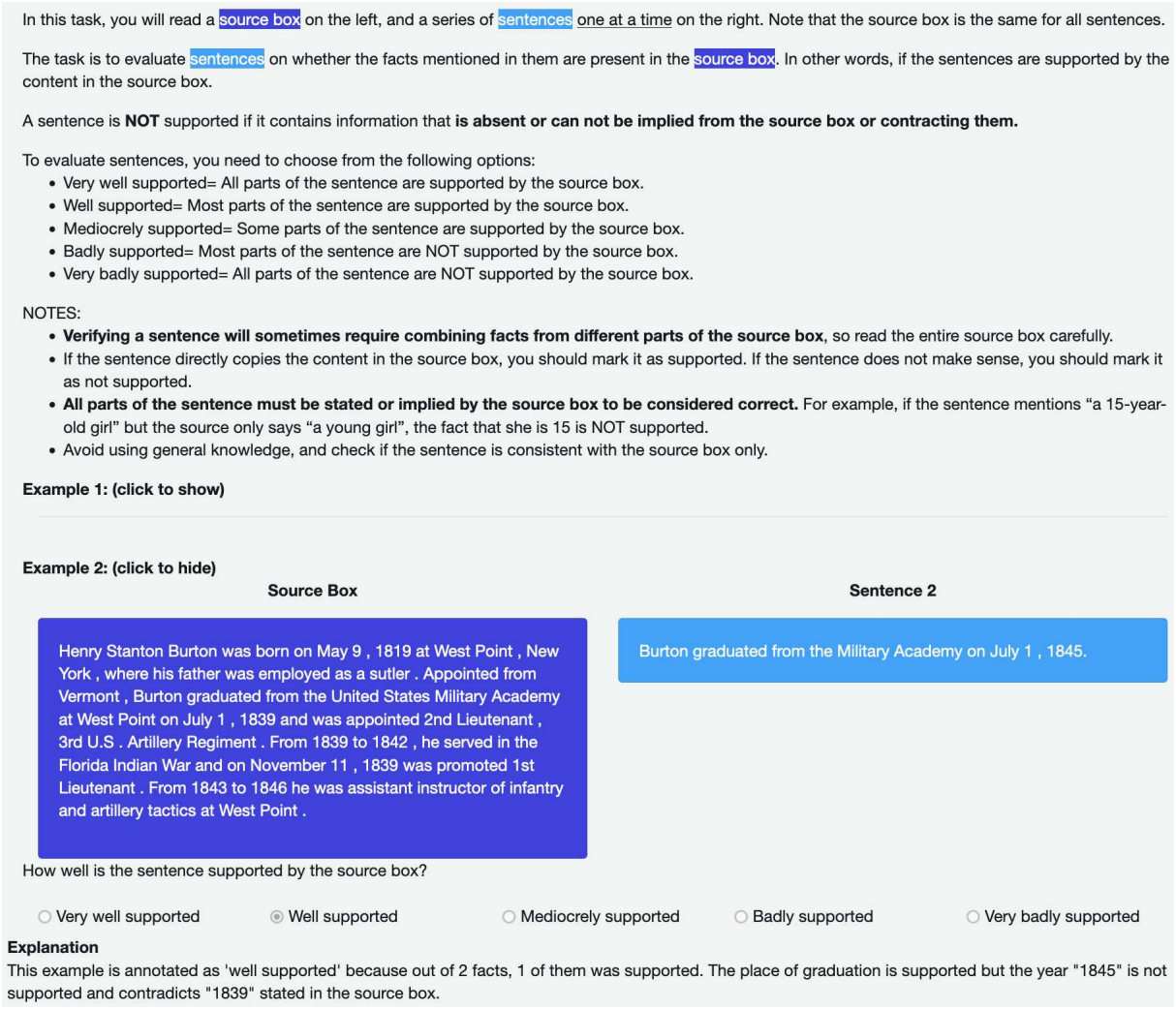

Figure 7: An illustration of human evaluation of recall-oriented factuality w.r.t reference. References are presented one sentence at a time and are evaluated on how well they are supported by the generated paragraphs.

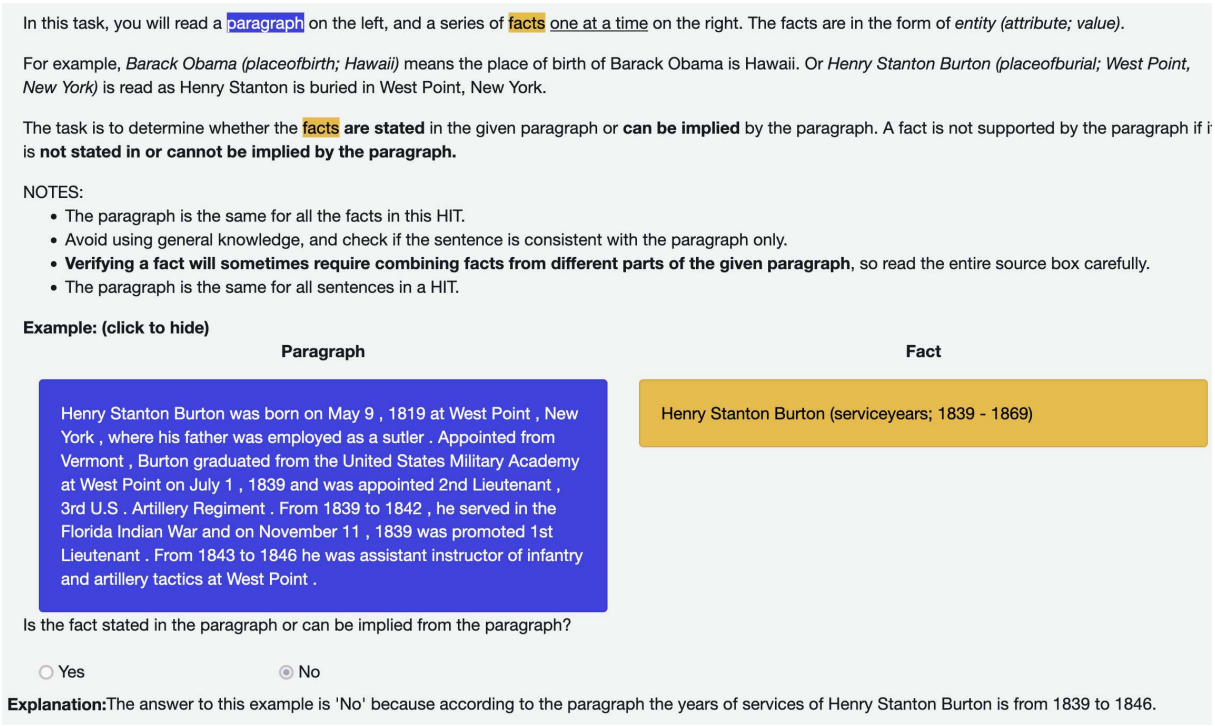

Figure 8: An illustration of human evaluation of recall-oriented factuality w.r.t factual triples. Factual triples are presented one at a time and are evaluated on whether they are supported by the generated paragraphs or not.