Generating the middle sentence of two sentences using pre-trained models: a first step for text morphing

WANG Pengjie, WANG Liyan, Yves LEPAGE
EBMT/NLP Lab
Graduate School of Information, Production and Systems, Waseda University
{wangpengjie_0000@asagi.;wangliyan0905@toki.;yves.lepage@}waseda.jp

Abstract

Text morphing is a task of natural language processing. This task can be applied to many other tasks, such as generating data sets for writing aid systems or the study of sentence relationships. We propose an approach to generate the middle sentence of two input sentences. By iterating the steps, we show how to solve the text morphing task. We get a training dataset by using a naïve approach. We fine-tune a pre-trained GPT-2 model [7] to get a model that can generate middle sentences. Compared with the previous models, the results show that the generated sentences can be of high quality while maintaining the meaning relationship with the input sentences.

1 Introduction

Text morphing is a task of text generation that targets generating intermediate sentences that are fluent and smooth between two input sentences [4].

We denote two given input sentences as S_{start} and S_{end}. An example of text morphing is shown in Table 1.

S_{start}: i watched them .
S_1: i ’ll see them with tom .
S_2: i ’ll start with them .
S_3: i ’ll start with the questions .
S_{end}: let ’s start with the easy questions .

Table 1 An example of text morphing generated by our fine-tuned GPT-2 model. S_{start} and S_{end} are given sentences.

In this paper, we focus on generating a middle sentence between two input sentences.

2 Related work

For the text morphing generation task, the simplest solution is to decode the interpolation between two sentence vectors. For a normal autoencoder (AE), in most cases, the encoder generates unreasonable sentences when decoding interpolation.

Bowman et al. [2] proposed a Sentence Variational Autoencoder (SVAE) model to generate high-quality sentences using an embedding space. This approach successfully generates a series of coherent and reasonable sentences between two latent variables. However, due to a unique model in this approach, these generated sentences usually are far from the input sentences in meaning.

In addition to using a vector space, the text morphing task can be solved by editing techniques. Guu et al. [3] proposed a generative sentence model that edits a prototype sentence into a new sentence. Based on the idea of Guu et al. [3], Huang et al. [4] proposed a method to generate intermediate sentences by gradually editing the sentences from the start sentence to the end sentence. Unfortunately, their model is not open source, which prevents testing effectiveness.

GPT-2 is a pre-trained neural language model using the transformer model [8] and the autoregressive mechanism to predict the next token based on the previous contents. It provides an easy way to fine-tune the model to generate domain-specific text. We propose to fine-tune GPT-2 to get a model that generates middle sentences.

This paper will mainly compare our model with the AE and SVAE model in terms of middle sentence generation. In the following parts, we introduce the AE and SVAE models in detail.

2.1 Auto-encoder

An encoder converts sentences into vectors. It takes the middle interpolation between the two vectors as the middle vector, and then generates the middle sentence after decoding. We use a fastText pre-trained model [1] to compute word vectors. The sum of word vectors to represent the sentence vector. Such interpolation between two sentence vectors may not have a good effect after decoding because the information carried by interpolation is often difficult to decode into a reasonable sentence.

2.2 Sentence Variational Autoencoder

Bowman et al. [2] proposed the Sentence Variational Autoencoder (SVAE) model to solve some AE model generation problems.

The Variational Autoencoder (VAE) [5] replace the deterministic function in the standard auto-encoder with a learned posterior recognition model, $q(\vec{z}|x)$. $\vec{z}$ is called the latent variable. Bowman et al. [2] used a particular model to force the decoder to decode a reasonable sentence while decoding $\vec{z}$.

However, when an embedding is converted to $\vec{z}$, the $\vec{z}$ loses some original embedding information. Therefore, in subsequent generated tasks, the correlation between the inputs and generated sentences will be reduced.

3 Assessment

In this section, we introduce the metrics used to evaluate the results.

3.1 BERTScore

We use BERTScore [9] to evaluate the semantic similarity between two sentences. When the BERTScore between two sentences approaches 1, it indicates that the meaning of the two sentences tend to be the same. When the BERTScore is close to 0, it indicates that the meaning of the two sentences tend to be unrelated.

3.2 Grammar checker

We use GECToR[1] [6], a state-of-the-art grammatical error correction (GEC) model (as of January 2021), to check our generated sentences for grammatical correctness.

In the case of ignoring uppercase, if a sentence passes through the GECToR model without modifications, we assume that the sentence is correct.

3.3 Perplexity

We use perplexity (PPL) to determine whether our generated sentences are plausible. The sentences in the Tatoeba corpus that we use are mostly short. Therefore we calculate PPL in a 3-gram model.

3.4 Jaccard distance

Jaccard distance is a measure of the difference between two sets. We use it when evaluating the test results of text morphing.

We define:

- $d_j(S, E) = \dfrac{|S \cup E| - |S \cap E|}{|S \cup E|}$.
- In this formula, S stands for the set of tokens in the start sentence, and E stands for the set of tokens in the end sentence.

4 Methodology

We aim to obtain a model that can generate a middle sentence between two given input sentences.

4.1 Approach

Our approach mainly consists of the following three steps:

1. We use a pre-trained auto-encoder to generate the middle sentence and get the start-middle-end triplet.
2. We filter the sentence triplets and select the triplets that meet our requirements.
3. We fine-tune a pre-trained GPT-2 model with high-quality triplets to get a model that will generate a middle sentence between two given input sentences.

4.2 A method to select the triplet data

We use BERTScore to measure the semantic similarity between two sentences. The values of BERTScore in this paper are all the F1 values of BERTScore.

For ease of use, we use the following abbreviations in this paper:

- For a start-end sentence pair or a start-middle-end triplet, their BERTScore (start, end) is abbreviated as SE-BERTScore.
- For a start-middle-end triplet, the mean of the BERTScore (start, middle) and the BERTScore end, middle) is abbreviated as SME-BERTScore.
- For a start-middle-end triplet, the absolute value of the difference between the BERTScore (start, middle) and the BERTScore (end, middle) is denoted with DIF-BERTScore.

[1] https://github.com/grammarly/gector

Our definition of middle sentences comes from the following unique concepts of analogy:

Start : Middle :: Middle : End.

Specifically, it should meet the following requirements:

1. For a start-middle-end triplet, the SME-BERTScore should be larger than the SE-BERTScore.
2. For a start-middle-end triplet, the DIF-BERTScore should be as small as possible.

5 Experiment

Our experiment has two steps: start-middle-end triplet data collection and GPT-2 fine-tuning.

5.1 Dataset collection

We use a FastText pre-trained model to calculate the word vectors, and we represent a sentence vector by the sum of the word vectors. We split 80k sentences in the Tatoeba database into train/valid/test according to 80/10/10. We use this data to train a decoder. The parameters are as follows.

Hyperparameter	Number
GRU hidden size	300
Word embedding size	300
Embedding dropout	0.4
Num layers	1
Batch size	128
Learning rate	0.001

Table 2 The base model decoder settings.

Based on the interval of SE-BERTScore, we equally selected 1,110K sentence pairs from the Tatoeba corpus, as shown in Figure 1.

We split the 1,100K sentence pairs in the way of 100/11. The 1,000 K sentence pairs are used for generation tasks in the AE model, and 110K sentence pairs are used to test the middle sentence generation. In the 110K sentence pairs, we selected 10K on average to test text morphing.

5.2 Data procession

We collect start-middle-end triplets that meet the following four requirements:

- The middle sentence has not been modified by the grammar error correction model GECToR.
- The value of SME-BERTScore is larger than SE-BERTScore.
- The value of DIF-BERTScore is less than 0.05.
- The PPL value of the middle sentence is less than 30.

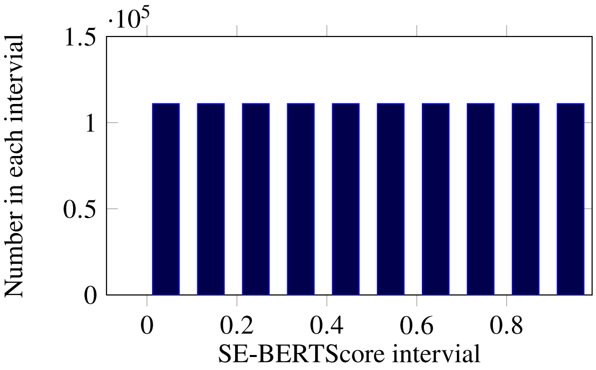

Figure 1 BERTScore range - Number histogram

We generated and equally selected 5,000 high-quality start-middle-end triplets according to the SE-BERTScore interval. We split the triples according 90/10 into train/valid data.

5.3 Fine-tuning GPT-2

The fine-tuning of GPT-2 is a simple procedure. We need to give the labeled start-middle-end triplet data to GPT-2 for fine-tuning.

The fine-tuning parameters are as follows. We use different learning rates for two experiments, Exp.1 and Exp.2.

Hyperparameter	Value
GPT-2 model	345M
Optimizer	adam
Train steps	1,800
Batch size	1
Learning rate (Exp. 1)	0.00002
Learning rate (Exp. 2)	0.00001

Table 3 The GPT-2 fine-tuning settings

6 Result and analysis

We use 110k sentences with the average distribution of SE-BERTScore to test each model. As a comparison, we use SVAE and the AE model as baselines. Because the SVAE structure is different from the AE model, we do not use parameters used in AE but use the default parameters provided by their implementation[2].

The achievement rate of strict standard (ARSS) is the proportion of the generated sentences that follows the following requirements:

- SME-BERTScore is larger than SE-BERTScore.
- DIF-BERTScore is less than 0.05.

Furthermore, the achievement rate of general standard

[2] https://github.com/timbmg/Sentence-VAE

SE-BS	Start	Middle			End
		AE	SVAE	Our Exp. 1	
1.0	i told mary to stay where she was .	i told mary where she was to stay .	i 'll never forget this incident .	i told mary to stay where she was .	i told mary to stay where she was .
0.8	mary says she 's feeling tired .	she says she 's feeling tired .	i 'm not going to miss you .	she says he 's feeling tired .	tom says he 's feeling tired .
0.6	she told me that she 'd wait for us .	she told me that she did that for us .	they 're not giving up .	she told me that she 'd do that for us .	she told me that she did that for us .
0.4	you 're worth gold .	you 're not gold .	you must not behave .	you 're not worth anything .	you 're not old .
0.2	google+ is a new social network .	tom is a new financial guy .	i 'm going to see him tomorrow .	tom is a new person .	tom is out of the tournament .

Table 4 Some middle sentences generation examples are selected according to the SE-BS interval. SE-BS stands for SE-BERTScore.

model		BERTScore (%)		PPL[3]	ER (%)	ARSS (%)	ARGS (%)
		SME	DIF				
Baseline	AE	52.43	20.47	53.40	40.27	9.49	33.22
	SVAE	22.13	**7.72**	**14.76**	**2.12**	4.63	15.83
Our model	Exp. 1	59.88	20.08	22.44	16.89	**13.95**	**43.09**
	Exp. 2	**62.06**	21.95	24.24	18.32	13.52	42.13

Table 5 The SME stands for the SME-BERTScore. The DIF stands for the DIF-BERTScore. The ER stands for the error rate that sentences cannot pass a grammar checker. The ARSS stands for the achievement rate of strict standards. The ARGS stands for the achievement rate of the general standards For SME and ARSS and ARGS, the higher it is, the better. For DIF, PPL, and ER, the smaller it is, the better.

(ARGS) is The proportion of the generated sentences that follows the following requirements:

- SME-BERTScore is larger than SE-BERTScore.
- DIF-BERTScore is less than 0.20.

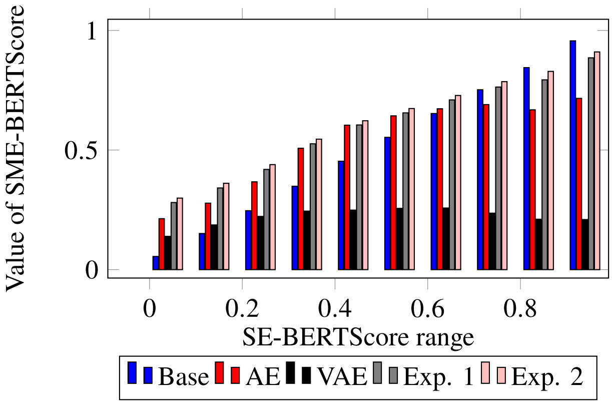

Figure 2 BERTScore distribution. The x-coordinate is the SE-BERTScore interval, and the y-coordinate is the average SME-BERTScore generated by each model within each interval.

In Figure 2, the base is the mean value of the SE-BERTScore in its interval. When the SME-BERTScore of a model is higher than the base value, it means that for this model, the generated sentence is semantically related to the start-end sentence pair and serves as a middle sentence between the start sentence and the end sentence.

According to the contents in Figure 2, Table 4, and Table-5, our model can generate high-quality middle sentences better than previous models (AE and VAE) under the strict and general standard conditions.

SE-BS	AE	SVAE	Exp. 1
0.60	Start: sami loved the lord .		
	the boy i recommend . the boy is open . the boy closed tv .	i 'm not a millionaire . i 'm not worried about it . i 'm not worried about it .	she loves the king . she loves the box . she opens the box .
	End: sami closed the box .		
0.20	Start: this building is new .		
	this building is new . this building is right now . tom 's not right now .	i 'm not going to compete with tom . i 'm not going to miss you . i 'm not going to miss you .	this is not new . this is not here . tom 's not here .
	End: tom isn 't here right now .		

Table 6 Some text morphing examples are selected according to the SE-BS interval.

Text morphing is a continuous process, and Jaccard distance can measure the degree of difference in form between two sentences. When the average Jaccard distance (JD) is lower, text morphing is smoother.

model		Avg. JD (%)	PPL
Baseline	AE	23.21	49.87
	SVAE	50.34	**16.36**
Our model	Exp. 1	**21.21**	24.00
	Exp. 2	24.24	24.00

Table 7 Avg. JD stands for the average Jaccard distance between the generate sentences and input sentences. The lower is meaning text morphing is more smooth.

The examples in Table 6 and the data in Table 7 show that our model can achieve smooth text morphing while maintaining high quality.

7 Conclusion

To summarize the contribution of this paper are as follows:

- We propose a new approach to solve the text morphing task.
- We fine-tune a pre-trained model that generates a middle sentence between inputs. This model can generate middle sentences from sentence pairs with different semantic similarity.
- The quality of middle sentences generated by our model is better than that of previous models, based on pretrained auto-encoders.

[3] When calculating PPL, we only count the cases where the number of start-end sentence tokens is larger than 2.

References

[1]Bojanowski, Piotr, Grave, Edouard, Joulin, Armand, and Mikolov, Tomas. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

[2]Bowman, Samuel R., Vilnis, Luke, Vinyals, Oriol, Dai, Andrew, Jozefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://www.aclweb.org/anthology/K16-1002.

[3]Guu, Kelvin, Hashimoto, Tatsunori B., Oren, Yonatan, and Liang, Percy. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018. doi: 10.1162/tacl_a_00030. URL https://www.aclweb.org/anthology/Q18-1031.

[4]Huang, Shaohan, Wu, Yuehua, Wei, Furu, and Zhou, M. Text morphing. *ArXiv*, abs/1810.00341, 2018.

[5]Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. In Bengio, Yoshua and LeCun, Yann, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[6]Omelianchuk, Kostiantyn, Atrasevych, Vitaliy, Chernodub, Artem N., and Skurzhanskyi, Oleksandr. GECToR - Grammatical Error Correction: Tag, Not Rewrite. In Burstein, Jill, Kochmar, Ekaterina, Leacock, Claudia, Madnani, Nitin, Pilán, Ildikó, Yannakoudakis, Helen, and Zesch, Torsten, editors, *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2020, Online, July 10, 2020*, pages 163–170. Association for Computational Linguistics, 2020. URL https://www.aclweb.org/anthology/2020.bea-1.16.pdf.

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

[8]Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Ł ukasz, and Polosukhin, Illia. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[9]Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q., and Artzi, Yoav. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.