Machine Translation Testing via Pathological Invariance

Shashij Gupta
Department of Computer Science and Engineering
IIT Bombay, India
shashijgupta@cse.iitb.ac.in

ABSTRACT

Due to the rapid development of deep neural networks, in recent years, machine translation software has been widely adopted in people's daily lives, such as communicating with foreigners or understanding political news from the neighbouring countries. However, machine translation software could return incorrect translations because of the complexity of the underlying network. To address this problem, we introduce a novel methodology called *PaInv* for validating machine translation software. Our key insight is that sentences of different meanings should not have the same translation (*i.e.*, *pa*thological *inv*ariance). Specifically, PaInv generates syntactically similar but semantically different sentences by replacing one word in the sentence and filter out unsuitable sentences based on both syntactic and semantic information. We have applied PaInv to Google Translate using 200 English sentences as input with three language settings: English→Hindi, English→Chinese, and English→German. PaInv can accurately find 331 pathological invariants in total, revealing more than 100 translation errors.

ACM Reference Format:
Shashij Gupta. 2020. Machine Translation Testing via Pathological Invariance. In *42nd International Conference on Software Engineering Companion (ICSE '20 Companion), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3377812.3382162

1 INTRODUCTION

Over the decade the use of machine translation software (e.g. Google Translate[1]) has greatly increased. For example, In 2016, Google translate had 500 million users and translated more than 100 billion words per day [17]. However, modern machine translation systems are not as reliable as one might think. In recent years, incorrect translations from these systems have lead to serious and harmful consequences in real-world settings, such as financial loss, social issues, defaming, and threats to personal safety [6, 12, 14, 15]. To tackle this issue, this paper proposes a novel testing methodology, namely PaInv, based on ***Pa**thological **Inv**ariance*: sentences of different meanings have identical translations. Specifically, PaInv generates sentence of different meanings by replacing one word in a sentence with a non-synonymous word. We provide a practical implementation of PaInv by adapting BERT [7] to generate candidate words and using WordsAPI [1] and NLTK [3] to filter out synonyms. To evaluate the effectiveness of PaInv, we use it to test Google Translate on 200 real-world English sentences. PaInv successfully reports 331 pathological invariants with 45.3% precision. With a tunable parameter (Section 3), PaInv can report 10 pathological invariants with 100% precision. All the reported pathological invariants have been released[2] for independent validation.

2 RELATED WORK

Adversarial machine learning aims at fooling machine translation models with malicious input. Most of the existing adversarial techniques are white-box [5, 13, 19], which require knowledge of network structure and parameters. Different from them, PaInv is black-box. Existing black-box techniques [2, 9, 11] rely on perturbations or paraphrasing that easily lead to invalid sentences (e.g., syntax errors or misspellings). Differently, the erroneous sentences reported by PaInv do not contain lexical or syntax errors.

Machine translation testing aims at automatically finding lexically and syntactically correct sentences that trigger translation errors. [18, 20] proposed two models to detect under-translation and over-translation errors respectively, while PaInv targets general errors. He et al. [10] and Sun et al. [16] develop metamorphic testing techniques for general translation errors based on the assumption that similar sentences should have similar translations (evaluated by sentence structures [10] or four existing distance metrics [16]). Differently, PaInv is based on pathological invariance. Thus, we believe PaInv can complement these approaches.

3 APPROACH AND IMPLEMENTATION

The input to PaInv is a list of sentences in source language (e.g., English), while the output is a list of suspicious sentence pairs and their translations. In particular, a sentence pair contains sentences of different meanings but identical translation returned by the machine translation system under test. Thus, at least one of the sentences is translated incorrectly. PaInv consists of four main steps as illustrated in Figure 1.

Generating syntactically-similar sentences. To obtain sentences of different meaning but with identical translation in a black-box manner, for an original sentence, PaInv generates a list of syntactically-similar sentences by replacing one word in the sentence. We mask a word in the sentence and use the remaining words in the sentences to generate a list of suitable words with BERT [8], a state-of-the-art masked language model. In particular, except for stopwords (i.e., NLTK stopwords [3]), we replace each noun, verb, adverb, adjective,

[1] https://translate.google.com/

[2] https://github.com/shashijgupta/PathologicalInvarianceTesting

The work was performed during my internship at ETH Zurich in collaboration with Dr. Pinjia He and Prof. Zhendong Su.

ICSE '20 Companion, May 23–29, 2020, Seoul, Republic of Korea
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7122-3/20/05…$15.00
https://doi.org/10.1145/3377812.3382162

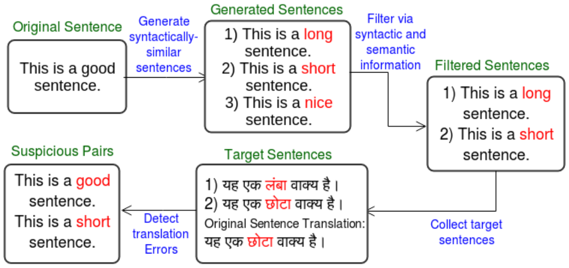
Original Sentence
This is a good sentence.
Generate syntactically-similar sentences
Generated Sentences
1) This is a long sentence.
2) This is a short sentence.
3) This is a nice sentence.
Filter via syntactic and semantic information
Filtered Sentences
1) This is a long sentence.
2) This is a short sentence.
Collect target sentences
Target Sentences
1) यह एक लंबा वाक्य है।
2) यह एक छोटा वाक्य है।
Original Sentence Translation:
यह एक छोटा वाक्य है।
Detect translation Errors
Suspicious Pairs
This is a good sentence.
This is a short sentence.

Figure 1: Overview of our approach (English→Hindi)

and possessive pronoun one at a time. For each word position, we generate 50 sentences using the top-50 words returned by BERT.

Filtering via syntactic and semantic information. We intend to generate sentences of different meanings. However, due to the limitation of the mask language model, the generated sentence could be semantically-similar to the original sentence and thus leads to false positives. To address this problem, in this step, we provide three filtering mechanisms. (1) *Filtering by synonyms.* If a sentence is generated by replacing a word in the original sentence with its synonym (e.g., "*good* talk" and "*nice* talk"), it is likely that the two sentences have correct identical translations. Thus, for each word returned by BERT, we check whether it is synonymous to the replaced word by WordsAPI [1], an industrial and paid natural language service, and filter out the synonyms accordingly. Before filtering, we conduct stemming (e.g., "waits"→"wait") and lemmatization (e.g., "ate"→"eat") via NLTK libraries. (2) *Filtering by constituency structure.* We filter out a sentence if its constituency structure (obtained via constituency parser [21]) is different from that of the original sentence because this kind of sentences often are syntactically-wrong (e.g., replacing a verb with a noun). (3) *Filtering by sentence embeddings.* We further use Universal Sentence Encoder [4] to calculate semantic similarity between two sentences. PaInv filters out a sentence if its similarity to the original sentence is larger than a pre-defined threshold (e.g., "He never took himself too seriously" and "He never treated himself too seriously"). All the three filtering mechanisms are necessary because they target different kinds of false positives (*i.e.*, synonyms, syntax errors, sentences with identical meaning).

Collecting target sentences. We feed the original and the generated sentences to the machine translation system under test and collect their target sentences.

Detecting translation errors. If the translation of a generated sentence is identical to the translation of the original sentence, PaInv will report the generated sentence, the original sentence, and their translations as a suspicious pair.

4 EVALUATION

The goal of PaInv is to find pathological invariants, which are suspicious pairs of sentences that have real translation errors. With a tunable similarity threshold in the "filtering by sentence embedding" step, we can trade-off between (1) how accurate PaInv is on reporting pathological invariants; and (2) how many pathological invariants can PaInv find. Thus, PaInv is evaluated by (1) *Precision*: the ratio of pathological invariants among all the reported pairs; (2) *Recall*: the ratio of reported pathological invariants among all pathological invariants without filtering by sentence embeddings; and (3) *F1-score*: the harmonic mean of precision and recall.

Lang. setting	TP	FP	TN	FN	Precision	Recall	F1
Eng-Ch	103	69	90	33	0.6	0.76	0.67
Eng-Hi	76	71	0	0	0.52	1.0	0.69
Eng-Ge	118	171	0	0	0.41	1.0	0.58

Table 1: Precision, Recall and F1-score of PaInv.

To evaluate the effectiveness of PaInv, we apply it to Google Translate with 200 real-world English sentences collected by [10] (100 in "politics" dataset and 100 in "business" dataset). We manually check all the reported suspicious pairs. In particular, we label a pair as a pathological invariant if (1) the sentences are lexically and syntactically correct; (2) the source sentences have different meanings; and (3) at least one translation contain error(s). The results are presented in Table 1. It shows the best F1-score achieved for each language pair by tuning the threshold parameter. We report True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) after filtering for each language setting. The results show that PaInv achieves very high recall and decent precision for all the language settings.

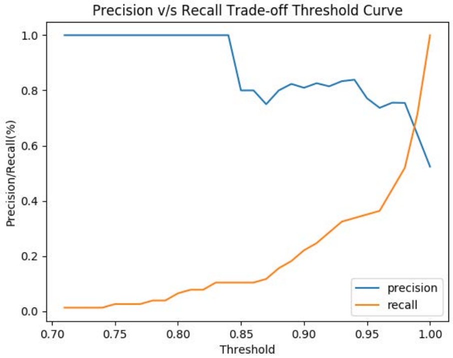

Figure 2: Precision v/s recall trade-off Curve under different thresholds (English→Hindi)

In addition, Figure 2 shows the precision and recall under different similarity thresholds. Developers could tune the similarity threshold according to real-world needs. For example, if developers want PaInv to be as precise as possible, they could use a relatively small threshold (e.g., 0.75) to achieve high precision (e.g., 100%) with the cost of fewer reported pairs. In the following, we present two pathological invariants found by PaInv from Google Translate. The first one is for English→Hindi and second one is for both English→Chinese and English→German:

I had a story to tell and I wanted to finish it, Draper says.
I had a story to tell and I wanted to finish it, Kane says.

They are doing something completely different.
They are doing anything completely different.

Thus, PaInv is effective in finding real-world translation errors and complement existing approaches. In addition, the general concept of pathological invariance could be adapted to various scenarios, such as speech recognition, image captioning, etc.

REFERENCES

[1] [n. d.]. WordsAPI. https://www.wordsapi.com/

[2] Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[3] Edward Loper Bird, Steven and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

[4] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *arXiv e-prints* (2018).

[5] Akshay Chaturvedi, Abijith KP, and Utpal Garain. 2019. Exploring the Robustness of NMT Systems to Nonsensical Inputs. *arXiv preprint arXiv:1908.01165* (2019).

[6] Gareth Davies. 2017. Palestinian man is arrested by police after posting 'Good morning' in Arabic on Facebook which was wrongly translated as 'attack them'. https://www.dailymail.co.uk/news/article-5005489/Good-morning-Facebook-post-leads-arrest-Palestinian.html

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints* (2018).

[9] Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On Adversarial Examples for Character-Level Neural Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*.

[10] Pinjia He, Clara Meister, and Zhendong Su. 2020. Structure-Invariant Testing for Machine Translation. In *Proc. of the 42nd International Conference on Software Engineering (ICSE)*.

[11] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

[12] Fiona Macdonald. 2015. The Greatest Mistranslations Ever. http://www.bbc.com/culture/story/20150202-the-greatest-mistranslations-ever

[13] Paul Michel, Xian Li, Graham Neubig, and Juan Miguel Pino. 2019. On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

[14] Arika Okrent. 2016. 9 Little Translation Mistakes That Caused Big Problems. http://mentalfloss.com/article/48795/9-little-translation-mistakes-caused-big-problems

[15] Thuy Ong. 2017. Facebook apologizes after wrong translation sees Palestinian man arrested for posting 'good morning'. https://www.theverge.com/us-world/2017/10/24/16533496/facebook-apology-wrong-translation-palestinian-arrested-post-good-morning

[16] Zeyu Sun, Jie M Zhang, Mark Harman, Mike Papadakis, and Lu Zhang. 2020. Automatic Testing and Improvement of Machine Translation. In *Proc. of the 42nd International Conference on Software Engineering (ICSE)*.

[17] Barak Turovsky. 2016. Ten years of Google Translate. https://blog.google/products/translate/ten-years-of-google-translate/

[18] Wenyu Wang, Wujie Zheng, Dian Liu, Changrong Zhang, Qinsong Zeng, Yuetang Deng, Wei Yang, Pinjia He, and Tao Xie. 2019. Detecting Failures of Neural Machine Translation in the Absence of Reference Translations. In *Proc. of the 49th IEEE/IFIP International Conference on Dependable Systems and Networks (industry track)*.

[19] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[20] Wujie Zheng, Wenyu Wang, Dian Liu, Changrong Zhang, Qinsong Zeng, Yuetang Deng, Wei Yang, Pinjia He, and Tao Xie. 2019. Testing untestable neural machine translation: an industrial case. In *Proc. of the 41st International Conference on Software Engineering: Companion Proceedings*.

[21] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and Accurate Shift-Reduce Constituent Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 434–443.