Automatic Improvement of Machine Translation Using Mutamorphic Relation (Invited Talk Paper)

Jie M. Zhang*
University College London
jie.zhang@ucl.ac.uk

ABSTRACT

This paper introduces Mutamorphic Relation for Machine Learning Testing. Mutamorphic Relation combines data mutation and metamorphic relations as test oracles for machine learning systems. These oracles can help achieve fully automatic testing as well as automatic repair of the machine learning models.

The paper takes TransRepair as an example to show the effectiveness of Mutamorphic Relation in automatically testing and improving machine translators, TransRepair detects inconsistency bugs without access to human oracles. It then adopts probability-reference or cross-reference to post-process the translations, in a grey-box or black-box manner, to repair the inconsistencies. Manual inspection indicates that the translations repaired by TransRepair improve consistency in 87% of cases (degrading it in 2%), and that the repairs of have better translation acceptability in 27% of the cases (worse in 8%).

KEYWORDS

mutamorphic relation, mutation testing, metamorphic testing

ACM Reference Format:
Jie M. Zhang. 2020. Automatic Improvement of Machine Translation Using Mutamorphic Relation (Invited Talk Paper). In *42nd International Conference on Software Engineering (ICSE '20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3387940.3391541

1 MUTAMORPHIC RELATION

Mutamorphic relation is a type of metamorphic relation with mutated data inputs in machine learning systems.

Let S be a test data set. Let r be a mutation degree (ratio of mutated instances), S_r is a mutated data set constructed by changing r proportion of the test data instances. Let h be a machine learning model, h takes S as inputs and produce outputs $h(S)$.

Definition 1.1 (Mutamorphic Relation). Mutamorphic Relation is the relationship between test input changes and test output changes. The input changes are conducted via data mutation:

$$R(S, S_r) \rightarrow R(h(S), h(S_r)) \quad (1)$$

*Brief 2-page paper to accompany Jie M. Zhang's MET 2020 invited talk. More details about the introduced work can be found at [10].

ICSEW '20, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7963-2/20/05…$15.00
https://doi.org/10.1145/3387940.3391541

Mutamorphic relation has been adopted in many scenarios in machine learning testing [17]. In machine learning testing, test oracle is an obstacle in automatic testing and repair. Mutamorphic relation provides support in such need of fully automatic process.

This talk is going to take TransRepair as a show case of the effectiveness of mutamorphic relation in testing and improving machine translators.

2 MACHINE TRANSLATOR IMPROVEMENT

Machine learning has been successful in providing general-purpose natural language translation systems, with many systems able to translate between thousands of pairs of languages effectively in real time [4].

2.1 Motivation

Machine translation systems are not perfect and the bugs that users experience have a different character from those on traditional, non-machine learning-based, software systems [1, 6, 7, 17].

The consequences of mistranslation through machine-based translators have been shown to be serious. For example, machine translations have been shown to exhibit pernicious fairness bugs that disproportionately harm specific user constituencies [9]. We have found such examples of fairness bugs in widely used industrial strength translation systems. Figure 1 shows several such Google Translate results for the language pair (English → Chinese)[1]. As can be seen from the figure, Google Translate translates 'good' into 'hen hao de' (which means 'very good') when the subject is 'men' or 'male students'. However, interestingly, but also sadly, it translates 'good' into 'hen duo' (which means 'a lot') when the subject is 'women' or 'female students'[2].

Such inconsistency may confuse users, and is also clearly *unfair* to female researchers in computer science; producing 'a lot' of research is clearly a more pejorative interpretation, when compared to producing 'very good' research. To avoid such unfair translations (at scale), we need techniques that can automatically identify and remedy such inconsistencies.

2.2 Automatic Test Generation

In order to tackle the testing problem, we use mutamorphic relation – an approach that combines mutation [5, 8, 15, 16] with metamorphic testing [2, 14]. The approach conducts context-similar mutation to generate mutated sentences that can be used as test

[1]The four translations were obtained on 23rd July, 2019. These examples are purely for illustration purposes, and are not intended as a criticism of Google Translate. It is likely that other mainstream translation technologies will have similar issues.
[2]Similar issues also exist in translations between other languages. With a cursory check, we already found a case with German→Chinese.

English	Chinese (Google Translation)	Notes
Men do good research in computer science.	Nanren zai jisuanji kexue fangmian zuole hen hao de yanjiu 男人在计算机科学方面做了很好的研究	good → hen hao de **(very good)**
Women do good research in computer science.	Nüxing zai jisuanji kexue fangmian zuole henduo yanjiu 女性在计算机科学方面做了很多研究	good → henduo **(a lot)**
Male students do good research in computer science.	Nan xuesheng zai jisuanji kexue fangmian zuole hen hao de yanjiu 男学生在计算机科学方面做了很好的研究	good → hen hao de **(very good)**
Female students do good research in computer science.	Nü xuesheng zai jisuanji kexue fangmian zuole henduo yanjiu 女学生在计算机科学方面做了很多研究	good → henduo **(a lot)**

Figure 1: Examples of fairness issues brought by translation inconsistency (from Google Translate)

inputs for the translator under test. When a context-similar mutation yields above-threshold disruption to the translation of the non-mutated part, the approach reports an inconsistency bug.

2.3 Automatic Translation Repair

Traditional approaches to 'repairing' machine learning systems typically use data augmentation or algorithm optimisation. These approaches can best be characterised as to "improve" the overall effectiveness of the machine learner, rather than specific repairs for individual bugs; they also need data collection/labelling and model retraining, which usually have a high cost.

Traditional approaches to 'repairing' software bugs are white box, because the techniques need to identify the line(s) of source code that need(s) to be modified in order to implement a fix. Such approaches inherently cannot be applied to fix software for which source code is unavailable, such as third-party code.

Our insight is that by combining the results of repeated (and potentially inconsistent) output from a system, we can implement a light-weight *black-box* repair technique as a kind of 'post-processing' phase that targets specific bugs. Our approach is the first repair technique to repair a system in a purely black-box manner. We believe that black-box repair has considerable potential benefits, beyond the specific application of machine translation repair. It is the only available approach when the software engineer is presented with bugs in systems for which no source code is available.

2.4 Evaluation Results

TransRepair is evaluated on two state-of-the-art machine translation systems, Google Translate and Transformer [11]. In particular, we focus on the translation between the top-two most widely-spoken languages: English and Chinese. These languages each have over one billion speakers worldwide [3]. Nevertheless, only 10 million people in China (less than 1% of the population) are able to communicate via English [12, 13]. Since so few people are able to speak both languages, machine translation is often attractive and sometimes necessary and unavoidable.

Our results indicate that TransRepair generates valid test inputs effectively with a precision of 99%; 2) TransRepair automatically reports inconsistency bugs effectively with the learnt thresholds, with a mean F-measure of 0.82/0.88 for Google Translate/Transformer; 3) Both Google Translate and Transformer have inconsistency bugs. Automated consistency metrics and manual inspection reveal that Google Translate has approximately 36% inconsistent translations on our generated test inputs. 4) Black-box repair reduces 28% and 19% of the bugs of Google Translate and Transformer. Grey-box reduces 30% of the Transformer bugs. Manual inspection indicates that the repaired translations improve consistency in 87% of the cases (reducing it in only 2%), and have better translation acceptability in 27% of the cases (worse in only 8%)

ACKNOWLEDGEMENT

Jie M. Zhang is supported by the ERC advanced grant with No. 741278.

REFERENCES

[1] Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proc. ICLR.*
[2] Tsong Y Chen, Shing C Cheung, and Shiu Ming Yiu. 1998. *Metamorphic testing: a new approach for generating next test cases.* Technical Report.
[3] David M Eberhard, Gary F Simons, and Charles D Fennig. 2019. Ethnologue: Languages of the world. (2019).
[4] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, James Law, Kevin Lee, Jason Lu, Pieter Noordhuis, Misha Smelyanskiy, Liang Xiong, and Xiaodong Wang. 2018. Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective. In *24th International Symposium on High-Performance Computer Architecture (HPCA 2018), February 24-28, Vienna, Austria.*
[5] Yue Jia and Mark Harman. 2011. An Analysis and Survey of the Development of Mutation Testing. *IEEE Transactions on Software Engineering* 37, 5 (September–October 2011), 649 – 678.
[6] Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on Synthetic Noise Improves Robustness to Natural Noise in Machine Translation. *arXiv preprint arXiv:1902.01509* (2019).
[7] Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282* (2018).
[8] Mike Papadakis, Marinos Kintis, Jie Zhang, Yue Jia, Yves Le Traon, and Mark Harman. 2019. Mutation testing advances: an analysis and survey. In *Advances in Computers.* Vol. 112. Elsevier, 275–378.
[9] Parmy Olson. 2018. The Algorithm That Helped Google Translate Become Sexist. https://www.forbes.com/sites/parmyolson/2018/02/15/the-algorithm-that-helped-google-translate-become-sexist/#224101cb7daa.
[10] Zeyu Sun, Jie M. Zhang, Mark Harman, Mike Papadakis, and Lu Zhang. 2020. Automatic Testing and Improvement of Machine Translation. In *Proc. ICSE (to appear).*
[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems.* Curran Associates Inc., 6000–6010.
[12] VoiceBoxer. 2016. WHAT ABOUT ENGLISH IN CHINA? http://voiceboxer.com/english-in-china/.
[13] Rining Wei and Jinzhi Su. 2012. The statistics of English in China: An analysis of the best available data from government sources. *English Today* 28, 3 (2012), 10–14.
[14] Jie Zhang, Junjie Chen, Dan Hao, Yingfei Xiong, Bing Xie, Lu Zhang, and Hong Mei. 2014. Search-based inference of polynomial metamorphic relations. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering.* ACM, 701–712.
[15] Jie Zhang, Lingming Zhang, Mark Harman, Dan Hao, Yue Jia, and Lu Zhang. 2018. Predictive mutation testing. *IEEE Transactions on Software Engineering* 45, 9 (2018), 898–918.
[16] Jie Zhang, Muyao Zhu, Dan Hao, and Lu Zhang. 2014. An empirical study on the scalability of selective mutation testing. In *2014 IEEE 25th International Symposium on Software Reliability Engineering.* IEEE, 277–287.
[17] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. 2019. Machine Learning Testing: Survey, Landscapes and Horizons. *arXiv preprint arXiv:1906.10742* (2019).