Sentence Analogies for Text Morphing

Zhicheng Pan[1,*], Xinbo Zhao[1] and Yves Lepage[1]

[1]Waseda University, 2-7 Hibikino, Kitakyushu, 808-0135, Japan

Abstract

Text morphing is a Natural Language Processing (NLP) task which aims at generating sequences of fluent and smooth intermediate sentences between two input sentences, the start and end sentences. In this paper, we show how to use sentence analogies to augment data for this task. We rely on the notion of analogy to produce sequences of sentences exhibiting step-by-step transitions. We use these sequences to fine-tune a large-scale pre-trained language model that is used for text generation. The performance is evaluated by two criteria: fluency and transition smoothness on both the semantic and formal levels. Compared to a variational autoencoder generative model, our model is shown to generate smoother transitions, although the generated sentences are slightly less fluent.

Keywords

Sentence analogy, text morphing, data creation

1. Text Morphing

Text morphing is an NLP task that consists in generating a sequence of sentences that make the transition between a start sentence and an end sentence. Tables 1 and 2 show examples. This acceptation of text morphing is slightly different from the one in [1], where it is nearer to the meaning found in image morphing where information from two or several images is blended into one.

Obviously, text morphing can draw from techniques in Natural Language Generation (NLG) [2]. Traditional methods in NLG generally start from scratch. This is the case of left to right generation using latent sentence vector sampling [3]. In text morphing, we start from two given sentences and generate intermediate sentences. [4] proposes a generative language model for sentences that first samples a prototype sentence from a training corpus and then edits it into a new sentence. Based on that, [5] defined Text Morphing with the goal of generating intermediate sentences that are fluent and smooth between two input sentences.

[3] proposed an RNN-based variational autoencoder generative model which can generate coherent and diverse sentences using the latent space. It can also generate sentences from points between two sentence encodings. The model is called Sentence Variational Autoencoder (Sentence-VAE). Sentence-VAE incorporates distributed latent representations of entire sentences. It uses a continuous latent variable to capture global characteristics. The transitions

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.
panzhicheng@toki.waseda.jp (Z. Pan); zhao.symbol@fuji.waseda.jp (X. Zhao); yves.lepage@waseda.jp (Y. Lepage)

Table 1
Example text morphing sequence generated by the Sentence-VAE model (copied from [3]). The start and end sentences are given, the intermediate sentences S_1 to S_4 are generated sentences.

Start sentence	*he was silent for a long moment .*
S_1	*he was silent for a moment .*
S_2	*it was quiet for a moment .*
S_3	*it was dark and cold .*
S_4	*there was a pause .*
End sentence	*it was my turn .*

Table 2
Example text morphing sequence (copied from [5]). The start and end sentences are given, the intermediate sentences S_1, S_2, S_3 are generated sentences.

Start sentence	*The noodles and pork belly was my favourite .*
S_1	*The pork belly was my favourite .*
S_2	*The pork was very good .*
S_3	*The staff was very good .*
S_4	*The staff is very friendly .*
End sentence	*Love how friendly the staff is .*

obtained through variational latent space are smoother and the fluency of the generated sentences is higher. By searching paths through the latent space, it can generate coherent new sentences which interpolate between two already known sentences. Table 1 shows an example of a text morphing sequence generated by Sentence-VAE.

[4] proposed a new generative language model for sentences that first samples a prototype sentence from the training corpus and then edits it into a new sentence [4]. They perform experiments on the Yelp review corpus [6] and the One Billion Word Language Model Benchmark [7]. The result shows that the model they proposed improves the fluency of the generated sentences.

Building on the previous work, [5] took text editing a step further and proposed a novel model called Morphing Networks which can generate intermediate sentences by editing vectors obtained from a start sentence and an end sentence. The generated intermediate sentences should be fluent and the transitions should be smooth. They aim to gradually approach the end sentence by editing the start sentence step by step, that is, with increasing similarity to the end sentence. Each edit produces a new sentence, and ideally, the editing path is smooth because they only change a small part of the sentence, a few words or a phrase, with each edit. Table 2 shows an example that exhibits relatively smooth and natural transition between two sentences.

2. Proposed Method for Text Morphing

In nowadays NLP, it has become classical to fine-tune a large-scale pre-trained language model to perform a given downstream task, as this has been proven to be efficient in many cases. To perform fine-tuning in a supervised way, implies the use of a data set for the task in question. In our case, this means a data set of text morphing sequences.

In this paper, we show how to create text morphing sequences by exploiting the notion of analogy between sentences. This point is the original point in our proposed method.

Our method thus consists of the following two steps. Firstly and most importantly, we use the notion of sentence analogy (Subsection 2.2) to create a data set of text morphing sequences (Subsection 2.1). Secondly, we use this data set to fine-tune a large-scale pre-trained language model (Subsection(3.4) on the task of text morphing.

Below, we detail the original point in our method, i.e., the creation of a data set of text morphing sequences using the notion of analogies between sentences. We also explain how we solve sentence analogies.

2.1. Creating a Data Set of Text Morphing Sequences

We construct a data set of text morphing sequences by solving sequences of analogies between sentences. We start with a sentence analogy $A : B :: C_0 : x,$ where A, B and C_0 are sentences extracted from a data set of sentence analogies and x is unknown.[1] olving the equation delivers a sentence $x = C_1$. We recursively apply the process by replacing C_0 with C_1, etc., leaving A and B unchanged. In this way, we obtain a sequence of sentences C_0, C_1, …, C_n. It is a text morphing sequence where C_0 and C_n are the start and end sentences and C_1, ..., C_{n-1} are the intermediate sentences.[2] See Figure 1.

Since we are constantly replacing C_{i-1} with the next sentence C_i predicted by sentence analogy, the direction of changes in the entire sequence is given by the direction between A and B. Now, as the variation is, by definition of the analogy $A : B :: C_{i-1} : C_i,$ limited by the variation between A and B, the transitions should be smooth, if A and B are not too distant. The tool used to solve the sentence analogies should be responsible for the fluency of the generated sentences C_i.

To summarize, in this process, the sentence C_0 is transformed slowly step by step into C_n, along the direction defined by A and B. Notice that we give the start sentence C_0, but that we do not know in advance the end sentence C_n.

2.2. Solving Sentence Analogies

The previous process requires a tool to solve sentence analogies. Sentence analogies are more difficult to solve than word analogies (go is to $went$ as $walk$ is to $walked$ or $Tokyo$ is to $Japan$ as $Bejing$ is to $China$). The syntactic structure and semantic complexity of sentences makes the difficulty.

[1] http://lepage-lab.ips.waseda.ac.jp/en/projects/kakenhi-kiban-c-18k11447/ See Experimental Results.
[2] In our experiments, we set n to have 1 to 5 intermediate sentences. When n becomes larger, we observe that the same sentence may be repeated in the sequence of sentences.

$$\left.\begin{array}{ccccccc}
A & : & B & :: & C_0 & : x \Rightarrow x = & C_1 \\
A & : & B & :: & C_1 & : x \Rightarrow x = & C_2 \\
A & : & B & :: & C_2 & : x \Rightarrow x = & C_3 \\
\vdots & : & \vdots & :: & \vdots & : x \Rightarrow x = & \vdots \\
A & : & B & :: & C_{n-1} & : x \Rightarrow x = & C_n
\end{array}\right\} \Longrightarrow (C_0, C_1, C_2, \ldots, C_n)$$

Figure 1: Process of creating a text morphing sequence.

It 's really not that interesting .	:	*It 's really not that hot .*	::	*It 's not that bad .*	$: x$	$\Rightarrow$	$x =$	*It 's not that cold .*
You 're not from around here , are you ?	:	*You 're not staying here , are you ?*	::	*You 're confused again , are n't you ?*	$: x$	$\Rightarrow$	$x =$	*You 're disappointed , are n't you ?*

Figure 2: Semantico-formal analogies from the data set released in [10].

In early proposals to solve sentence analogies [8, 9], sentences have been considered as strings of words or characters. The disadvantage is that the semantics of sentences is not controlled. [10] proposed to combine both the form of sentences (strings of words) with the meaning of words (vector representations of words). They released a set of 5,600 so-called semantico-formal analogies in English. Examples are shown in Figure 2.

[11] proposed to learn the mapping between three vector representations of sentences (A, B and C) for and the vector representation of the sentence D solution of the analogy $A : B :: C : D$. The three vectors for A, B and C can be obtained from word or sentence embedding models. To decode the sentence D from its vector representation, they proposed a vec2seq model, implemented as a fully connected network, to map vector representations of sentences onto their corresponding sentences.

Here, we solve sentence analogies using yet another method described elsewhere [12]. It consists in fine-tuning a large-scale pre-trained model on the data set of semantico-formal analogies mentioned above. The fine-tuned model obtained can perform the task of solving sentence analogies directly in an end-to-end manner. Different language models were tested and the most efficient one was a fine-tuned GPT-2 model. We use that one in the experiments reported in this paper.

3. Experiment Settings

3.1. Data Used

The previously mentioned data set of semantico-formal analogies is used to create text morphing sequences that will be used to train a GPT-2 model for the task of text morphing. This data set was created from sentences extracted from the English part of the Tatoeba resource[3].

3.2. Assessment of Text Morphing Sequences

Assessment of text morphing is done according to two dimensions. Firstly, by the smoothness of the transitions between the sentences in the morphing sequences: two consecutive sentences should not differ by too much for the entire sequence to be considered smooth. Secondly, by the quality of each individual intermediate sentence: all sentences generated should sound natural, fluent, grammatical, in a word, it should be reasonable.

3.2.1. Transition Smoothness

We define it as the average of all edit distances between consecutive sentences in the morphing sequence. The edit distance between two strings gives the number of edit operations needed to transform a given string into another one. It is thus particularly well suited for our purpose. Here we use the Levenshtein distance [13] in which deletion, insertion, and substitution are the basic edit operations. Lower scores indicate smoother transitions.

3.2.2. Fluency of a Text Morphing Sequence

Perplexity, as classically used in language modelling, is a measure of the reasonableness of sentences. We thus define the fluency of a text morphing sequence as the average of the perplexity scores over all intermediate sentences (excluding the start and end sentences). A lower score indicates higher fluency.[4]

3.3. Creation of a Data Set of Text Morphing Sequences Using Sentence Analogies

As mentioned at the end of Section 2.2, to solve sentence analogies, we fine-tune a pre-trained language model, GPT-2 [14], on the task of solving sentence analogies. The sentence analogies used during this training are from the semantico-formal analogy data set mentioned in Section 2.2.

To create a data set of text morphing sequences, we then use each semantico-formal sentence analogy as a starting point as described in Section 2.1 and illustrated in Figure 1.

We asses the quality of the created text morphing sequences with the metrics introduced in Section 3.2, but, in addition, we compare with an existing model.

[3]https://tatoeba.org/

[4]We use the a 3-gram language model, trained on the Tatoeba corpus, with the KenLM toolkit https://github.com/kpu/kenlm to compute the perplexity of the sentences.

Table 3
The GPT-2 fine-tuning settings for text morphing.

Hyperparameter	Value
GPT-2 model	345M
Optimizer	adam
Batch size	1
Learning rate (LR=2)	0.00002

[5] did not release their code, although they claim better results than the Sentence-VAE model [3]. The code for this latter model is available[5]. So we adopt it as our baseline. In our experiments, we trained the Sentence-VAE model using the Tatoeba data set from which the above-mentioned semantico-formal analogies were extracted. By exploring the paths between the start and end sentences created with our method, in the latent space of the obtained Sentence-VAE model, we can generate a certain number of coherent sentences, which constitute a text morphing sequence.

In this way, we can compare the transition smoothness and the fluency of two comparable sets of text morphing sequences, created from the same start and end sentences, by two methods, the Sentence-VAE model, and our proposed method.

3.4. Fine-Tuning a Pre-Trained Model with the Data Set of Text Morphing Sequences Created Using Sentence Analogies

We fine-tune the pre-trained GPT-2 model using the sentence sequences generated in the previous section. For comparison, as in the previous section, we still use the Sentence-VAE model as a baseline model. Due to limitations in memory, we choose the medium-sized GPT-2 model (345M). The GPT-2 fine-tuning parameters are shown in Table 3. For the baseline model, we trained the Sentence-VAE model using the Tatoeba corpus dataset which consists of 110,000 English sentences.

GPT-2 [14] is a large transformer-based language model created by OpenAI. GPT-2 uses the Decoder structure of the Transformer [15], with some changes to the Transformer Decoder. They verified that unsupervised language modeling is able to learn the features required for supervised tasks. GPT-2 pretraining uses the foregoing to predict the next word, which is suitable for text generation tasks since text generation usually generates the next word based on currently available information.

GPT-2 is a large model based on transformer training on a very large dataset with a large scale, and GPT-2 has a good performance in text generation, both in terms of contextual coherence and sentiment expression.

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

Table 4
Quality of text morphing sequences. On the *left*, in the creation of the data set to be used in fine-tuning. On the *right*, in performing the task of text morphing with the fine-tuned model. For both measures of transition smoothness and fluency, the lower, the better.

	Data Set Creation		Text Morphing	
	Transition Smoothness	Fluency	Transition Smoothness	Fluency
Sentence-VAE [3]	3.52	**1.35**	3.73	**1.36**
Our method	**1.31**	1.57	**0.72**	1.75

Table 5
Statistics of created morphing sequences dataset.

	sequences	sentences	words/sent	chars/sent
generated dataset	5,228	26,140	6.41	23.82

4. Results

4.1. Results for the Creation of the Data Set of Text Morphing Sequences

The quality of the created text morphing sequences, that will be used afterwards to train a large-scale language model for the task of text morphing, is shown in Table 4. Our proposed approach delivers smoother sentences which are semantically relatively correct, in comparison with the Sentence-VAE model proposed in [3] for generating morphing sequences.

Table 4 shows that the transition smoothness (average of edit distance between consecutive sentences) of Sentence-VAE is 3.52, while it is 1.31 with our proposed method. This means that for each transition, the Sentence-VAE model changes on average three and a half words on average, while our proposed method changes 1.3 words only, less than half in comparison. The average number of words per sentence being 6.7, the baseline method changes half the sentence at each transition. Our method makes more subtle and smoother changes.

The fluency, as measured by perplexity, is 1.35 in the method using Sentence-VAE, while it is 1.57 in our method (the scores are small because the sentences are short). According to these numbers, the sentences generated by the Sentence-VAE model are more reasonable, but whether there is a real difference may be disputable. We conclude that, in comparison with the Sentence-VAE model, our proposed method delivers smoother sentences that are relatively fluent.

The following Table 5 shows basic statistics of our created dataset. An example of generated morphing sentences is given in Table 6 below.

Table 6
Morphing sequences obtained with the Sentence-VAE method (on the *left*) and our proposed method (on the *right*) for the same start and end sentences.

Sentence-VAE [3]	Our method [this paper]
I 'm ready to go .	*I 'm ready to go .*
what do you think of this is ?	*I really have to go .*
this is a good textbook .	*I really need to go .*
i have a lot of money in this store .	*I need to go somewhere.*
I have to go somewhere .	*I have to go somewhere .*

4.2. Results for the Text Morphing Task

The quality of text morphing is shown in the same table as before, Table 4. The results of this experiment are similar to those obtained in the previous section when creating a data set of text morphing sequences. This indicates that our trained model can deliver smoother sentences which are semantically relatively correct, in comparison with the Sentence-VAE model for the task of text morphing.

The transition smoothness of the Sentence-VAE model is 3.73, while it is 0.72 with our proposed method. The previous remarks made above apply similarly here for this model. It is not a surprise as we use it here in the same way as before. Our proposed method shows improvement in transition smoothness relatively to the creation of text morphing sequences: the average edit distance between two consecutive sentences has been almost divided by two.

The perplexity of the method using the Sentence-VAE model is 1.36, while the perplexity with our proposed method is 1.75. Again, there is no difference between the scores in the data creation step and the text morphing tesk for the Sentence-VAE model because we use it in the same way in both cases. Our proposed method generates sentences with a slightly worse perplexity in the text morphing task compared with the creation of text morphing sequences using sentence analogy. However, again, we can conclude that our proposed fine-tuned model delivers sentences which are relatively fluent, but smoother, in comparison with the Sentence-VAE model.

4.3. Discussion

When creating text morphing sequences, we observed that, sometimes the same sentences were generated repeatedly or several sentences were generated alternately. We explain these phenomena by the relative shortness of the sentences used. The sentences contained in our data set are less than 10 words long. Shorter sentences allow for fewer options for changes when the text is morphed, and sometimes repetition occurs, which induces no change.

Table 7
Examples of text morphing sequences (of length 3) generated by the fine-tuned GPT-2 model. Start sentences on the first row, end sentences on the last row.

I deserve this .	*I really do not know .*	*I see the problem .*
I do not deserve this . *I deserve that .* *I do not need that .*	*I do not know .* *I do not know anything .* *I do not know .*	*I know the truth .* *I know the problem .* *I know the truth .*
I do not need a girlfriend .	*I do not understand anything .*	*I know the solution .*

5. Conclusion

We proposed to perform text morphing by fine-tuning a large-scale pre-trained language model on the task, as is classical nowadays in NLP. But for that, data was needed. We relied on analogies to create text morphing sequences. We proposed an original method which consists in starting with an analogical equation and in letting the solver perform changes in the direction defined by the two terms on the left of the analogical equation. Variations are obtained step by step and this results in text morphing sequences.

The performance of the fine-tuned model was evaluated with transition smoothness and fluency. Our model achieved more than three times smoother transitions than the baseline we considered, the Sentence Variational Autoencoder generative model. However, the baseline was shown to generate slightlymore fluent sentences than our proposed model.

References

[1] R. A. Connor, Multi-stage text morphing, patent US 2011/0184725 A1, 2011. URL: https://patents.google.com/patent/US20110184725.

[2] A. Gatt, E. Krahmer, Survey of the state of the art in natural language generation: Core tasks, applications and evaluation, Journal of Artificial Intelligence Research 61 (2018) 65–170.

[3] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, (CoNLL2016), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 10–21. URL: https://www.aclweb.org/anthology/K16-1002. doi:10.18653/v1/K16-1002.

[4] K. Guu, T. B. Hashimoto, Y. Oren, P. Liang, Generating sentences by editing prototypes, Transactions of the Association for Computational Linguistics 6 (2018) 437–450. URL: https://www.aclweb.org/anthology/Q18-1031. doi:10.1162/tacl_a_00030.

[5] S. Huang, Y. Wu, F. Wei, M. Zhou, Text morphing, ArXiv (not published elsewhere) abs/1810.00341 (2018).

[6] N. Asghar, Yelp dataset challenge: Review rating prediction, 2016. URL: https://arxiv.org/abs/1605.05362.
[7] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, One billion word benchmark for measuring progress in statistical language modeling, CoRR abs/1312.3005 (2013).
[8] M. Nagao, A framework of a mechanical translation between Japanese and English by analogy principle, in: A. Elithorn, R. Banerji (Eds.), Proceedings of the international NATO symposium on Artificial and human intelligence, Elsevier Science Publishers, NATO, 1984, pp. 173–180. URL: http://www.mt-archive.info/Nagao-1984.pdf.
[9] Y. Lepage, G. Peralta, Using paradigm tables to generate new utterances similar to those existing in linguistic resources, in: Proceedings of the 4th internation conference on Language Resources and Evaluation (LREC 2004), volume 1, Lisbon, 2004, pp. 243–246.
[10] Y. Lepage, Semantico-formal resolution of analogies between sentences, in: Z. Vetulani, P. Paroubek (Eds.), Proceedings of the 9th Language & Technology Conference (LTC 2019) – Human Language Technologies as a Challenge for Computer Science and Linguistics, 2019, pp. 57–61. URL: http://lepage-lab.ips.waseda.ac.jp/media/filer_public/32/04/32049346-75dd-4bd1-93cc-ae221e49a2e9/ltc-005-lepage.pdf.
[11] L. Wang, Y. Lepage, Vector-to-sequence models for sentence analogies, in: IEEE (Ed.), Proceedings of the 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS 2020), 2020, pp. 441–446. URL: https://ieeexplore.ieee.org/document/9263191. doi:10.1109/ICACSIS51025.2020.9263191.
[12] L. Wang, Z. Pan, H. Xiao, Y. Lepage, Solving sentence analogies by using embedding models combined with a vector-to-sequence decoder or by fine-tuning pre-trained language models, 2022. Under review.
[13] V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, Soviet Physics-doklady 10 (1966) 707–710.
[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, Technical Report, OpenIA, 2019.
[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems (NIPS 2017), volume 30, 2017, pp. 6000–6010.