Testing Untestable Neural Machine Translation: An Industrial Case

Wujie Zheng*, Wenyu Wang†, Dian Liu*, Changrong Zhang*, Qinsong Zeng*,
Yuetang Deng*, Wei Yang‡, Pinjia He§, Tao Xie†
* Tencent Inc., China
† University of Illinois at Urbana-Champaign, USA
‡ University of Texas at Dallas, USA
§ ETH Zurich, Switzerland

***Abstract*—Neural Machine Translation (NMT) has shown great advantages and is becoming increasingly popular. However, in practice, NMT often produces unexpected translation failures in its translations. While reference-based black-box system testing has been a common practice for NMT quality assurance during development, an increasingly critical industrial practice, named *in-vivo testing*, exposes unseen types or instances of translation failures when real users are using a deployed industrial NMT system. To fill the gap of lacking test oracles for in-vivo testing of NMT systems, we propose a new methodology for automatically identifying translation failures without reference translations. Our evaluation conducted on real-world datasets shows that our methodology effectively detects several targeted types of translation failures. Our experiences on deploying our methodology in both production and development environments of WeChat (a messenger app with over one billion monthly active users) demonstrate high effectiveness of our methodology along with high industry impact.**

***Keywords*—neural machine translation, in-vivo testing, AI quality assurance**

I. Introduction

Neural Machine Translation (NMT) [1] has shown great advantage and is becoming increasingly popular. However, in practice, NMT often produces unexpected translation failures in its translations. Various past incidents [2] show that translation failures could lead to serious consequences. As a common in-house black-box testing approach aiming to eliminate in-field NMT translation failures, developers construct test sets from parallel corpora (i.e., large collections of bilingual text pairs containing original texts and one or more corresponding reference translations). Developers typically use special test oracles based on translation quality scores (e.g., BLEU score [3]) to determine whether a test case passes.

While black-box system testing has been shown useful for in-house quality assurance of industrial NMT systems, *in-vivo testing* [4], which executes test cases in the production environment, is also becoming increasingly critical in industrial cases. It exposes unseen types or instances of translation failures not revealed by in-house black-box system testing and enables developers to monitor and handle translation failures instantly in the production environment. Unfortunately, existing in-house test oracles cannot be applied to in-vivo testing directly due to lacking reference translations. Existing heuristics [5] on speculating the expected outputs for unlabeled data on classification/prediction-oriented machine learning systems are also ineffective due to the intractability of natural languages.

To fill the gap of lacking test oracles for in-vivo testing of industrial NMT systems, we propose a new methodology for automatically identifying translation failures without reference translations; our methodology can directly serve as a test oracle for in-vivo testing. Our experiences on deploying our methodology in both production and development environments of WeChat (a messenger app with over **one billion** monthly active users) demonstrate that our methodology is both practical and scalable.

II. Our Methodology

The key insight underlying our proposed methodology is that there are some properties of natural language translation that can be checked systematically. If there is any violation of these properties in the translation under inspection, there are likely translation failures. In addition, we can leverage the information from the input (i.e., the original text) of the NMT system to provide hints for finding translation failures, whereas such information is overlooked when the translation quality score is calculated. An example property that generally holds for translations is that the original text and the translation generally have one-to-one mappings in terms of their constituents, e.g., words/phrases. As initial efforts following this methodology, we develop a technique for detecting two specific types of violations of this property: (1) *under-translation*, where some words/phrases from the original text are missing in the translation, and (2) *over-translation*, where some words/phrases from the original text are unnecessarily translated multiple times. Our technique leverages Item-based Collaborative Filtering [6].

III. Baseline Evaluation

We evaluate our technique on four real-world datasets against two baseline techniques for comparison: *primitive dictionary looking-up* (denoted as 'std-dict'), in which we use only one existing generic translation dictionary to support detection, and *word alignment* [7] (named as 'word-align') from the traditional Statistical Machine Translation (SMT) [8]. We train our technique and the *word alignment* baseline technique with 16 million sentence pairs crawled from various sources (e.g., example word/phrase usages in dictionaries) on

TABLE I: Results of under-translation detection on the evaluation datasets

	proposed			std-dict			word-align		
	P	R	F	P	R	F	P	R	F
ench_news	**0.51**	0.69	**0.59**	0.28	**1.00**	0.43	0.35	0.85	0.49
chen_news	**0.43**	0.65	**0.52**	0.16	**1.00**	0.27	0.18	**1.00**	0.30
ench_oral	**0.52**	0.40	**0.45**	0.15	**1.00**	0.26	0.20	0.79	0.32
chen_oral	**0.30**	0.49	**0.37**	0.12	**0.97**	0.22	0.14	0.73	0.24

TABLE II: Results of over-translation detection on the evaluation datasets

	proposed			std-dict			word-align		
	P	R	F	P	R	F	P	R	F
ench_news	**0.38**	0.75	**0.50**	0.13	0.50	0.20	0.24	**1.00**	0.38
chen_news	**0.73**	**1.00**	**0.84**	0.13	**1.00**	0.23	0.40	**1.00**	0.57
ench_oral	**0.33**	**1.00**	**0.50**	0.00	0.00	0.00	0.14	**1.00**	0.25
chen_oral	**0.80**	0.80	**0.80**	0.38	**1.00**	0.56	0.28	**1.00**	0.43

the Internet. Each dataset contains original sentences randomly sampled from larger datasets (crawled online and different from training datasets), translations under inspection (that contain translation failures), and labels manually added by us indicating the existence of the two types of violations.

Tables I and II show the result summary of under-translation detection and over-translation detection, respectively. The precision, recall, and F-measure are abbreviated as ‘P’, ‘R’, and ‘F’, respectively, in both tables. As shown in Tables I and II, our technique achieves the highest F-measures on all the datasets, compared with the two baseline techniques (‘std-dict’ and ‘word-align’).

IV. Deployment Experience

By using our technique for in-vivo testing in the production environment, developers are able to collect translation tasks resulting in translation failures and observe the performance of deployed models in real-world usages. Developers are also able to handle the translation failures instantly through switching to backup models (e.g., SMT models), improving the overall translation quality with little effort. On top of that, our technique is able to process about **12 million** unique translation tasks every day, with over 200 translations processed each second during the busiest periods. This result indicates the good performance and applicability of our technique on real-world tasks. Our technique also helps enhance in-house black-box system testing during development of the NMT system. By using our technique, the developers manage to find the system outputs (i.e., the translations) that contain translation failures but are previously undetected by test oracles based on translation quality scores (due to missing reference translations). Figure 1 shows the combined statistics of under- and over-translation violations in real-world English and Chinese translations by our NMT model within 6 months after the deployment of our technique. The figure shows that the percentage of translations with under- or over-translation violations decreases significantly over time. These statistics reflect the effectiveness of our technique in helping development and improvement of a machine translation system.

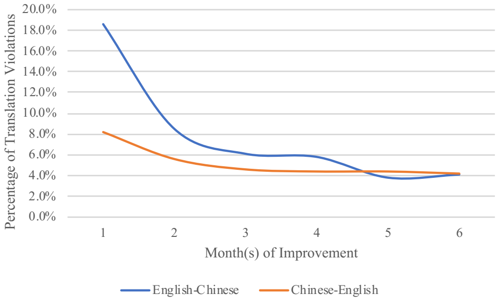

Fig. 1: Combined statistics of under- and over-translation violations in real-world English and Chinese translations by our NMT model

Additionally, our technique helps build an in-house test set containing **130,000** English and **180,000** Chinese words/phrases. These words/phrases serve as in-house test cases for testing and improving WeChat’s continuously-improved machine translation model. These words/phrases help diagnose issues in not only WeChat’s NMT system but also competing machine translation systems released by other providers. By analyzing the translation failures on the in-house test cases, we can gain insights on their potential causes, including defects in the design, implementation, or training data of the machine translation system producing these failures. All these preceding results demonstrate high effectiveness of our technique along with high industry impact. We also build an online demonstration[1] for our technique on English and Chinese translation tasks.

Acknowledgment

The authors from Illinois CS were supported in part by NSF under grants no. CNS-1513939, CNS-1564274, CCF-1816615, and by a 3M Foundation Fellowship.

References

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[2] A. Okrent. (2016) 9 little translation mistakes that caused big problems. [Online]. Available: http://mentalfloss.com/article/48795/9-little-translation-mistakes-caused-big-problems

[3] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proc. 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 311–318.

[4] C. Murphy, G. Kaiser, I. Vo, and M. Chu, “Quality assurance of software applications using the in vivo testing approach,” in *Proc. International Conference on Software Testing Verification and Validation*, 2009, pp. 111–120.

[5] K. Pei, Y. Cao, J. Yang, and S. Jana, “DeepXplore: Automated whitebox testing of deep learning systems,” in *Proc. 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. 10th International Conference on World Wide Web*, 2001, pp. 285–295.

[7] S. Vogel, H. Ney, and C. Tillmann, “HMM-based word alignment in statistical translation,” in *Proc. 16th Conference on Computational Linguistics-Volume 2*, 1996, pp. 836–841.

[8] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, “The mathematics of statistical machine translation: Parameter estimation,” *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[1] https://bit.ly/2P4hEB4