Test-time Augmentation for Factual Probing

Go Kamoda[1] Benjamin Heinzerling[2,1] Keisuke Sakaguchi[1,2] Kentaro Inui[1,2]
[1]Tohoku University [2]RIKEN
go.kamoda@dc.tohoku.ac.jp benjamin.heinzerling@riken.jp
{keisuke.sakaguchi, kentaro.inui}@tohoku.ac.jp

Abstract

Factual probing is a method for checking if a language model "knows" certain world knowledge facts. A problem in factual probing is that small changes to prompts can result in large output changes. Previous work aimed to alleviate this problem by optimizing prompts via text mining or finetuning. However, such approaches are relation-specific and do not generalize to unseen relations types. Here, we propose to use test-time augmentation (TTA) as a relation-agnostic method for reducing sensitivity to prompt variations by automatically augmenting and ensembling prompts at test time. Experiments show that, while TTA reduces overconfidence in incorrect generations, accuracy increases only in few cases. Error analysis reveals the difficulty of producing high-quality prompt variations as the main challenge for TTA.

1 Introduction

Pre-trained language models (LMs) such as BERT [1] and T5 [2] implicitly encode world knowledge from the training corpus in their parameters. Petroni et al. [3] demonstrated that world knowledge can be retrieved from a masked LM via cloze-style prompts, e.g., "The capital city of Alaska is [MASK]."

However, since small changes to the prompt can lead to drastic output changes [4] it is difficult to distinguish whether the model did not learn a fact during pre-training or if it did, but does not output the correct answer with the given prompt. Subsequent work aimed at finding better prompts for factual probing, typically by employing supervised learning to find an optimal input token sequence of tokens for a given relation [5, 6, 7]. Since these approaches require supervision for each relation, they do not generalize to unseen relation types, and hence are not practically appealing.

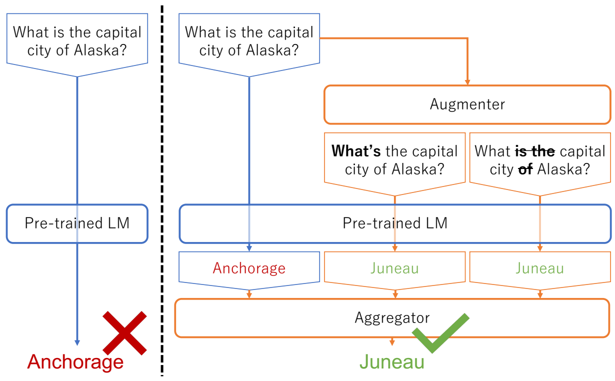

Figure 1 With (right) and without (left) TTA for factual probing. The orange components are added in our method. The Augmenter automatically augments the original prompt. The aggregator takes the generations from all prompts as input and outputs one generation with the highest score.

In this paper, we apply the idea of test time augmentation (TTA) to the factual probing task. TTA is a method used in the field of computer vision, which augments input images through simple operations (flipping the image, changing the contrast, etc.) at test time. The augmentations are helpful in covering overconfident and incorrect outputs. Krizhevsky et al. [8] used test-time augmentation for ImageNet classification, and subsequent work in the field of computer vision [9, 10, 11] utilizes test-time augmentation to get better performance in accuracy or robustness. The motivations are common with factual probing tasks; we also want language models to be robust to wordings and be less overconfident. To apply TTA to the task, an augmenter and an aggregator are added to the stream of the model prediction (Figure 1). First, the input prompt is automatically augmented by the augmenter. The augmented prompts are then individually fed to a model. The aggregator will aggregate the model's output to determine the final result. We 1) evaluated the result's exact match accuracy and investigated the impact of the number of augmented prompts on the accuracy and 2) inspected the change in the confidence of the generations.

Our results showed that the greater the number of augmented prompts, the better the performance when implementing TTA. TTA was also effective at reducing the number of overconfident and incorrect outputs. In terms of accuracy, TTA was only effective in a few cases. We analyzed the cause of this to be the poor quality of augmented prompts declines the accuracy of the model without TTA.[1)]

2 Setup

2.1 Dataset

We constructed a dataset of 12,500 relational facts from wikidata. Each fact is composed of a subject, a relation, and an object. We filtered out facts with multiple objects to collect unique facts. To reduce the bias of the distribution of objects, we adopted truncated sampling to select 500 instances per predicate. We provided a human-made prompt template for each relation (e.g., "What is the capital city of {subject}?").

2.2 Augmenter

We used three types of prompt augmentations. The first type is synonym replacement, which replaces words in the input prompt with a synonym. For instance, the word "buried" was replaced with "inhumed" by this type of augmentation[2)]. Candidate synonyms are provided from GloVe [12] embedding or WordNet [13]. The second augmentation method we used is back-translation. We used French, Russian, German, Spanish, and Japanese as the target language. The third augmentation method is stopwords-filtering.

From a single original input, 1 prompt is augmented by stopwords-filtering, and 4 prompts are augmented by each of the other methods, providing a maximum total of 29 augmented prompts.

2.3 Model

We ran experiments on the following pre-trained language models: Google's T5 for Closed Book Question Answering (small, large, 3b, 11b)[14], Google's FLAN models (small, xl)[15], and T0_3B model from Big Science[16].

Models decode with beam-search where the beam size

1) https://github.com/cl-tohoku/TTA4FactualProbing

2) Prompt #7 in table 5 in appendix

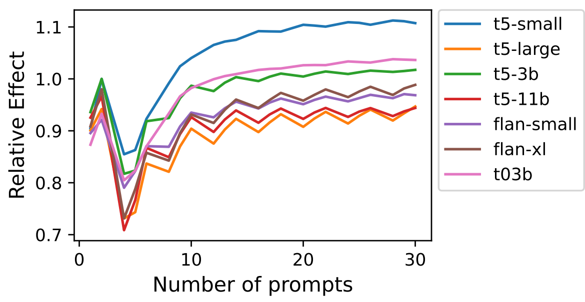

Figure 2 The relation between the number of prompts and the average relative effect of TTA. A relative effect of 1.0 means no change in accuracy between with and without TTA.

is fixed to 10 and return generated sequences with scores. Scores are in the order of log-likelihood (negative), and the exponentiated scores are in the order of probability.

2.4 Aggregator

We aggregate generations by taking the sum of generation probability.

$$s(y'|x,r) = \sum_{i=1}^{K} P_{\mathrm{LM}}(y'|p_i) \tag{1}$$

$$y = \operatorname{argmax}(s(\cdot|x,r))_{y'} \tag{2}$$

The model output with generation probabilities (P_{LM}) for all augmented prompts (p_i) will be fed into the aggregator to choose one final prediction. The aggregator recalculates the generation score (s) by taking the sum of the generation probabilities of identical generations (Eq.1). The final prediction of an object (y) for the fact with subject x and relation r is the one with the highest score (Eq.2).

2.5 Evaluation Metric

We measure the effect of TTA by the relative difference of exact match accuracy. To prevent division by zero, a constant of 1 is added to both the numerator and the denominator (Eq.3). The metric judges correct only if the final generation outputted is identical to the gold label provided in the dataset. Evaluation on flan models is an exception, and we adopt case-insensitive match accuracy.

$$\text{relative effect} = \frac{(\#\text{ corrects w/ TTA}) + 1}{(\#\text{ corrects w/o TTA}) + 1} \tag{3}$$

3 Results

By augmenting prompts, we got 9 types of prompts (1 original, 1 stopwords-filtering, 2 synonym replacement, 5

Table 1 Confusion matrix of accuracy. (model: t5-small)

		w/ TTA	
		Correct	Incorrect
w/o	Correct	7.1	1.6
TTA	Incorrect	2.5	89

Table 2 Confusion matrix of accuracy. (model: t5-11b)

		w/ TTA	
		Correct	Incorrect
w/o	Correct	28	3.7
TTA	Incorrect	1.9	66

back-translation). We evaluated all 511 ($= 2^9 - 1$) combinations of the 9 augmentation types for each model. Figure 2 shows relationships between the number of prompts and the average relative effect of TTA. As the number of prompts increases, the accuracy converges to a particular value, suggesting that the more augmentation we provide, the greater the accuracy gets. On the t5-small model, TTA raised the model accuracy as expected. There was a small improvement on the T0_3b and t5-3b models. In other models, TTA could not increase the accuracy even when the original prompt is augmented into 30 prompts.

Table 1 and Table 2 shows the confusion matrix of the t5-small model, which had the greatest increase in accuracy when applied TTA, and the t5-11b model which has the largest number of parameters out of all models we investigated. The tables compare the number of corrects/incorrect with and without TTA. Data for "with TTA" is the accuracy after aggregating all 30 prompts.

3.1 Positive Effects

Table 3 shows one example of TTA increasing the accuracy on the t5-11b model. The model generated an incorrect label from the original prompt but was able to cover it up by generating the gold label from some of the augmented prompts. This is an ideal behavior when applying TTA to the factual probing task.

Confidence One of the aims to apply TTA was to reduce the number of overconfident and incorrect generations. In this section, we investigate the effect of TTA on the confidence of the model.

In our method, the aggregator re-ranks generations by calculating the sum of generation probability for all identical generations for each fact instance. The confidence of

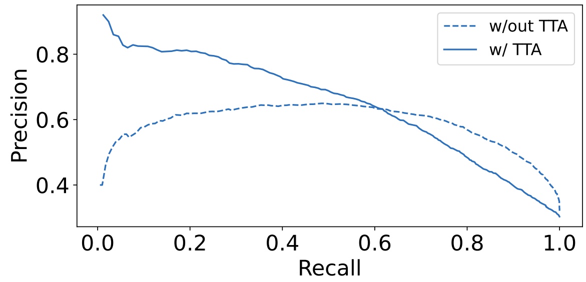

Figure 3 Precision-recall curve when changed confidence threshold (model: t5-11b). Low recall means a high confidence threshold.

the aggregator can be expressed by the ratio of the score to the final output and the sum of the calculated scores (Eq.4).

$$\text{confidence} = \frac{\text{score}_{\text{final output}}}{\sum_{\text{candidates}} \text{score}} \tag{4}$$

After we calculated the confidence, we put the rankings of the confidence into bins of size 50 without considering whether the generation was correct or incorrect. We express $\text{bin}_i (1 < i < 250, i \in \mathbb{N})$ as the bin with the i^{th} highest confidence, $\#\text{corrects}_i$ as the number of correct generation in bin_i, and $\#\text{incorrects}_i$ as the number of incorrect generation in bin_i. When we treat i as a confidence threshold, precision and recall can be defined by Eq.5 and Eq.6.

$$\text{Precision}_i = \frac{\sum_{j=1}^{i} \#\text{corrects}_j}{\sum_{j=1}^{i} \#\text{corrects}_j + \sum_{j=1}^{i} \#\text{incorrects}_j} \tag{5}$$

$$\text{Recall}_i = \frac{\sum_{j=1}^{i} \#\text{corrects}_j}{\sum_{j=1}^{250} \#\text{incorrects}_j} \tag{6}$$

Figure 3 shows the calculated precision-recall curve for i in the range 1-250. Without TTA, the model precision was relatively low when the confidence threshold was high (= when the recall was small). This means that the model is outputting incorrect generations with high confidence. After applying TTA, the precision of the left side of the figure improved, indicating that TTA effectively reduced overconfident incorrect generations. In addition, the precision rose monotonically as the confidence threshold increased. This suggests that confidence can work as a convenient parameter to control model precision.

3.2 Negative Effects

When the original prompt elicited the gold label but the aggregation result outputs the incorrect label, the ac-

Table 3 Example of TTA improving performance. The gold label for this fact instance is " South America ", and the aggregator returned " South America ".

#	Type	Prompt	Generation
0	Original	What continent is Para District located on?	Africa
2	WordNet	What continent is Para District based on?	North America
12	bt-fr	What continent is the Para District located on?	South America
15	bt-ru	What continent is Pará County on?	South America
18	bt-de	On which continent is the Para District located?	South America

Table 4 Example of TTA degrading performance. The gold label for this fact instance is " Heidelberg ", but the aggregator returned " Erlanden, Germany ". The results of other prompts are in the appendix.

#	Type	Prompt	Generation
0	Original	Where is Hans-Georg Gadamer buried?	Heidelberg
1	Embedding	Accordingly is Hans-Georg Gadamer buried?	in Bonn
6	WordNet	Where is Hans-Georg Gadamer inhume?	Erlangen, Germany
11	bt-fr	Where's Hans-Georg Gadamer buried.	Erlangen, Germany
15	bt-ru	Where's Hans-George Gadmer buried?	Wiesbaden, Baden-Württemberg
17	bt-de	Where's Hans-Georg Gadamer buried?	Erlangen, Germany
21	bt-es	Where is Hans-Georg Gadamer buried?	Heidelberg
25	bt-ja	Where are the goodly places? where is the plac...	Mount of Olives
29	no-stopwords	Where Hans-Georg Gadamer buried?	in Marburg

curacy declines. Table 4 shows an example of instances that caused the accuracy to decline. Only 9 out of 30 prompts are on the table, and others are in the appendix. The 30 prompts generated 18 unique generations as the generation with the highest score. 7 prompts generated "Erlanden, Germany", and 4 prompts generated "Heidelberg", the gold label.

When we look at the prompts in table 4, not all augmented prompts keep the semantics of the original prompt. For example, prompt #1 in the table replaced the word "Where" with "Accordingly", which is not a natural synonym. Prompt #7 uses the word "inhume", which is a synonym of "bury", but the use is grammatically incorrect. Prompt #15 is asking about a person whose name is "Hans-George Gadmer" and not "Hans-Georg Gadamer". The augmented prompt by back-translation with Japanese as the target language is hardly a paraphrase of the original prompt. Although the purpose of implementing TTA is to cover up incorrect generations by some prompts, expecting the behavior using numerous augmented prompts with low quality is a harsh setting. The augmented prompts are expected to maintain the semantic components of the original prompts while varying the syntactic structure or vocabulary. It is a crucial issue to produce paraphrases with high quality.

4 Conclusion

We applied the idea of test-time augmentation to the factual probing task on pre-trained language models to increase the model's accuracy and robustness. We used synonym replacement, back-translation, and stopwords-filtering to create augmented prompts automatically. Out of the 7 models we investigated, TTA had a significant effect only on the t5-small model. On other models, TTA had a negative effect in terms of accuracy. The unsound effect of TTA is anticipated to result from the poor quality of automatically augmented prompts. We want to examine deeper into the quality of augmented prompts. Searching for a better method to produce high-quality paraphrases would also be indispensable.

In terms of confidence of the model, our results showed that TTA effectively reduced the number of incorrect outputs. In addition, results indicated that TTA enables confidence to be the parameter that could conveniently control the model precision.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR20D2, Japan, and JSPS KAKENHI Grant Numbers 21K17814, 21K21343, and 22H00524. We would like to thank the members of the Tohoku NLP for their frequent participation in discussions during the course of this research.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. **CoRR**, Vol. abs/1810.04805, , 2018.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[3] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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)**, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[4] Boxi Cao, Hongyu Lin, Xianpei Han, Fangchao Liu, and Le Sun. Can prompt probe pretrained language models? understanding the invisible risks from a causal view. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 5796–5808, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[5] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics.

[6] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How Can We Know What Language Models Know? **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 423–438, 07 2020.

[7] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. In **North American Association for Computational Linguistics (NAACL)**, 2021.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In **Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1**, NIPS'12, pp. "1097–1105", Red Hook, NY, USA, 2012. Curran Associates Inc.

[9] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sebastien Ourselin, and Tom Vercauteren. Test-time augmentation with uncertainty estimation for deep learning-based medical image segmentation, 2018.

[10] Fábio Perez, Cristina Vasconcelos, Sandra Avila, and Eduardo Valle. Data augmentation for skin lesion analysis. 08 2018.

[11] Kazuhisa Matsunaga, Akira Hamada, Akane Minagawa, and Hiroshi Koga. Image classification of melanoma, nevus and seborrheic keratosis by deep neural network ensemble. 03 2017.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[13] Chakaveh Saedi, António Branco, João António Rodrigues, and João Silva. WordNet embeddings. In **Proceedings of the Third Workshop on Representation Learning for NLP**, pp. 122–131, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[14] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics.

[15] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In **International Conference on Learning Representations**, 2022.

[16] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In **International Conference on Learning Representations**, 2022.

Table 5 Example of TTA declining the accuracy. The gold label for this fact instance is " Heidelberg ", but the aggregator returned " Erlangen, Germany ". The results of other prompts are in the appendix.

#	Type	Prompt	Generation
2	Embedding	Consequently is Hans-Georg Gadamer buried?	in Bonn
3	Embedding	Where poses Hans-Georg Gadamer buried?	Erlangen
4	Embedding	Where represents Hans-Georg Gadamer buried?	Erlangen, Germany
5	WordNet	Where is Hans-Georg Gadamer immerse?	German philosophical neo-Kantia
7	WordNet	Where is Hans-Georg Gadamer inhumed?	Erlangen
8	WordNet	Where is Hans-Georg Gadamer swallow?	the throat
9	bt-fr	Where is Hans-Georg Gadamer buried?	Heidelberg
10	bt-fr	Where's Hans-Georg Gadamer buried?	Erlangen, Germany
12	bt-fr	Where's Hans-Georg Gadamer buried in the house?	in the garden
13	bt-ru	Where is Hans-George Gadmer buried?	Basel
14	bt-ru	Where is Hans-George Gadamer buried?	Erlangen, Germany
16	bt-ru	Where is Hans-Gorg Gadmer buried?	Wiesbaden, Baden-Württemberg
18	bt-de	Where is Hans-Georg Gadamer buried?	Heidelberg
19	bt-de	Where was Hans-Georg Gadamer buried?	in the Munich Waldfriedhof
20	bt-de	Where will Hans-Georg Gadamer be buried?	in the Cathedral of Speyer
22	bt-es	Where's Hans-Georg Gadamer buried?	Erlangen, Germany
23	bt-es	Where is Hans-Georg Qadamer buried?	In a Munich cemetery
24	bt-es	Where is Hans-Georg Gadhamer buried?	Innsbruck
26	bt-ja	Where are the goodly places? Where is the plac...	Bethel
27	bt-ja	Where are the goodly places? where are the pla...	the mountain of God
28	bt-ja	Where are the goodly places? where is the plac...	the place of his fathers

Appendix

Augmentation Methods

Synonym Replacement We use a python library "TextAttack". For synonym replacement using wordnet, we use WordNetAugmenter provided in the library. For synonym replacement using GloVe embedding, we use the transformation method WordSwapEmbedding to create an augmenter.

Back-translation We first translate the original prompt to 8 candidates in the target language. Each candidate is then translated back into 8 candidates in the source language, getting 64 back-translated prompt candidates in total. We adopt the round-trip probability as the score of the back-translated prompt candidates and select 4 candidates using the aggregation method mentioned in section 2.4. For translations, we used Marian MT models [3)]. The Marian MT models occupy roughly the same memory size as the t5-small model.

Stopwords-filtering This method drops stopwords and diacritics from the original prompt. We use a python library "Texthero" for the processing.

Aggregator

Counting the number of appearances in the generations is one method of aggregation. We did not use count-based aggregation because the possibility of having multiple generations with the same counts is high. The phenomenon is predicted to occur more when we make the model output more sequences for each prompt. In addition, this method cannot take confidence into account as all generations by beam-search are equally weighted.

Result

Table 5 shows the result of augmented prompts that we did not display on table 4.

3) https://github.com/Helsinki-NLP/Opus-MT