Sentence Infilling

Wei Wang

Overview

- Text Infilling (ArXiv 2019)
- T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)
- INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

Text Infilling

Wanrong Zhu[1], **Zhiting Hu**[2,3], **Eric P. Xing**[2,3]
Peking University[1], Carnegie Mellon University[2], Petuum Inc.[3]

Introduction

Text Infilling (ArXiv 2019)

- Text infilling, which fills missing text snippets of a sentence or paragraph, is also a common application in real life useful in numerous contexts, such as restoration of historical or damaged documents, contract or article writing with templates, text editing, and so forth.

- Previous studies are not directly applicable to many real scenarios where multiple portions at random positions of the text can be missing.

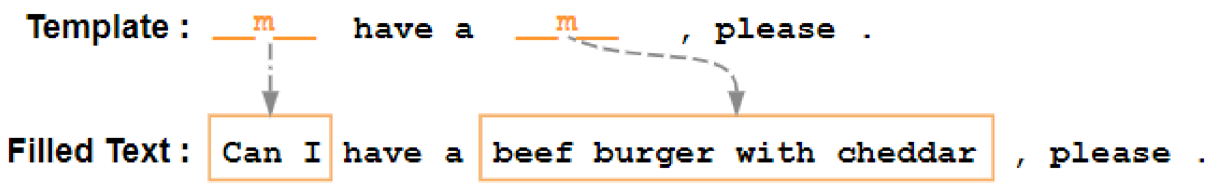

Template : __m__ have a __m__ , please .
Filled Text : Can I have a beef burger with cheddar , please .

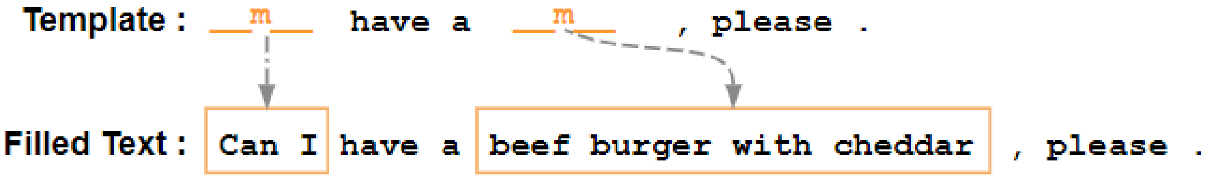
Figure 1: An example of text infilling.

Method

Text Infilling (ArXiv 2019)

- Given a text template where portions of a body of text are deleted or redacted, we want to fill in the blanks properly to produce complete, semantically coherent and meaningful text.

- We study the problem in a supervised setting. That is, we assume a set of pairs including both a template and example filled text for training.



Template : __m__ have a __m__ , please .

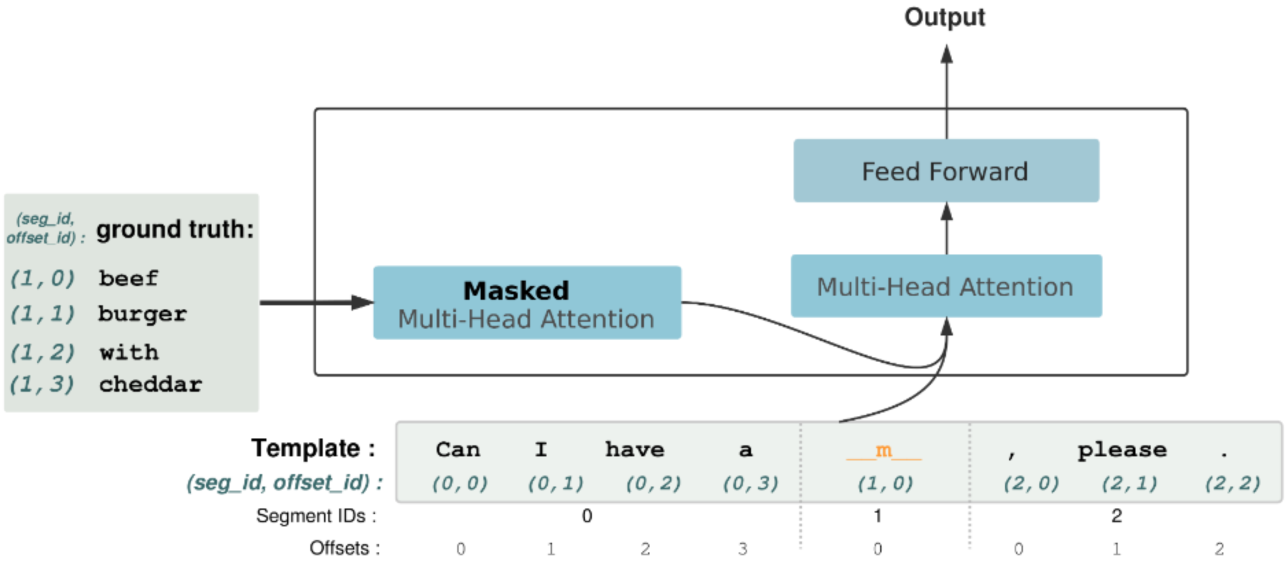
Filled Text : Can I have a beef burger with cheddar , please .

Figure 1: An example of text infilling.

Method

Text Infilling (ArXiv 2019)

- __m__ be a placeholder for a blank.
- <bob> and <eob> are the beginning token and ending token of each blank.
- <bos> and <eos> mark the first and last token for the whole sentence.
- The decoder will fill in the blanks one by one.
- For the infilling of each segment, the decoder fills in the missing token auto-regressively, conditioning on the template together with what has been filled in the template.



(seg_id, offset_id) :	ground truth:
(1,0)	beef
(1,1)	burger
(1,2)	with
(1,3)	cheddar

Template :	Can	I	have	a	__m__	,	please	.
(seg_id, offset_id) :	(0,0)	(0,1)	(0,2)	(0,3)	(1,0)	(2,0)	(2,1)	(2,2)
Segment IDs :	0				1	2		
Offsets :	0	1	2	3	0	0	1	2

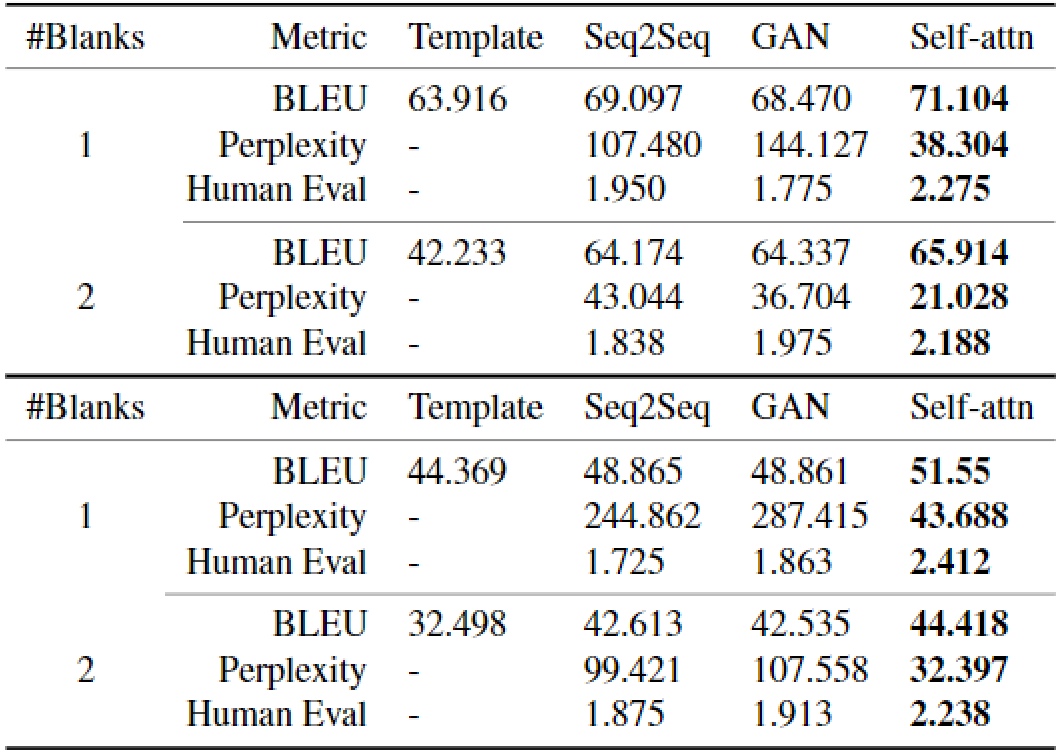
pos = seg_id * base + offset_id

base: a self-defined integer

Experiment

Text Infilling (ArXiv 2019)

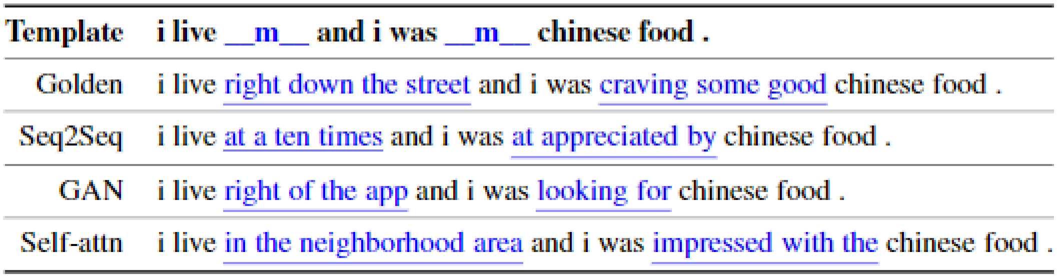
- All methods use the positional embedding as inputs.
- A higher mask rate and a larger number of blanks lead to a more difficult task.
- with increasing mask rate and #blanks, the model performance (BLEU and PPL) drops.
- Seq2seq and GAN provide comparable performance, while the self-attention model consistently outperforms both.



#Blanks	Metric	Template	Seq2Seq	GAN	Self-attn
1	BLEU	63.916	69.097	68.470	**71.104**
	Perplexity	-	107.480	144.127	**38.304**
	Human Eval	-	1.950	1.775	**2.275**
2	BLEU	42.233	64.174	64.337	**65.914**
	Perplexity	-	43.044	36.704	**21.028**
	Human Eval	-	1.838	1.975	**2.188**

#Blanks	Metric	Template	Seq2Seq	GAN	Self-attn
1	BLEU	44.369	48.865	48.861	**51.55**
	Perplexity	-	244.862	287.415	**43.688**
	Human Eval	-	1.725	1.863	**2.412**
2	BLEU	32.498	42.613	42.535	**44.418**
	Perplexity	-	99.421	107.558	**32.397**
	Human Eval	-	1.875	1.913	**2.238**

Table 1: Results of varying mask rates and number of blanks. The upper part of the table is the results of mask_rate=30%, while the lower part is the results of mask_rate=50%.



Template	**i live __m__ and i was __m__ chinese food .**
Golden	i live right down the street and i was craving some good chinese food .
Seq2Seq	i live at a ten times and i was at appreciated by chinese food .
GAN	i live right of the app and i was looking for chinese food .
Self-attn	i live in the neighborhood area and i was impressed with the chinese food .

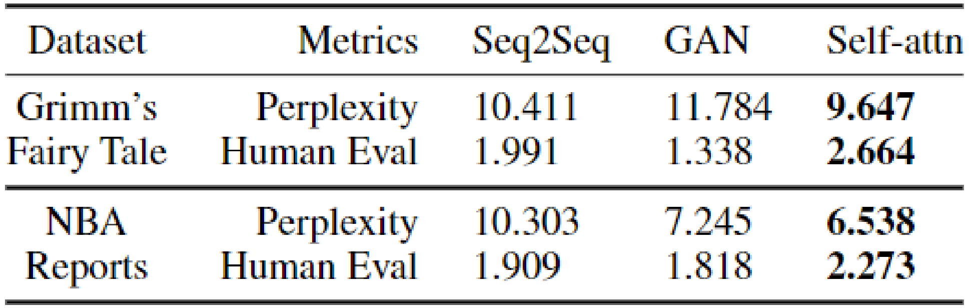
Table 2: Example model outputs on a Yelp test case,

Experiment

Text Infilling (ArXiv 2019)

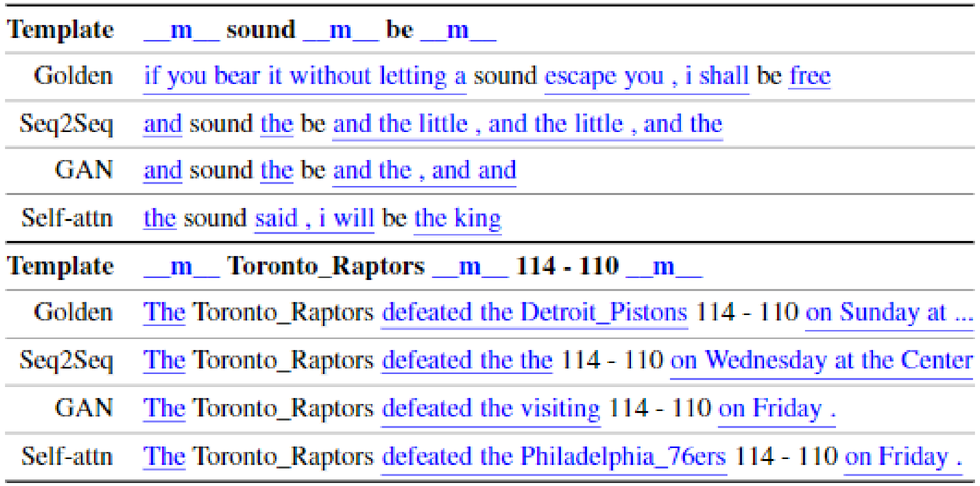
Long Content Infilling

- Grimm's Fairy Tale, leaving only one noun and one verb in the template. The resulting average mask rate is 81.3%.
- NBA news, leave in each template the name of a player or a team, and the numbers. The resulting average mask rate is 78.1%
- With the increasing mask rate, the infilling task becomes more open-end, making BLEU score less suitable.
- Self-attention model again improves over other comparison methods on both datasets.
- Seq2seq and GAN-based model fail to generate semantically coherent and fluent patches to fill the templates. In contrast, the self-attention model tends to produce more reasonable and meaningful results.



Dataset	Metrics	Seq2Seq	GAN	Self-attn
Grimm's Fairy Tale	Perplexity Human Eval	10.411 1.991	11.784 1.338	**9.647** **2.664**
NBA Reports	Perplexity Human Eval	10.303 1.909	7.245 1.818	**6.538** **2.273**

Table 3: Automatic and human evaluation results for long content infilling.



Template	**__m__ sound __m__ be __m__**
Golden	if you bear it without letting a sound escape you , i shall be free
Seq2Seq	and sound the be and the little , and the little , and the
GAN	and sound the be and the , and and
Self-attn	the sound said , i will be the king
Template	**__m__ Toronto_Raptors __m__ 114 - 110 __m__**
Golden	The Toronto_Raptors defeated the Detroit_Pistons 114 - 110 on Sunday at ...
Seq2Seq	The Toronto_Raptors defeated the the 114 - 110 on Wednesday at the Center
GAN	The Toronto_Raptors defeated the visiting 114 - 110 on Friday .
Self-attn	The Toronto_Raptors defeated the Philadelphia_76ers 114 - 110 on Friday .

Table 4: Example model outputs on Grimm's Fairy Tale (upper) and NBA Reports (lower).

Summary

Text Infilling (ArXiv 2019)

- The author proposed **a new task of text infilling**, which aims to fill missing portions of a given sentence/paragraph.
- They **studied several models** for the task, including a self-attention model with global context modeling and segment-aware position embedding.
- On a variety of supervised datasets, the **self-attention model** improved over the seq2seq and GAN-based models.

- However, the method cannot utilize off-the-shelf LMs to infill.

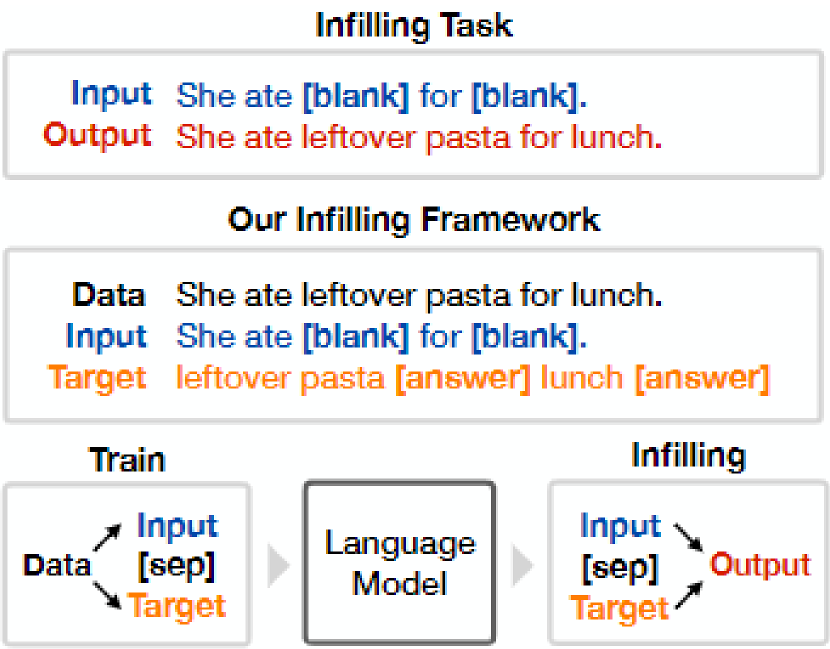
segment-aware position embedding

global context modeling

Enabling Language Models to Fill in the Blanks

ACL2020

- ILM enables off-the-shelf LMs to infill effectively.
- The method can infill multiple variable-length spans with different granularities (e.g. words, n-grams, and sentences).



Infilling Task

Input She ate [blank] for [blank].

Output She ate leftover pasta for lunch.

Our Infilling Framework

Data She ate leftover pasta for lunch.

Input She ate [blank] for [blank].

Target leftover pasta [answer] lunch [answer]

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion

Tianming Wang and **Xiaojun Wan**

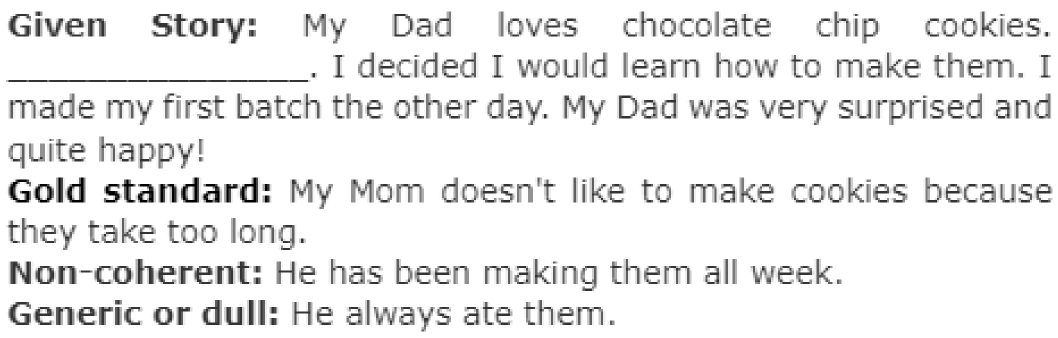
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University

{wangtm, wanxiaojun}@pku.edu.cn

Introduction

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)

- Story completion is a task of generating the missing plot for an incomplete story.
- This task requires machine to first understand what happens in the given story and then infer and write what would happen in the missing part.

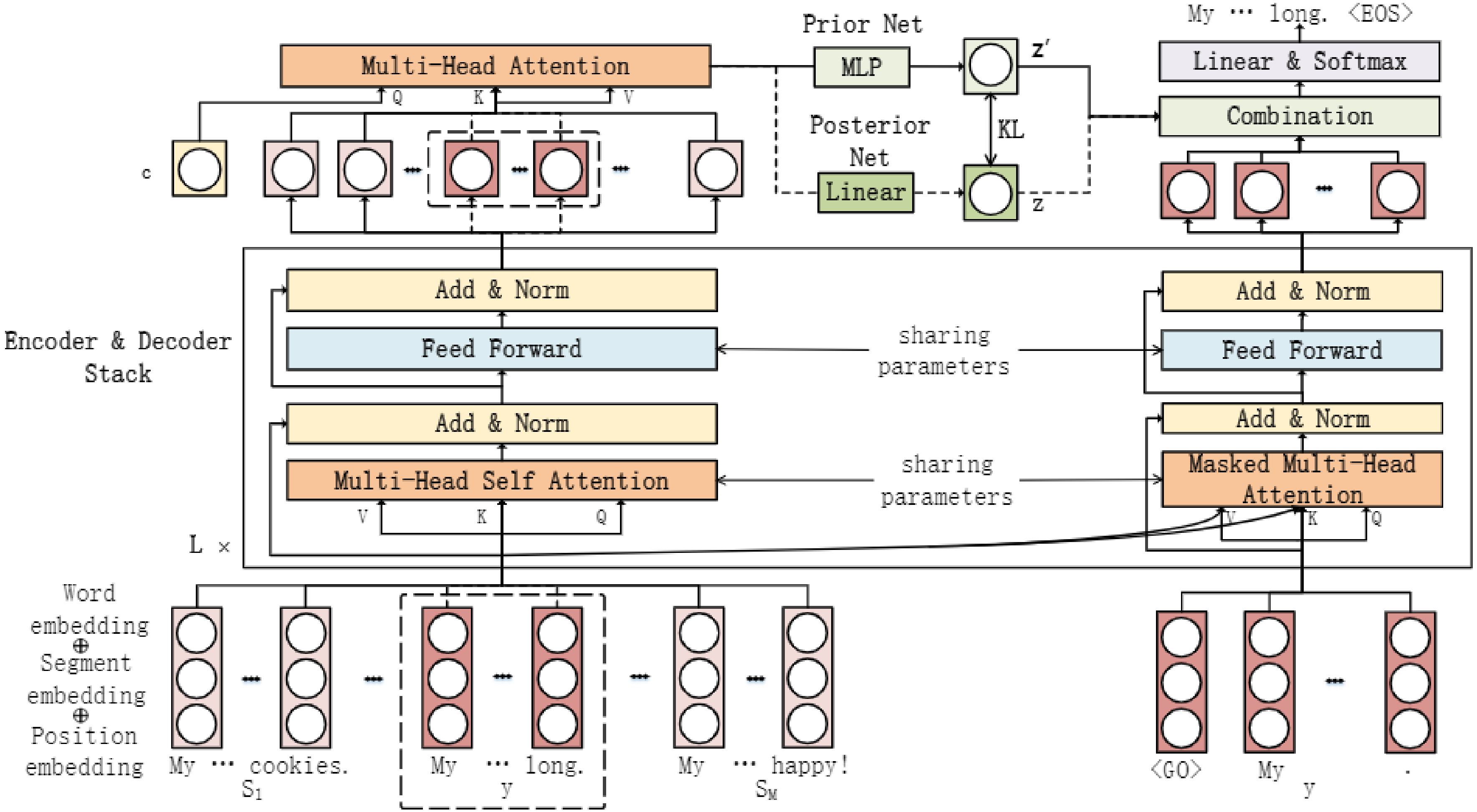


Given Story: My Dad loves chocolate chip cookies. ________________. I decided I would learn how to make them. I made my first batch the other day. My Dad was very surprised and quite happy!
Gold standard: My Mom doesn't like to make cookies because they take too long.
Non-coherent: He has been making them all week.
Generic or dull: He always ate them.

Figure 1: An example incomplete story with different generated plots.

Method

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)



Method

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)

- Input Representation

$$IR_{w_j^i} = WE_{w_j^i} \oplus SE_i \oplus PE_j$$

- Shared Attention Layers

$$\begin{aligned} E_{in}^l &= E_{out}^{l-1} \\ A &= \mathrm{MultiHead}(E_{in}^l, E_{in}^l, E_{in}^l) \\ B &= \mathrm{LayerNorm}(A + E_{in}^l) \\ E_{out}^l &= \mathrm{LayerNorm}(FFN(B) + B) \end{aligned}$$

$$\begin{aligned} D_{in}^l &= D_{out}^{l-1} \\ A &= \mathrm{MultiHead}(D_{in}^l, [E_{in}^l; D_{in}^l], [E_{in}^l; D_{in}^l]) \\ B &= \mathrm{LayerNorm}(A + D_{in}^l) \\ D_{out}^l &= \mathrm{LayerNorm}(FFN(B) + B) \end{aligned}$$

- T-CVAE

$$\begin{aligned} \log p(y|x) &= \log \int_z p(y|x,z)p(z|x)dz \\ &\geq \mathbb{E}_{q(z|x,y)}[\log p(y|x,z)] \\ &- D_{KL}(q_(z|x,y)||p(z|x)) \end{aligned}$$

$$h = \mathrm{MultiHead}(c, E_{out}^L(x;y), E_{out}^L(x;y))$$

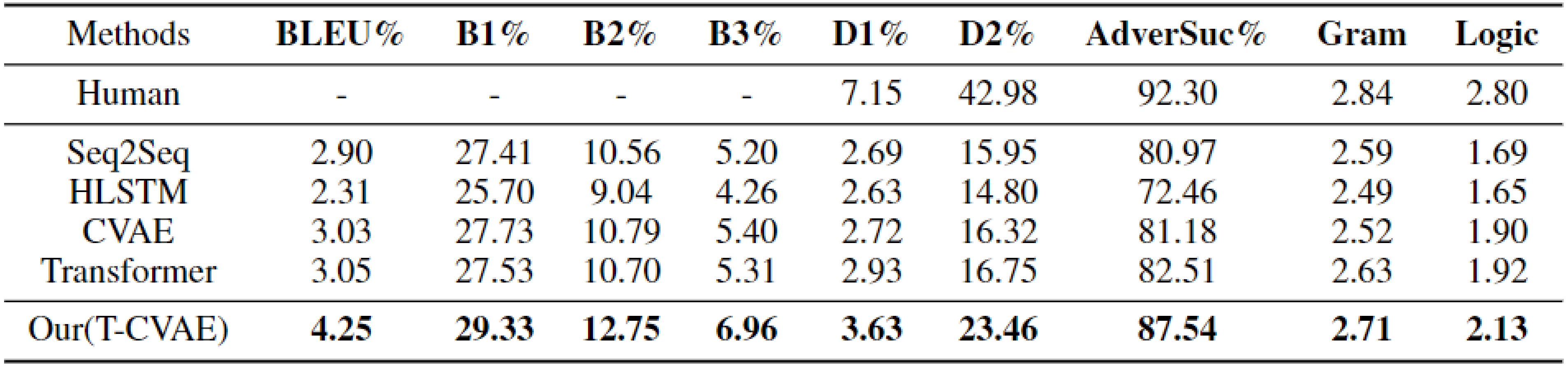
$$\begin{bmatrix} \mu \\ \log(\sigma^2) \end{bmatrix} = hW_q + b_q$$

$$h^{'} = \mathrm{MultiHead}(c, E_{out}^L(x), E_{out}^L(x))$$

$$\begin{bmatrix} \mu^{'} \\ \log(\sigma^{'2}) \end{bmatrix} = \mathrm{MLP}_p(h^{'})$$

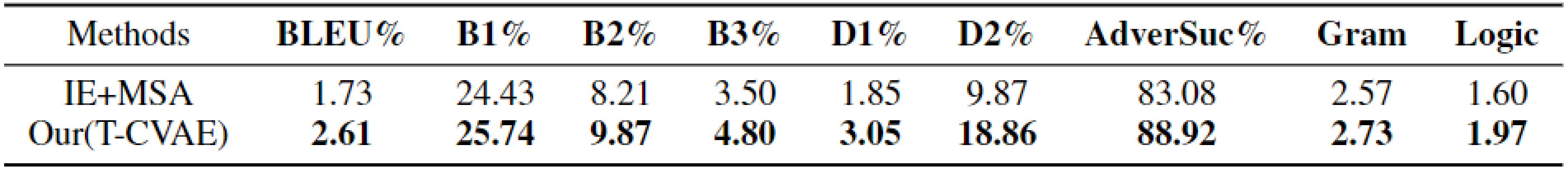
Experiment

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)



Methods	BLEU%	B1%	B2%	B3%	D1%	D2%	AdverSuc%	Gram	Logic
Human	-	-	-	-	7.15	42.98	92.30	2.84	2.80
Seq2Seq	2.90	27.41	10.56	5.20	2.69	15.95	80.97	2.59	1.69
HLSTM	2.31	25.70	9.04	4.26	2.63	14.80	72.46	2.49	1.65
CVAE	3.03	27.73	10.79	5.40	2.72	16.32	81.18	2.52	1.90
Transformer	3.05	27.53	10.70	5.31	2.93	16.75	82.51	2.63	1.92
Our(T-CVAE)	**4.25**	**29.33**	**12.75**	**6.96**	**3.63**	**23.46**	**87.54**	**2.71**	**2.13**

Table 1: Comparison results on the story completion task

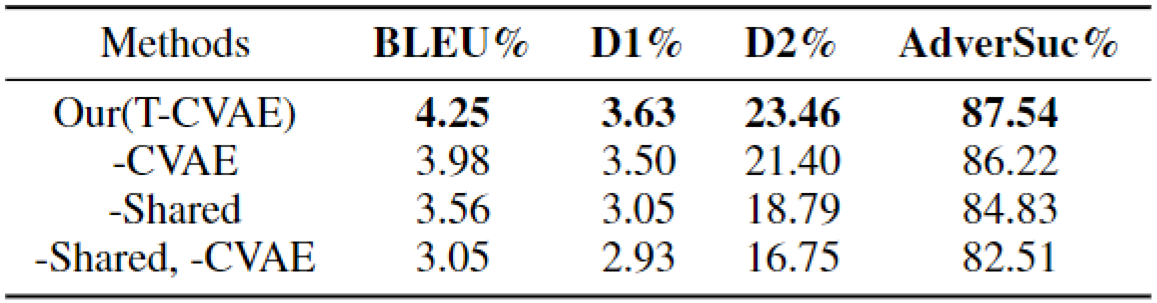


Methods	BLEU%	B1%	B2%	B3%	D1%	D2%	AdverSuc%	Gram	Logic
IE+MSA	1.73	24.43	8.21	3.50	1.85	9.87	83.08	2.57	1.60
Our(T-CVAE)	**2.61**	**25.74**	**9.87**	**4.80**	**3.05**	**18.86**	**88.92**	**2.73**	**1.97**

Table 3: Comparison results on the story ending generation task

Experiment

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)



Methods	BLEU%	D1%	D2%	AdverSuc%
Our(T-CVAE)	**4.25**	**3.63**	**23.46**	**87.54**
-CVAE	3.98	3.50	21.40	86.22
-Shared	3.56	3.05	18.79	84.83
-Shared, -CVAE	3.05	2.93	16.75	82.51

Table 2: Ablation study on story completion. -CVAE means only using Transformer and -Shared means using seperated attention layers.

- Removing both shared attention layer and latent variable, the model degrades to the standard Transformer and achieves the lowest score.

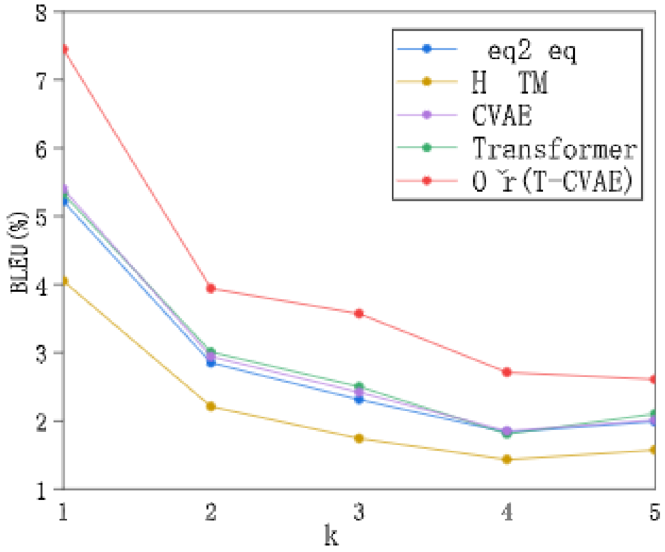
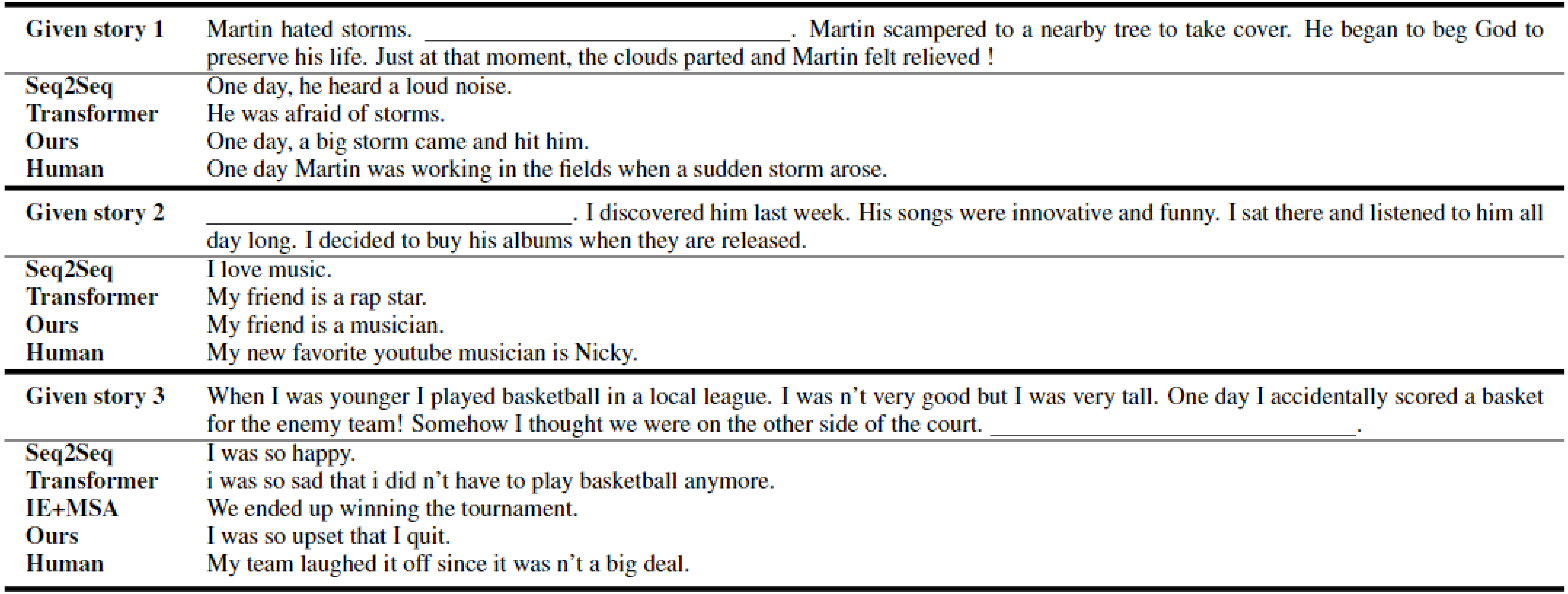


Figure 3: BLEU scores of different models on generating *k*-th sentence

- BLEU score goes down as k increases and it drops significantly from k = 1 to k = 2.
- starting plot is simple and generic, paves the way for the follow-ups; subsequent plots become more specific and informative, which are hard to predict.

Experiment

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)



Given story 1	Martin hated storms. ______________________________. Martin scampered to a nearby tree to take cover. He began to beg God to preserve his life. Just at that moment, the clouds parted and Martin felt relieved !
Seq2Seq	One day, he heard a loud noise.
Transformer	He was afraid of storms.
Ours	One day, a big storm came and hit him.
Human	One day Martin was working in the fields when a sudden storm arose.
Given story 2	______________________________. I discovered him last week. His songs were innovative and funny. I sat there and listened to him all day long. I decided to buy his albums when they are released.
Seq2Seq	I love music.
Transformer	My friend is a rap star.
Ours	My friend is a musician.
Human	My new favorite youtube musician is Nicky.
Given story 3	When I was younger I played basketball in a local league. I was n't very good but I was very tall. One day I accidentally scored a basket for the enemy team! Somehow I thought we were on the other side of the court. ______________________________.
Seq2Seq	I was so happy.
Transformer	i was so sad that i did n't have to play basketball anymore.
IE+MSA	We ended up winning the tournament.
Ours	I was so upset that I quit.
Human	My team laughed it off since it was n't a big deal.

Table 4: Case Study.

Summary

T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion (IJCAI 19)

- This is the first attempt to address the **story completion task** of generating missing plots in any position
- The author proposed a novel Transformer-based conditional variational autoencoder(**T-CVAE**) for this task.

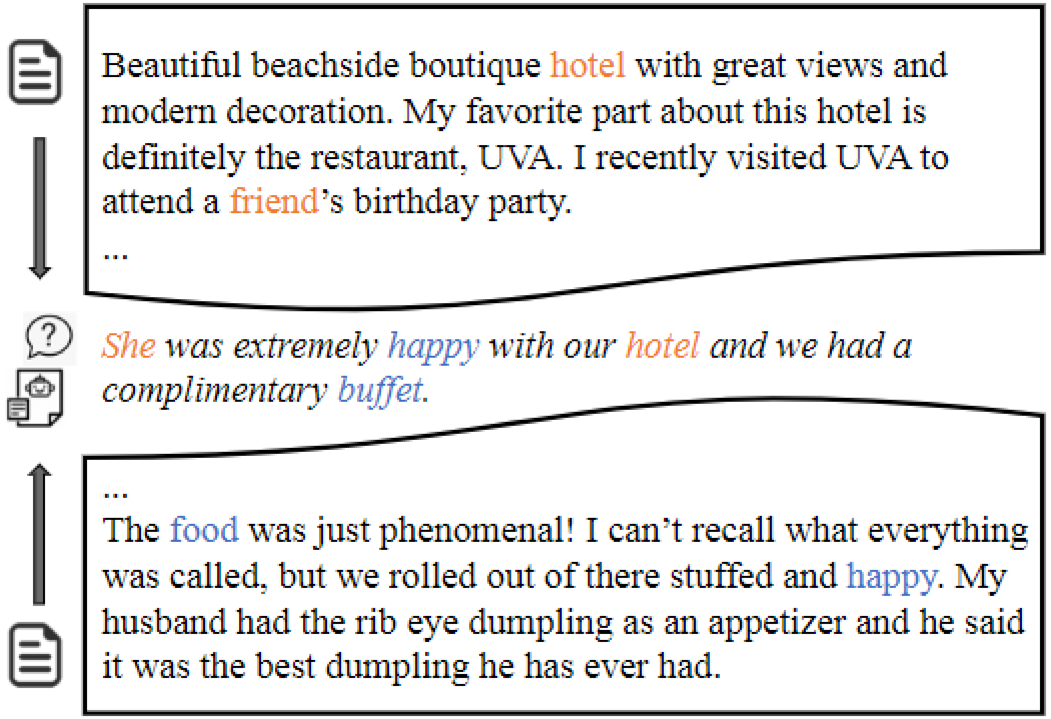
INSET: Sentence Infilling with INter-SEntential Transformer

Yichen Huang (黄溢辰)[1][2]*, Yizhe Zhang[1]*, Oussama Elachqar[1], Yu Cheng[1]
[1]Microsoft Corporation, Redmond, Washington 98052, USA
[2]Center for Theoretical Physics, MIT, Cambridge, Massachusetts 02139, USA
yichuang@mit.edu, {yizzhang, ouelachq, yu.cheng}@microsoft.com

Introduction

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

- Intermediate sentences 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.
- 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 facilitate many text generation applications.



Beautiful beachside boutique hotel with great views and modern decoration. My favorite part about this hotel is definitely the restaurant, UVA. I recently visited UVA to attend a friend's birthday party.
...

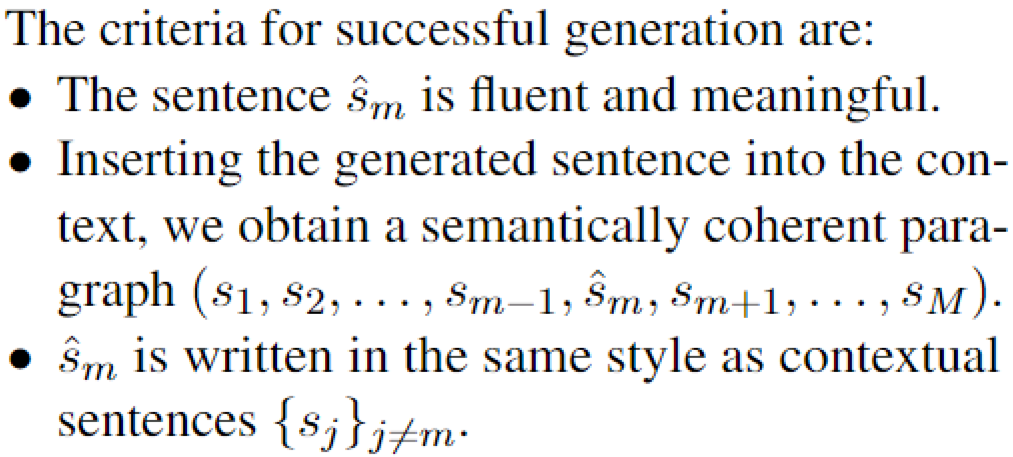
She was extremely happy with our hotel and we had a complimentary buffet.

...
The food was just phenomenal! I can't recall what everything was called, but we rolled out of there stuffed and happy. My husband had the rib eye dumpling as an appetizer and he said it was the best dumpling he has ever had.

Method

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

- The task is to generate a sentence in the missing position such that it fits the context.



The criteria for successful generation are:

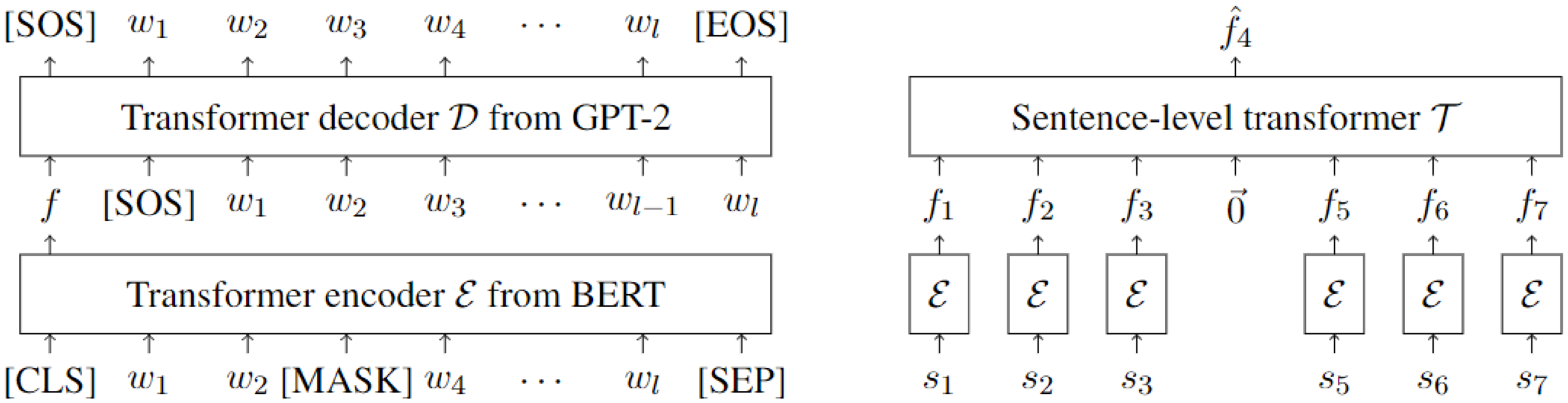
- The sentence $\hat{s}_m$ is fluent and meaningful.
- Inserting the generated sentence into the context, we obtain a semantically coherent paragraph $(s_1, s_2, \ldots, s_{m-1}, \hat{s}_m, s_{m+1}, \ldots, s_M)$.
- $\hat{s}_m$ is written in the same style as contextual sentences $\{s_j\}_{j \neq m}$.

- Since there could be multiple semantically different sentences that fit the same context well, it is not necessary for sm to be close to the ground truth. Rather, sm is considered successful as long as it satisfies the criteria above.

Method

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

- The 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.


[SOS] w_1 w_2 w_3 w_4 $\cdots$ w_l [EOS]
Transformer decoder $\mathcal{D}$ from GPT-2
f [SOS] w_1 w_2 w_3 $\cdots$ w_{l-1} w_l
Transformer encoder $\mathcal{E}$ from BERT
[CLS] w_1 w_2 [MASK] w_4 $\cdots$ w_l [SEP]
$\hat{f}_4$
Sentence-level transformer $\mathcal{T}$
f_1 f_2 f_3 $\vec{0}$ f_5 f_6 f_7
$\mathcal{E}$ $\mathcal{E}$ $\mathcal{E}$ $\mathcal{E}$ $\mathcal{E}$ $\mathcal{E}$
s_1 s_2 s_3 s_5 s_6 s_7

Method

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

- Sentence Representation Learning via Denoising Autoencoding.

To train the autoencoder, we use teacher forcing and minimize the negative log-likelihood loss by (fine-)tuning the parameters of E and D jointly.

- Sentence Feature Prediction.

A sentence-level transformer is used to predict the feature vector of the missing sentence from those of contextual sentences.

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

- Generating Sentences from Features.

At test time, we use the decoder D to generate the missing sentence by mapping the predicted feature vector to the text domain.

Method

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

Sentence Infilling with Lexical Constraints

- The constraint feature encoder.

It is a transformer encoder K that maps a set S of keywords to a feature vector.

- We train K with knowledge distillation.

The teacher model is the sentence encoder E.

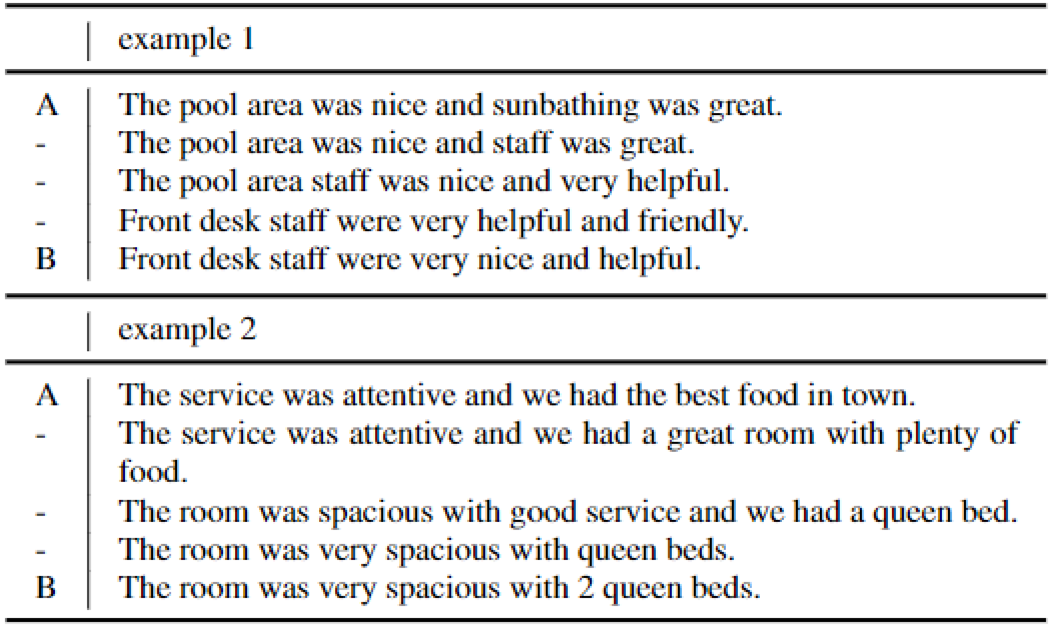
We use the cosine similarity loss between these two feature vectors to teach the student model K.

Experiment

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

Sentence Representation Learning.

- 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.

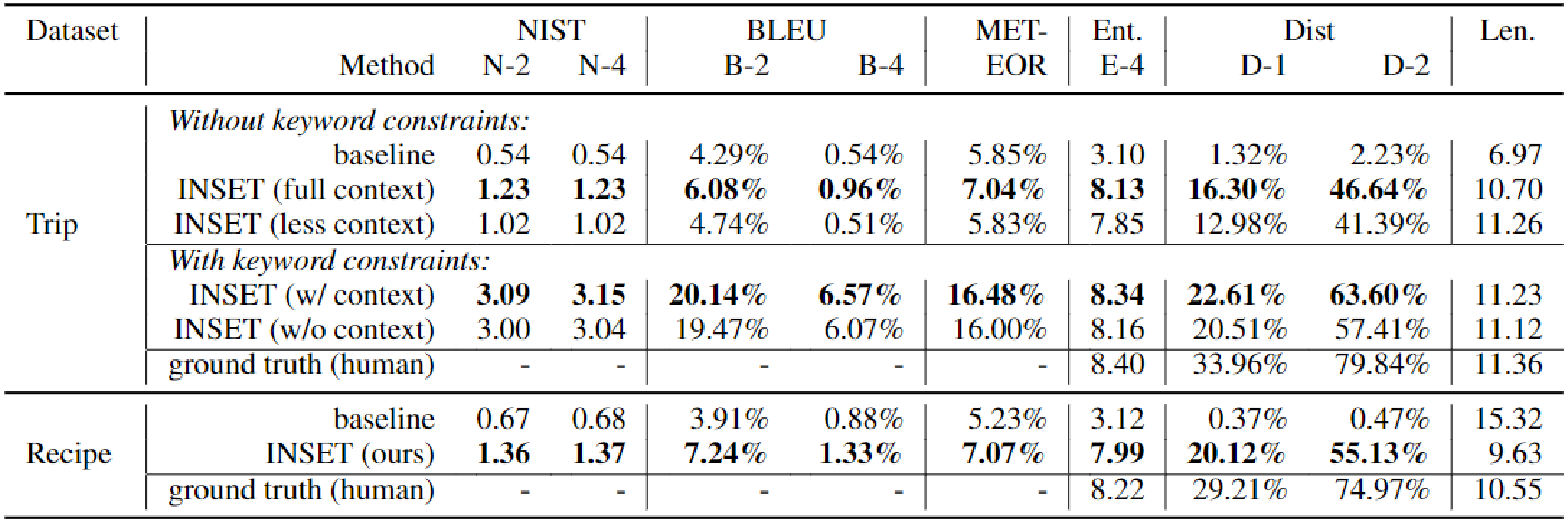


	example 1
A	The pool area was nice and sunbathing was great.
-	The pool area was nice and staff was great.
-	The pool area staff was nice and very helpful.
-	Front desk staff were very helpful and friendly.
B	Front desk staff were very nice and helpful.
	example 2
A	The service was attentive and we had 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.

Experiment

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)



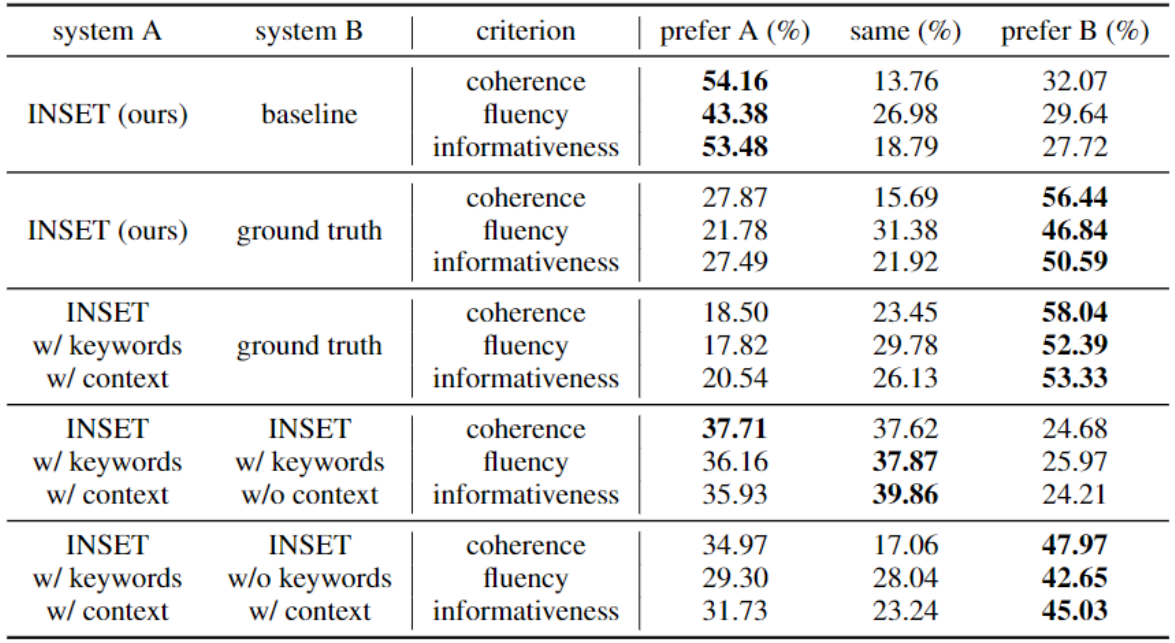
Dataset		NIST		BLEU		MET-	Ent.	Dist		Len.
	Method	N-2	N-4	B-2	B-4	EOR	E-4	D-1	D-2	
Trip	*Without keyword constraints:*									
	baseline	0.54	0.54	4.29%	0.54%	5.85%	3.10	1.32%	2.23%	6.97
	INSET (full context)	**1.23**	**1.23**	**6.08%**	**0.96%**	**7.04%**	**8.13**	**16.30%**	**46.64%**	10.70
	INSET (less context)	1.02	1.02	4.74%	0.51%	5.83%	7.85	12.98%	41.39%	11.26
	With keyword constraints:									
	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.

- In the absence of keyword constraints, INSET outperforms the baseline in terms of all scores. This indicates that our results are semantically closer to the ground truth and are more diverse than the baseline.
- Table 2 also presents two ablation studies. Both ablation studies show that our model can make full use of context to improve the generation.

Experiment

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)



system A	system B	criterion	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**

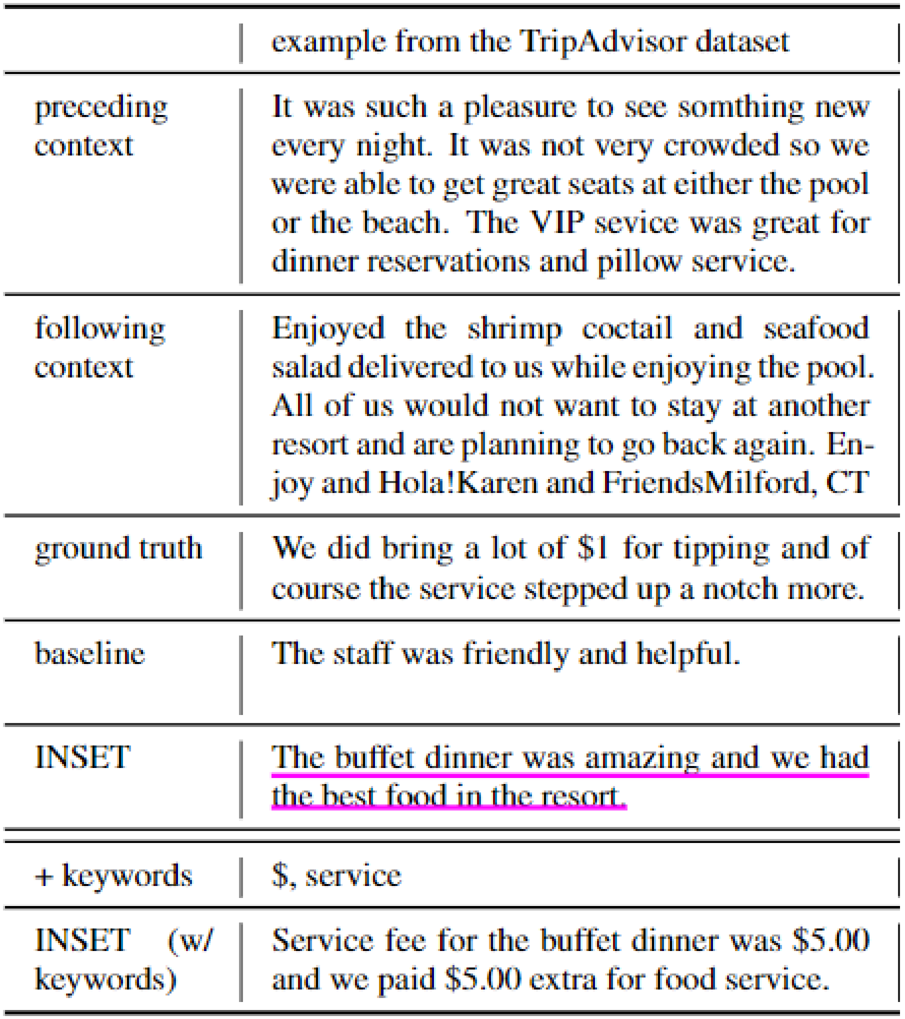
Table 3: Human evaluation. "w/(w/o) keywords" and "w/(w/o) context" indicate whether the generation is based on keywords and context, respectively. All numbers are percentages.

- The judges strongly prefer our results (without keywords) to the baseline in all aspects
- In the presence of keywords, our model can use context to improve all aspects of the generation.
- The presence of keywords reduces the performance of our model

Experiment

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

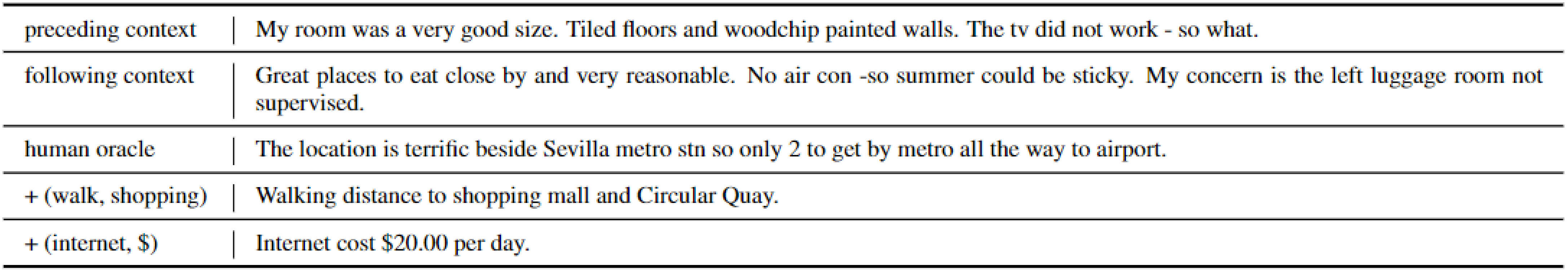
- Table 4 shows some examples from the TripAdvisor and Recipe datasets.
- The baseline tends to generate generic sentences, while our results (either with or without keywords) are more informative and can fit the surrounding context reasonably well.



	example from the TripAdvisor dataset
preceding context	It was such a pleasure to see somthing 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 sevice was great for dinner reservations and pillow service.
following context	Enjoyed the shrimp coctail 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
ground truth	We did bring a lot of $1 for tipping and of course the service stepped up a notch more.
baseline	The staff was friendly and helpful.
INSET	The buffet dinner was amazing and we had the best food in the resort.
+ keywords	$, service
INSET (w/ keywords)	Service fee for the buffet dinner was $5.00 and we paid $5.00 extra for food service.

Experiment

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)



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.

Summary

INSET: Sentence Infilling with INter-SEntential Transformer (ACL 20)

- The author proposed a new task of **sentence infilling**, which 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.
- A framework called INSET was designed to decouple three aspects of the task (understanding, planning, and generation) and address them in a unified manner.

Thanks