

INSET: Sentence Infilling with INter-SEntential Transformer

Yichen Huang (黄溢辰)^{1,2*}, Yizhe Zhang^{1*}, Oussama Elachqar¹, Yu Cheng¹

¹Microsoft Corporation, Redmond, Washington 98052, USA

²Center for Theoretical Physics, MIT, Cambridge, Massachusetts 02139, USA

yichuang@mit.edu, {yizhe, ouelachq, yu.cheng}@microsoft.com

Abstract

Missing sentence generation (or sentence infilling) fosters a wide range of applications in natural language generation, such as document auto-completion and meeting note expansion. This task asks the model to generate intermediate missing sentences that can syntactically and semantically bridge the surrounding context. Solving the sentence infilling task requires techniques in natural language processing ranging from understanding to discourse-level planning to generation. In this paper, we propose a framework to decouple the challenge and address these three aspects respectively, leveraging the power of existing large-scale pre-trained models such as BERT and GPT-2. We empirically demonstrate the effectiveness of our model in learning a sentence representation for generation and further generating a missing sentence that fits the context.

1 Introduction

Generating a span of missing tokens in a text chunk, known as “text infilling,” has attracted many attentions recently (Zhu et al., 2019; Song et al., 2019; Liu et al., 2019; Ippolito et al., 2019; Joshi et al., 2020). Here we study the related but somewhat different task of “sentence infilling.” Specifically, as illustrated in Figure 1, intermediate sentences (or chunks of text) are removed from long-form text (e.g., paragraphs, documents), and the task is to generate the missing pieces that can smoothly blend into and fit the context both *syntactically* and *semantically*. The generation can be either based only on context, or based on both context and side information such as constraint keywords. Compared with text infilling, sentence infilling requires the model to handle inter-sentential correlation and to reason about missing semantic information. Developing models for sentence infilling can potentially

*These authors contributed equally to this work.

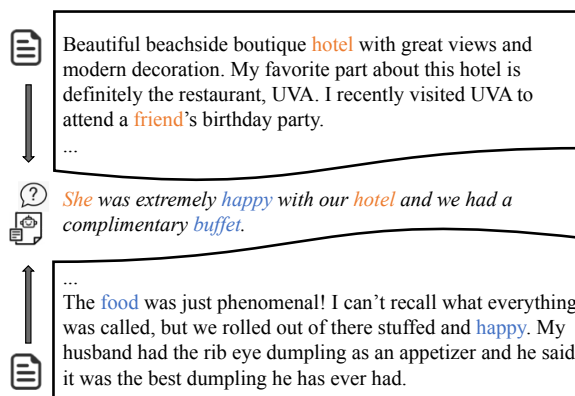


Figure 1: Sentence infilling: generating an intermediate sentence that provides a smooth semantic transition from the preceding to the following context. This example is generated by our model on the TripAdvisor dataset. The colors mark the correspondence between the generated sentence and the context.

facilitate many text generation applications. Possible scenarios include, but are not limited to: document auto-completion by detecting and suggesting missing bridging sentences in the surrounding context; collaborative document writing by modifying and unifying different writing styles from multiple authors; meeting note expansion by extending a set of keywords (lexical constraints) to a full sentence, leveraging the surrounding context.

There are many challenges associated with this long-form sentence infilling task, which is typically a one-to-many problem in that the possible outputs can be diverse. As the generated sentence should connect separate text pieces in a syntactically and semantically smooth and coherent manner, the task requires a wide range of *understanding*, *planning*, and *generation* techniques. Large-scale pre-trained language models such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) have dramatically enhanced the understanding and generation modules. However, how to in-

tegrate them in a holistic manner, and to analyze and establish the long-range dependence structure by high-level semantic planning is still challenging and yet to explore, as semantic appropriateness is usually subtler than syntactic appropriateness, which can be well characterized by autoregressive language models.

Several works have been done in this direction. MASS (Song et al., 2019) obtains sentence representations by predicting a span of missing tokens. It can be used to generate missing text, but the missing span length needs to be pre-specified. Other related works (Liu et al., 2019; Joshi et al., 2020) also require knowledge of the span length as an input to their models, and thus are different from our work. Text infilling (Zhu et al., 2019) sequentially generates tokens for the missing part of a sentence until an end-of-blank token is generated. Its generation can be of arbitrary length. By design, all these previous approaches operate at the token level, and thus arguably focus more on lexical appropriateness than the global semantics.

In this paper, we propose INter-SENTential Transformer (INSET), a novel approach to sentence infilling. Our model first produces sentence-level semantic features that capsule the missing high-level information. Then, grounded on the predicted semantic features, the model generates the syntactic and lexical features to embody the predicted sentence. Specifically, *understanding*, *planning*, and *generation* are handled by three modules in a synergistic manner:

- a *BERT-based encoder* to map each sentence to the latent semantic space.
- a *sentence-level semantic planner* to infer the missing information that can bridge the semantics of preceding and following context.
- a *GPT-based generator* (decoder) to map semantic features back to the text domain.

The main contributions and advantages of this work are summarized as follows:

- We study the task of sentence infilling, which requires the model to handle inter-sentential correlation and to predict missing semantic information. This goes beyond text infilling (Zhu et al., 2019), which asks the model to fill in the missing part of a single sentence.
- Our approach decouples understanding, planning, generation, and leverages existing large-scale pre-trained understanding and generation models (BERT, GPT-2). The components

of our model can be separately examined and improved with additional data.

- Our model predicts a feature vector in the latent semantic space for the missing sentence and maps the vector to text. Thus, it takes care of semantic smoothness and appropriateness.
- Our model allows the generation to be of arbitrary length, as in (Zhu et al., 2019).
- Compared with directly processing text, our approach significantly reduces computation time and memory usage during training, as (after pre-computing sentence features) the sequence length is the number of sentences rather than that of tokens.

2 Related Work

Pre-Trained Language Model. Language models pre-trained on a large corpus improve natural language understanding and generation through transferable contextualized word representations (Devlin et al., 2019; Lample et al., 2019) and models (Howard and Ruder, 2018). Large transformer models (Vaswani et al., 2017) like GPT-2 (Radford et al., 2019), Megatron (<https://github.com/NVIDIA/Megatron-LM>), and T5 (Raffel et al., 2019) can achieve state-of-the-art results without training on any particular language modeling benchmark. (Keskar et al., 2019) proposes a conditional generation model, trained to condition on control codes that govern style, content, and other task-specific properties. Different from them, our model builds sentence representations via autoencoding with a pair of BERT and GPT-2.

Context-Aware Text Generation. There are some related works on context-aware text generation (Mikolov and Zweig, 2012; Tang et al., 2016; Mangrulkar et al., 2018). Most previous works on language modeling with contextual information (Wang and Cho, 2016; Wang et al., 2018; Sordani et al., 2015b; Wen et al., 2015; Vinyals and Le, 2015) treat the preceding sentences as context. Compared with these sequential generation tasks, our task is constrained by bidirectional context, and is more challenging.

Text infilling (Zhu et al., 2019) aims at filling in the missing part, given the rest of a sentence. (Liu et al., 2019) proposes an iterative inference algorithm based on gradient search for text infilling. For story infilling, (Ippolito et al., 2019) first predicts rare words in the missing span, and then generates text conditioned on these words. SpanBERT (Joshi

et al., 2020) masks random contiguous spans and (pre-)trains a language model to predict tokens in the span. XL-Editor (Shih et al., 2019) adapts XL-Net (Yang et al., 2019) to text infilling and other editing tasks.

(Kang and Hovy, 2019) models logic connections between sentences and generates intermediate sentences grounded on inter-sentential “flow.” (Bhagavatula et al., 2020) formulates abductive commonsense reasoning as a natural language inference task to decide the appropriate reason that could explain the observation in one sentence given the background described by another sentence. (Cheng et al., 2020) proposes a text style transfer task to translate a sentence in the context of a paragraph into the desired style. These three works study generation tasks that address inter-sentential relationship, and thus may be conceptually related to our motivation.

Compared with (Zhu et al., 2019; Liu et al., 2019; Ippolito et al., 2019; Joshi et al., 2020; Shih et al., 2019; Kang and Hovy, 2019; Bhagavatula et al., 2020; Cheng et al., 2020), our approach is clearly different. We fully exploit existing large-scale pre-trained models BERT and GPT-2 to learn smooth sentence embeddings in the latent semantic space, and then process sentence-level information in this space.

Hierarchical Text Generation. Hierarchical text generation with high-level semantic planning has been studied in many previous works. (Sordani et al., 2015a) presents a hierarchical recurrent encoder-decoder architecture for context-aware query suggestion. (Zhang et al., 2019) proposes a framework to infer semantic features for response generation using self-supervised learning. Previous works have used multi-level LSTM encoders (Yang et al., 2016; Hu et al., 2020) and hierarchical autoencoders (Li et al., 2015) to learn hierarchical representations for long text. (Shen et al., 2019) uses a variational autoencoder to encode an entire paragraph into a single latent variable, from which the paragraph can be generated hierarchically. In comparison, our task is to generate intermediate sentences in the surrounding context.

3 Tasks and Methods

3.1 Task Definition

The task of sentence infilling is formally defined as follows. Consider a dataset of N

paragraphs $\{p^{(k)}\}_{k=1}^N$. Each paragraph $p^{(k)} = (s_1^{(k)}, s_2^{(k)}, \dots, s_{M_k}^{(k)})$ consists of M_k consecutive sentences. For each k , we are given a positive integer $m_k \leq M_k$ and the context $(s_1^{(k)}, s_2^{(k)}, \dots, s_{m_k-1}^{(k)}, s_{m_k+1}^{(k)}, \dots, s_{M_k}^{(k)})$, but the m_k ’th sentence $s_{m_k}^{(k)}$ is missing. The task is to generate a sentence $\hat{s}_{m_k}^{(k)}$ in the missing position such that it fits the context. For simplicity and without any confusion, we drop the index k from now on (note that M and m may depend on k).

The criteria for successful generation are:

- The sentence \hat{s}_m is fluent and meaningful.
- Inserting the generated sentence into the context, we obtain a semantically coherent paragraph $(s_1, s_2, \dots, s_{m-1}, \hat{s}_m, s_{m+1}, \dots, s_M)$.
- \hat{s}_m is written in the same style as contextual sentences $\{s_j\}_{j \neq m}$.

Since there could be multiple semantically different sentences that fit the same context well, it is not necessary for \hat{s}_m to be close to the ground truth s_m . Rather, \hat{s}_m is considered successful as long as it satisfies the criteria above.

3.2 INSET: Inter-Sentential Transformer

Model Overview. At a high level, our model consists of two components: a (denoising) *autoencoder* and a *sentence-level transformer*. The former maps each sentence to a fixed-length feature vector in the latent semantic space, and reconstructs the sentence from the representation. The latter predicts the semantic features of the missing sentence from those of contextual sentences. We call our model INter-SEntential Transformer (INSET).

Formally, let $(\mathcal{E}, \mathcal{D})$ be an autoencoder, where \mathcal{E} (\mathcal{D}) is the encoder (decoder) such that $\mathcal{E} \circ \mathcal{D}$ and $\mathcal{D} \circ \mathcal{E}$ are supposed to be identity maps. Let \mathcal{T} be a sentence-level transformer with positional encoding \mathcal{P} . The transformer \mathcal{T} takes the contextual information as input and outputs a hypothetical representation of the missing sentence. Specifically,

$$\hat{s}_m = \mathcal{D}(\mathcal{T}(f_1 + \mathcal{P}(1), f_2 + \mathcal{P}(2), \dots, f_{m-1} + \mathcal{P}(m-1), \vec{0} + \mathcal{P}(m), f_{m+1} + \mathcal{P}(m+1), \dots, f_M + \mathcal{P}(M))[m])), \quad (1)$$

where $f_j = \mathcal{E}s_j$ is the encoding of the sentence s_j , $\vec{0}$ is the zero vector representing the missing sentence, and $\mathcal{T}(\dots)[m]$ is output of the transformer \mathcal{T} in the missing position m .

The autoencoder and the sentence-level transformer can be trained separately. We first train the

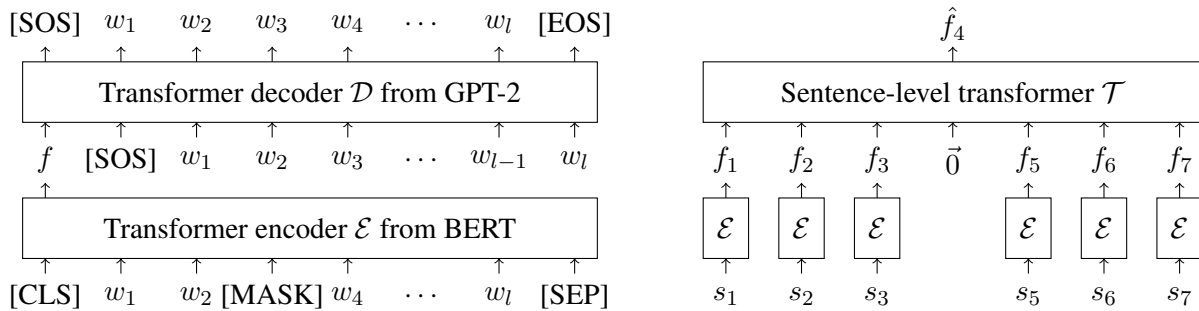


Figure 2: Model overview. Left panel: Denoising autoencoder. The encoder \mathcal{E} takes a corrupted sentence (with each token w_i for $i = 1, 2, \dots, l$ masked randomly) as input and outputs a representation of the sentence. The decoder \mathcal{D} should reconstruct the original uncorrupted sentence. The training parameters of \mathcal{E} and \mathcal{D} are initialized with those of BERT and GPT-2, respectively. Right panel: Sentence-level transformer. Using the encoder \mathcal{E} , we obtain the representation of every contextual sentence. These sentence representations are fed into a sentence-level transformer \mathcal{T} , which outputs a representation of the missing sentence.

former on individual sentences. Then, we precompute and save the feature vectors of all sentences. While training the latter, it is not necessary to load the former. This makes training more efficient.

Sentence Representation Learning via Denoising Autoencoding. Large-scale pre-training approaches (e.g., BERT) lead to superior performance in many language understanding tasks related to sentence representation learning (Reimers and Gurevych, 2019). However, the features learned by BERT (or fine-tuned on downstream tasks) cannot be directly used for generation tasks, as the masked language model objective of BERT does not enforce the reconstruction of the original sentence from the extracted features. Instead of directly using BERT features, we learn sentence representations via autoencoding. This naturally integrates BERT and GPT-2, and combines sentence representation learning and generation.

As shown in the left panel of Figure 2, we pad the [CLS] token at the beginning of each sentence s_j . We initialize the encoder \mathcal{E} with BERT, and extract the output f_j corresponding to the [CLS] token as the embedding of s_j . We initialize the decoder \mathcal{D} with GPT-2, and feed f_j as the embedding of the zeroth token. Then, we have \mathcal{D} generate a sequence of tokens in the hope that the sequence matches s_j (padded with special tokens [SOS] at the beginning and [EOS] at the end). To train the autoencoder, we use teacher forcing and minimize the negative log-likelihood loss by (fine-)tuning the parameters of \mathcal{E} and \mathcal{D} jointly.

An autoencoder embeds sentences into vectors in the latent space. We hope that the embedding is smooth in the sense that semantically similar

sentences are mapped to vectors that are close to each other. In particular, interpolation between two points in the latent space should correspond to a smooth semantic transition in the text domain. To this end, we use the following two tricks.

First, we employ a denoising autoencoder, which is known to yield a smoother embedding (Vincent et al., 2008). To add noise, we randomly mask each token in s_j with probability 15% by replacing the masked tokens with a special token [MASK]. During training, we use the “noisy” s_j with masks as input to the encoder, and use the “clean” s_j without masks to compute the loss function. Of course, one could try more sophisticated noise-adding strategies (Lewis et al., 2019).

Second, we use early stopping. In our experiments, we observe that as training proceeds, the validation loss of the autoencoder keeps decreasing. In the absence of masks, presumably it would eventually decay to zero so that the autoencoder perfectly reconstructs every sentence. However, this does not necessarily imply that the embedding is smooth. On the contrary, an overtrained autoencoder often tries to remember every individual token and thus fails to achieve smoothness in the latent semantic space. Moreover, it can catastrophically forget some of the information in the initial pre-trained model (GPT-2) and partially lose the power of generating fluent sentences. In practice, we select a checkpoint by monitoring its validation performance on sentence interpolation. Some examples of sentence interpolation are shown in Table 1.

Sentence Feature Prediction. After encoding sentences into feature vectors, we use a sentence-

level transformer \mathcal{T} to predict the feature vector of the missing sentence from those of contextual sentences. This is analogous to the task of predicting masked tokens for (pre-)training BERT (Devlin et al., 2019), but now it is at the sentence level. Indeed, sentence feature vectors in \mathcal{T} correspond to token embeddings in BERT, and sentence position ID in \mathcal{T} corresponds to position ID in BERT.

We train the transformer \mathcal{T} with the objective

$$\mathcal{L}_{\text{SentTrans}} = 1 - \cos(f_m, \mathcal{T}(\dots)[m]), \quad (2)$$

where $\cos(\dots)$ is the cosine similarity between the ground truth sentence feature vector f_m and the prediction $\mathcal{T}(\dots)[m]$ in Eq. (1). Note that $\cos(\dots)$ is a good similarity measure only when its arguments are unit vectors. This is guaranteed by the technical trick of fixing the parameters of the last LayerNorm of the transformers \mathcal{E} and \mathcal{T} , i.e., do not compute the gradients of these parameters in backpropagation.

Generating Sentences from Features. At test time, we use the decoder \mathcal{D} to generate the missing sentence by mapping the predicted feature vector to the text domain. Note that standard generation schemes such as top- k sampling, beam search, and nucleus sampling (Holtzman et al., 2020) can be used without additional modeling effort.

Computational Efficiency. Compared with vanilla GPT-2, our model can process and analyze a document containing many sentences at the discourse level with dramatically lower time and space complexity. To estimate quantitatively, suppose that a document contains N_s sentences, each of which has N_t tokens. Then, the time complexity is reduced from $\mathcal{O}(N_s^2 N_t^2)$ to $\mathcal{O}(N_s^2 + N_s N_t^2)$. Moreover, sentence features can be precomputed once and then reused for every epoch or even in other tasks on the same dataset. If sentence features have been precomputed and are already directly available, the time complexity is further reduced to $\mathcal{O}(N_s^2)$.

3.3 Sentence Infilling with Lexical Constraints

We further introduce a related task called *sentence infilling with lexical constraints*, which is the same as sentence infilling except that now we are given some keywords of the missing sentence as an additional input to hint the generation. The keywords are treated as *soft* constraints (aka priming): The

generated sentence is not directly enforced to contain the exact keywords. It may contain a synonym or share some semantics with the keywords.

We expect that the presence of keyword constraints makes the task more difficult rather than easier, although incorporating keywords can significantly improve the BLEU score of the generation with respect to the ground truth. Intuitively, keywords force the model to speculate the semantics of the ground truth sentence, and significantly reduce the number of possible solutions. In the absence of keywords, the model has the freedom of completing the task according to its own way of thinking.

To handle keyword constraints, we introduce a new component called the *constraint feature encoder* to our architecture. It is a transformer encoder \mathcal{K} that maps a set S of keywords to a feature vector that lives in the same latent space of sentence embeddings. We train \mathcal{K} with knowledge distillation (Kim and Rush, 2016). The teacher model is the sentence encoder \mathcal{E} , which maps a sentence containing the keywords in S to a feature vector. We use the cosine similarity loss between these two feature vectors to teach the student model \mathcal{K} .

For implementation details, suppose we have two keywords w_1 and w_2 . Then, the input to \mathcal{K} is three tokens ([CLS], w_1 , w_2). We replace the zero vector in Eq. (1), which represents the missing sentence, with the output of \mathcal{K} above the [CLS] token. We do not use positional encoding in \mathcal{K} because keywords do not have order.

4 Experiments

4.1 Experimental Setup

We evaluate our model on two datasets (TripAdvisor and Recipe). We have released the source code to facilitate future research (<https://github.com/dreasysnail/INSET>).

Dataset and Preprocessing. We conduct experiments on the TripAdvisor and Recipe datasets. For the TripAdvisor dataset of hotel reviews (Wang et al., 2010), we partially follow the preprocessing in (Cho et al., 2019). Our preprocessing includes, but is not limited to: (i) discarding reviews containing non-English tokens; (ii) removing duplicate reviews so that only one copy is retained. We set the maximum number of tokens in a sentence to be 32 and the minimum number of sentences in a review to be 7 (so that the context is not too short). Any review with longer sentences or having

a smaller number of sentences is discarded.

We use the following strategy to mask sentences. For a paragraph consisting of $M \geq 7$ consecutive sentences, we split it into $M - 6$ data points, each of which has exactly 7 sentences. The j 'th data point spans from the j 'th to the $(j + 6)$ 'th sentence (inclusive) of the paragraph, for $j = 1, 2, \dots, M - 6$. We mask the middle (i.e., 4th) sentence for each data point so that the masking rate is $1/7 \approx 14.3\%$, which is close to that (15%) of BERT. After preprocessing, the size of the dataset (training, validation, test) is (1108134, 62543, 533) data points.

Our strategy of always masking the middle sentence out of 7 sentences is not only the simplest but also without loss of generality. Our model is directly applicable to the situation where we randomly mask, e.g., 3 out of 20 sentences. However, the quality of human evaluation may be affected because the patience and attention of human evaluators may decrease as the context length increases. For the effectiveness of human evaluation, we use the simplest strategy to mask sentences.

The Recipe dataset is obtained from (<https://commoncrawl.org>), where the metadata is formatted according to Schema.org (<https://schema.org/Recipe>). We use the same preprocessing as that of the TripAdvisor dataset except that instructions with less than 5 sentences are discarded. After preprocessing, the size of the dataset (training, validation, test) is (1073886, 56055, 500) data points. Recipe instructions usually describe a time-ordered procedure, and thus are ideal for testing the reasoning capability of the model.

Evaluation Metrics. Following (Galley et al., 2019; Zhang et al., 2020), we perform automatic evaluation using standard machine translation metrics, including BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and METEOR (Lavie and Agarwal, 2007). As a variant of BLEU, NIST weights n -gram matches by their information gain, and thus penalizes uninformative n -grams. We also use Entropy (Zhang et al., 2018) and Dist- n (Li et al., 2016) to evaluate lexical diversity. See (Galley et al., 2019) for more details.

BLEU, NIST, and METEOR measure the similarity between the generated sentence and the ground truth. They are not ideal scores for our task because a sentence that is semantically very different from the ground truth could possibly fit the context perfectly. However, it may still be helpful to compute these commonly used scores. It

is an important and challenging open problem to design an automatic score that faithfully measures the overall quality of the generation in our task.

Baseline. Our baseline is the self-attention model for text infilling (Zhu et al., 2019). It is a transformer language model with novel positional encoding. The traditional approach of encoding the absolute position of each token is not directly applicable to our task because we do not know in advance the absolute positions of contextual tokens after the missing sentence. To resolve this issue, (Zhu et al., 2019) divides the text into segments. In the case of only one masked sentence, the first (third) segment consists of contextual tokens before (after) the mask, and the second corresponds to the mask. Then, each token is indexed by its segment ID and its position ID within the segment. The missing tokens are sequentially generated from these IDs and the current surrounding context.

Training the baseline model on our dataset, we use the same set of hyperparameters as in the original reference except that the batch size is set to 250 (it is 400 in (Zhu et al., 2019)). This avoids out-of-memory errors. Note that we are handling much longer sequences (usually > 100 tokens) than (Zhu et al., 2019), in which the maximum number of tokens in a sequence is only 16.

The baseline model is trained for a sufficient number (30) of epochs until the validation (negative log-likelihood) loss and perplexity clearly saturate. We report the results of the checkpoint with the smallest validation loss and perplexity. Note that we observe that other checkpoints in the saturation regime behave very similarly on the test set.

Keyword Extraction. In the task of sentence infilling with lexical constraints, we need to extract keywords from the masked sentence. Keyword extraction is a classical problem in information retrieval. Standard methods include, but are not limited to, tf-idf (term frequency-inverse document frequency) (Ramos, 2003). We have tried tf-idf, but it does not work well for the TripAdvisor dataset of hotel reviews. One reason is that this dataset has quite a few typos, and unfortunately tf-idf favors them because typos occur less frequently than normal words. This issue can be resolved by manually filtering out all typos. After the fix, however, we observe that the quality of extracted keywords remains unsatisfactory.

We use the following strategy to extract key-

words. We first define a list of stop words. To this end, we use the stop word list from NLTK (Bird et al., 2009) and manually add a number of words (e.g., “hotel”) that are not very informative for the particular dataset of hotel reviews. For each sentence, we select non-stop words that appear most frequently in the entire dataset. We usually select two keywords per sentence, but occasionally select one or even zero if few words remain after filtering out stop words and typos. We observe that the keywords extracted with this strategy can pivot the gist of most sentences well.

Model Size and Hyperparameters. Our architecture has several components. The encoder \mathcal{E} and the sentence-level transformer \mathcal{T} have the same size as BERT_{BASE}. The decoder \mathcal{D} has the same size as GPT-2 (117M). In the presence of lexical constraints, the constraint feature encoder \mathcal{K} has the same size as BERT_{BASE}. During decoding, we use beam search with beam size 5.

4.2 Experimental Results

Sentence Representation Learning. We first qualitatively evaluate the smoothness of the latent-space sentence embeddings learned via denoising autoencoding. Table 1 shows two examples of sentence interpolation on the TripAdvisor dataset. In each example, the first and last sentences are inputs by hand, and the 3 intermediate ones are interpolations generated by our (denoising) autoencoder. We observe that the interpolations not only combine words from input sentences, but are readable, meaningful, and show a smooth semantic transition from the first to the last sentence. We speculate that the power of generating fluent and semantically coherent sentence interpolations is derived from BERT and GPT-2. Inherited from these large-scale pre-trained models, the latent-space sentence embedding is reasonably smooth as our sentence interpolation results show.

Automatic Evaluation. Table 2 shows the BLEU, NIST, METEOR, Entropy, Dist- n scores, and the average length (number of words) of the generated sentences. For the TripAdvisor dataset, we also present results in the presence of keyword constrains.

Table 2 compares the baseline (Zhu et al., 2019), our results, and the ground truth. In the absence of keyword constraints, INSET outperforms the baseline in terms of all scores on both datasets. This indicates that our results are semantically closer

example 1	
A	The pool area was nice and sunbathing was great.
-	The pool area was nice and staff was great.
-	The pool area staff was nice and very helpful.
-	Front desk staff were very helpful and friendly.
B	Front desk staff were very nice and helpful.
example 2	
A	The service was attentive and we had the best food in town.
-	The service was attentive and we had a great room with plenty of food.
-	The room was spacious with good service and we had a queen bed.
-	The room was very spacious with queen beds.
B	The room was very spacious with 2 queen beds.

Table 1: Sentence interpolation. “A” and “B” are two sentences in the test set. The intermediate sentences are generated by interpolating between the latent-space representations of A and B.

to the ground truth and are more diverse than the baseline. In terms of the average generation length, our results are much closer to the ground truth than the baseline is.

Table 2 also presents two ablation studies. The first shows the performance decrease with less context. Recall that each data point in the TripAdvisor dataset has 6 contextual sentences (full context). We train INSET on the same set of data points but truncate the context to 4 sentences (less context). The second ablation study shows the effect of context in the presence of keywords. We compare two models. The first (INSET w/ context) is the model described in Subsection 3.3. Its generation is based on both keywords and context. The second model (INSET w/o context) is $\mathcal{D} \circ \mathcal{K}$, which directly decodes the output of the constraint feature encoder \mathcal{K} using the decoder \mathcal{D} . Its generation is only based on keywords but not context. We observe that the scores of the first model are higher than those of the second. Both ablation studies show that our model can make full use of context to improve the generation.

Human Evaluation. We performed human evaluation of our method on the TripAdvisor dataset. We used a crowd evaluation platform to compare two systems and assess their fluency, informativeness, and relevance to the surrounding context (coherence) on 500 random samples from the test set. Following recommended best practices, each sample was evaluated by five judges. We performed simple spam detection by excluding judges that were too fast or performed too low on a gold set. To avoid bias, we randomized the position of each system while asking judges to compare our systems (with and without keywords) with the ground truth

Dataset	Method	NIST		BLEU		MET-	Ent.	Dist		Len.
		N-2	N-4	B-2	B-4	EOR	E-4	D-1	D-2	
Trip	<i>Without keyword constraints:</i>									
	baseline	0.54	0.54	4.29%	0.54%	5.85%	3.10	1.32%	2.23%	6.97
	INSET (full context)	1.23	1.23	6.08%	0.96%	7.04%	8.13	16.30%	46.64%	10.70
	INSET (less context)	1.02	1.02	4.74%	0.51%	5.83%	7.85	12.98%	41.39%	11.26
	<i>With keyword constraints:</i>									
	INSET (w/ context)	3.09	3.15	20.14%	6.57%	16.48%	8.34	22.61%	63.60%	11.23
INSET (w/o context)	3.00	3.04	19.47%	6.07%	16.00%	8.16	20.51%	57.41%	11.12	
	ground truth (human)	-	-	-	-	-	8.40	33.96%	79.84%	11.36
Recipe	baseline	0.67	0.68	3.91%	0.88%	5.23%	3.12	0.37%	0.47%	15.32
	INSET (ours)	1.36	1.37	7.24%	1.33%	7.07%	7.99	20.12%	55.13%	9.63
	ground truth (human)	-	-	-	-	-	8.22	29.21%	74.97%	10.55

Table 2: Automatic evaluation. “w/ context” indicates that the generation is based on both keywords and context. “w/o context” indicates that the generation is only based on keywords but not context. “Ent.” and “Len.” stand for Entropy and the average generation length, respectively.

system A	system B	critierion	prefer A (%)	same (%)	prefer B (%)
INSET (ours)	baseline	coherence	54.16	13.76	32.07
		fluency	43.38	26.98	29.64
		informativeness	53.48	18.79	27.72
INSET (ours)	ground truth	coherence	27.87	15.69	56.44
		fluency	21.78	31.38	46.84
		informativeness	27.49	21.92	50.59
INSET w/ keywords w/ context	ground truth	coherence	18.50	23.45	58.04
		fluency	17.82	29.78	52.39
		informativeness	20.54	26.13	53.33
INSET w/ keywords w/ context	INSET w/ keywords w/o context	coherence	37.71	37.62	24.68
		fluency	36.16	37.87	25.97
		informativeness	35.93	39.86	24.21
INSET w/ keywords w/ context	INSET w/o keywords w/ context	coherence	34.97	17.06	47.97
		fluency	29.30	28.04	42.65
		informativeness	31.73	23.24	45.03

Table 3: Human evaluation. “w/(w/o) keywords” and “w/(w/o) context” indicate whether the generation is based on keywords and context, respectively. All numbers are percentages.

and the text infilling baseline (Zhu et al., 2019).

Table 3 shows the human evaluation results. The judges strongly prefer our results (without keywords) to the baseline in all aspects: coherence, fluency, and informativeness. They also strongly prefer the ground truth to our results. Moreover, our results with keywords and context are compared with three other systems: (i) the ground truth; (ii) our results with keywords but not context; (iii) our results with context but not keywords. The second comparison shows that in the presence of keywords, our model can use context to improve all aspects of the generation. The third comparison shows that the presence of keywords reduces the performance of our model, probably because keywords are constraints that the model must take care of.

Generated Examples. To qualitatively demonstrate the effectiveness of our model, Table 4 shows some examples from the TripAdvisor and Recipe datasets. We observe that the baseline (Zhu et al., 2019) tends to generate generic sentences, while our results (either with or without keywords) are more informative and can fit the surrounding context reasonably well. Table 5 shows examples generated by our model in the same context but with different keywords. Our model can extend keywords to a full sentence, adapting to the context. More examples generated by our model on both datasets are given in Appendix A.

5 Conclusions and Outlook

We study the task of sentence infilling, which is analogous to the masked language modeling task for (pre-)training BERT, but now it is at the sen-

	example from the TripAdvisor dataset	example from the TripAdvisor dataset	example from the Recipe dataset
preceding context	It was such a pleasure to see something new every night. It was not very crowded so we were able to get great seats at either the pool or the beach. The VIP service was great for dinner reservations and pillow service.	The walls are very thin. Since this is a family vacation type of hotel, people are up at the pool/bbq area/hallways during all hours of the night. Do not stay here if you need a quite night of sleep.	After another 15 minutes or so the mixture should thicken up. The mixture will continue to thicken as it cools.
following context	Enjoyed the shrimp cocktail and seafood salad delivered to us while enjoying the pool. All of us would not want to stay at another resort and are planning to go back again. Enjoy and Hola!Karen and FriendsMilford, CT	You have to take multiple elevators to go all the way to the 5th floor. My other complaint is that the hotel staff seemed a bit unprofessional. Not what I'm used to when I stay at Marriot properties.	Sterilize your jars and lids and while still hot fill with the jam leaving about a 1/2 inch headspace. Place lids onto the jars and boil in a water bath with jars covered by 3 inches of water for 10 minutes.
ground truth	We did bring a lot of \$1 for tipping and of course the service stepped up a notch more.	Also, the elevator situation is weird.	Remove from the heat and stir in your amaretto.
baseline	The staff was friendly and helpful.	The rooms are very clean and well kept.	Add the flour mixture to the dry ingredients and mix well.
INSET	The buffet dinner was amazing and we had the best food in the resort.	There is only one elevator block in the hotel.	Carefully remove the jars from hot water and keep going until a thick sauce is formed.
+ keywords	\$, service	elevator, situation	-
INSET (w/ keywords)	Service fee for the buffet dinner was \$5.00 and we paid \$5.00 extra for food service.	The elevator situation is extremely frustrating.	-

Table 4: Examples generated by our model and the baseline.

preceding context	My room was a very good size. Tiled floors and woodchip painted walls. The tv did not work - so what.
following context	Great places to eat close by and very reasonable. No air con -so summer could be sticky. My concern is the left luggage room not supervised.
human oracle	The location is terrific beside Sevilla metro stn so only 2 to get by metro all the way to airport.
+ (walk, shopping)	Walking distance to shopping mall and Circular Quay.
+ (internet, \$)	Internet cost \$20.00 per day.

Table 5: Examples generated by our model in the same context but with different keywords. “+ (· · ·)” is keywords.

tence level. Sentence infilling requires the model to handle long-range inter-sentential correlation and to process high-level semantic information. It is complementary to (token-level) masked language modeling, which focuses more on syntactic appropriateness and short-range correlation. We propose a framework called INSET to decouple three aspects of the task (understanding, planning, and generation) and address them in a unified manner. We demonstrate the effectiveness of our approach using automatic and human evaluation.

Our approach can be modified or extended in some ways. (i) We use a denoising autoencoder to obtain sentence embeddings. One can try to use a variational autoencoder (Kingma and Welling, 2014) instead. A large-scale pre-trained variational autoencoder (Li et al., 2020) could possibly improve the smoothness of sentence embeddings. (ii) Our model predicts a feature vector for the missing sentence. This vector can be fed into and serve as a guide to token-level models including the baseline (Zhu et al., 2019).

Since sentence infilling is analogous to masked language modeling, we expect that it can also be used as a pre-training task. For example, in ma-

chine translation of long texts, it is often the case that sentences are translated independently from each other. This sometimes leads to incoherence or even inconsistency between the translated sentences. A post-editor to fix the issue (Voita et al., 2019) should be able to understand inter-sentential relationship and to generate fluent sentences in the surrounding context, both of which can be learned from sentence infilling.

Note. After this paper was posted on arXiv, some related works appeared. (Shen et al., 2020) proposes Blank Language Model for text infilling and other tasks. (Donahue et al., 2020) trains (fine-tunes) a language model (GPT-2) for text and sentence infilling. (Li et al., 2020) pre-trains a large-scale variational autoencoder with a pair of BERT and GPT-2. (Ippolito et al., 2020) uses a sentence-level language model, which operates on sentence embeddings obtained from BERT, to predict story endings.

Acknowledgments

We thank Bill Dolan, Chris Quirk, and Jingjing Liu for helpful discussions and suggestions.

References

- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2020. [Abductive commonsense reasoning](#). In *International Conference on Learning Representations*.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.
- Yu Cheng, Zhe Gan, Yizhe Zhang, Oussama Elachqar, Dianqi Li, and Jingjing Liu. 2020. [Contextual text style transfer](#). arXiv:2005.00136.
- Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xijun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. 2019. [Towards coherent and cohesive long-form text generation](#). In *Proceedings of the First Workshop on Narrative Understanding*, pages 1–11, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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.
- George Doddington. 2002. [Automatic evaluation of machine translation quality using n-gram co-occurrence statistics](#). In *Proceedings of the Second International Conference on Human Language Technology Research, HLT '02*, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Chris Donahue, Mina Lee, and Percy Liang. 2020. [Enabling language models to fill in the blanks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. 2019. Grounded response generation task at DSTC7. http://workshop.colips.org/dstc7/papers/DSTC7_Task_2_overview_paper.pdf.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. 2020. [What makes a good story? Designing composite rewards for visual storytelling](#). In *AAAI Conference on Artificial Intelligence*.
- Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. [Unsupervised hierarchical story infilling](#). In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43, Minneapolis, Minnesota. Association for Computational Linguistics.
- Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. 2020. [Toward better storylines with sentence-level language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [SpanBERT: Improving pre-training by representing and predicting spans](#). *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Dongyeop Kang and Eduard Hovy. 2019. [Linguistic versus latent relations for modeling coherent flow in paragraphs](#). 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 5809–5815, Hong Kong, China. Association for Computational Linguistics.
- Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. [CTRL: A conditional transformer language model for controllable generation](#). arXiv:1909.05858.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.
- Diederik P. Kingma and Max Welling. 2014. [Auto-encoding variational bayes](#). In *International Conference on Learning Representations*.
- Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Herve Jegou. 2019. [Large memory layers with product keys](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8548–8559. Curran Associates, Inc.
- Alon Lavie and Abhaya Agarwal. 2007. [METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments](#). In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. arXiv:1910.13461.
- Chunyuan Li, Xiang Gao, Yuan Li, Xiujun Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020. **Optimus: Organizing sentences via pre-trained modeling of a latent space**. arXiv:2004.04092.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. **A diversity-promoting objective function for neural conversation models**. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. **A hierarchical neural autoencoder for paragraphs and documents**. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115, Beijing, China. Association for Computational Linguistics.
- Dayiheng Liu, Jie Fu, Pengfei Liu, and Jiancheng Lv. 2019. **TIGS: An inference algorithm for text infilling with gradient search**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4146–4156, Florence, Italy. Association for Computational Linguistics.
- Sourab Mangrulkar, Suhani Shrivastava, Veena Thenkanidiyoor, and Dileep Aroor Dinesh. 2018. **A context-aware convolutional natural language generation model for dialogue systems**. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 191–200, Melbourne, Australia. Association for Computational Linguistics.
- Tomas Mikolov and Geoffrey Zweig. 2012. **Context dependent recurrent neural network language model**. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **BLEU: a method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. **Language models are unsupervised multitask learners**. https://d4mucfpxsywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. **Exploring the limits of transfer learning with a unified text-to-text transformer**. arXiv:1910.10683.
- Juan Ramos. 2003. Using TF-IDF to determine word relevance in document queries. <https://www.cs.rutgers.edu/~mlittman/courses/ml03/iCML03/papers/ramos.pdf>.
- 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, Hong Kong, China. Association for Computational Linguistics.
- Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. 2019. **Towards generating long and coherent text with multi-level latent variable models**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2079–2089, Florence, Italy. Association for Computational Linguistics.
- Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. **Blank language models**. arXiv:2002.03079.
- Yong-Siang Shih, Wei-Cheng Chang, and Yiming Yang. 2019. **XL-Editor: Post-editing sentences with XLNet**. arXiv:1910.10479.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tiejun Liu. 2019. **MASS: Masked sequence to sequence pre-training for language generation**. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936, Long Beach, California, USA. PMLR.
- Alessandro Sordani, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015a. **A hierarchical recurrent encoder-decoder for generative context-aware query suggestion**. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM '15*, pages 553–562, New York, NY, USA. ACM.
- Alessandro Sordani, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015b. **A neural network approach to context-sensitive generation of conversational responses**. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

- Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. [Context-aware natural language generation with recurrent neural networks](#). arXiv:1611.09900.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. [Extracting and composing robust features with denoising autoencoders](#). In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pages 1096–1103, New York, NY, USA. Association for Computing Machinery.
- Oriol Vinyals and Quoc V. Le. 2015. [A neural conversational model](#). arXiv:1506.05869.
- Elena Voita, Rico Sennrich, and Ivan Titov. 2019. [Context-aware monolingual repair for neural machine translation](#). 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 877–886, Hong Kong, China. Association for Computational Linguistics.
- Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. [Latent aspect rating analysis on review text data: A rating regression approach](#). In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10*, pages 783–792, New York, NY, USA. Association for Computing Machinery.
- Tian Wang and Kyunghyun Cho. 2016. [Larger-context language modelling with recurrent neural network](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1329, Berlin, Germany. Association for Computational Linguistics.
- Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiayi Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2018. [Topic compositional neural language model](#). In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 356–365, Playa Blanca, Lanzarote, Canary Islands. PMLR.
- Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. [Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking](#). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 275–284, Prague, Czech Republic. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [XLNet: Generalized autoregressive pretraining for language understanding](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. [Hierarchical attention networks for document classification](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. [Generating informative and diverse conversational responses via adversarial information maximization](#). In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1810–1820. Curran Associates, Inc.
- Yizhe Zhang, Xiang Gao, Sungjin Lee, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. [Consistent dialogue generation with self-supervised feature learning](#). arXiv:1903.05759.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DialogPT: Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.
- Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. [Text infilling](#). arXiv:1901.00158.

A Additional Generated Examples

Tables 6, 7 show some additional examples generated by our model (without keywords) on the TripAdvisor and Recipe datasets, respectively. The results are semantically informative and can fit the surrounding context reasonably well. Table 8 provides additional examples to Table 5. Our model can incorporate keywords into the generated sentence in a smart way, adapting to the context.

	example 1	example 2
preceding context	I went in October to meet with their FABULOUS wedding coordinator Summer Laetari. Their property is very beautiful, it's extremely green and lush. Parrot Key has 4 pools.	Good Location if traveling for business or you have a car! Got this hotel thru a discount travel company and paid \$65.00 american a night. Excellent deal at this price.
following context	Their cottages are brand new, very clean and well appointed. If you are looking for a place to have a destination wedding I would recommend Parrot Key! My family and I have already planned another trip to visit next month.	Unfortunately the view is going to be partly blocked with yet another "Glass tower" going in. The room was spacious and clean. No tub in our room.
ground truth	It's very colorful and unique.	We had a terrific view from the 16th floor.
INSET	There is also a beach resort with lots of loungers.	We had a room on the upper floor which overlooks the lobby.
	example 3	example 4
preceding context	My family stayed here for 5 nights in August 2011. The resort is beautiful and the grounds are immaculately manicured. The kitchen is great for the family.	We stayed in 2 interconnecting rooms as we are a family of 5. We started off with a bad start, as the check in was not aware that we were with 3 kids. I booked directly with them and got a confirmation via email for 2 rooms for 2 adults.
following context	We would just pack a cooler and head out in our rental car and explore the island. The pools at the resort were fabulous and the staff was attentive. We used the grills(kept very clean) several nights.	Obviously this was not reflected in the paper work check-in had. We could only add an extra bed for an extra charge, but I refused to pay for this as I had phoned them before. The check-in lady would not bend, and we had to go for 2 rooms with 2 separate beds.
ground truth	We were able to keep essentials in the room which made those early morning excursions more enjoyable.	Before we arrived I called reservations to change this into 2 adults and 3 children.
INSET	We have plenty of kitchen utensils and the beach was a nice place to stay.	When we checked in we were told that we had to request another room on the 2nd floor due to the extra charges.
	example 5	example 6
preceding context	It was such a pleasure to see something new every night. It was not very crowded so we were able to get great seats at either the pool or the beach. The VIP service was great for dinner reservations and pillow service.	My intentions were to expect the worst which made my stay there that much better than everyone elses. If everyone thought they were staying at the Hyatt, no wonder they thought so negatively about the place. I am in my late twenties and wanted a place where I could walk to local bars, restaurants, etc.
following context	Enjoyed the shrimp cocktail and seafood salad delivered to us while enjoying the pool. All of us would not want to stay at another resort and are planning to go back again. Enjoy and Hola!Karen and FriendsMilford, CT	This was the perfect place for me. As far as the accomodations, the beds were small (but so was everywhere else in Europe) and the showers were unusual. Otherwise it was worth the money for a prime time location in the heart of the night life area.
ground truth	We did bring a lot of \$1 for tipping and of course the service stepped up a notch more.	without struggling to find my way home at night.
INSET	The buffet dinner was amazing and we had the best food in the resort.	So I had no reason to stay in the HOTEL itself.

Table 6: Generated examples by our model on the TripAdvisor dataset

	example 1	example 2
preceding context	Roll up rectangles width-wise and pinch ends to seal. Bake for 12 minutes or until the tops begin to brown.	Drizzle each potato cup with 1 teaspoon browned butter. Cover muffin tin tightly with aluminium foil and place in oven.
following context	Best when served warm. For added flavor, serve with strawberry jelly.	Remove from oven and turn broiler on high. Sprinkle potato rounds evenly with remaining parmesan cheese.
ground truth	Let cool on baking sheet.	Bake for 25 minutes.
INSET	Cool on wire rack and remove.	Bake for 20 minutes or until potatoes are tender.
	example 3	example 4
preceding context	Preheat oven to 425 degrees Fahrenheit. Line a baking sheet with a SILPAT mat.	Heat the oil in a pan at medium. Add the mushrooms and saute until tender, about 7-10 minutes.
following context	With a pastry cutter, cut in the coconut oil and the butter. Make a well and add in the milk 1/2 cup at a time, stirring gently with a wooden spoon.	Add the reserved water and simmer at medium-high until reduced by half, about 10 minutes. Meanwhile cook the pasta as directed on the package.
ground truth	In a bowl, mix the flour, baking powder, baking soda and sea salt.	Add shallots, garlic, thyme, salt and pepper and saute for 2 minutes.
INSET	In a medium bowl, mix together the flour, baking powder, sugar, salt and cinnamon.	Add the garlic and sautee until fragrant, about 2 minutes.
	example 5	example 6
preceding context	After another 15 minutes or so the mixture should thicken up. The mixture will continue to thicken as it cools.	Bake the graham cracker crust for 10 minutes. Remove from oven and allow to cool to room temperature.
following context	Sterilize your jars and lids and while still hot fill with the jam leaving about a 1/2 inch headspace. Place lids onto the jars and boil in a water bath with jars covered by 3 inches of water for 10 minutes.	Stir in the lime zest and lime juice. Stir until mixture is smooth and begins to slightly thicken.
ground truth	Remove from the heat and stir in your amaretto.	Meanwhile, combine the egg yolks and condensed milk in a medium bowl.
INSET	Carefully remove the jars from hot water and keep going until a thick sauce is formed.	In a medium bowl, combine the cream cheese and powdered sugar, stirring until smooth.

Table 7: Generated examples by our model on the Recipe dataset

preceding context	Also has a safe. The hotel is in a good location, beside the City Centre and there are a nice selection of shops within the Monte Carlo. Service was very good but avoid the concierge in the morning when people are booking tours, the queues are long.
following context	No wi-fi in the room which is a bit annoying but they have it in the foodcourt by Starbucks and McDs. Also we were disappointed to see the \$15/night resort fee was charged to our credit card after our stay. I don't recall them mentioning this at check-in.
human oracle	CVs is next door and it's 24/7 so you can buy snacks and anything else you fancy.
+ (breakfast, cereal)	Breakfast is included with cereal, muffins and breads.
+ (food, expensive)	Prices are expensive but food in the hotel is very cheap.

Table 8: Examples generated by our model in the same context but with different keywords. “+ (· · ·)” is keywords.