2019

The impact of a dot: Case studies of a noise metamorphic relation pattern

Chaohua Wu
University of Wollongong, cw811@uowmail.edu.au

Liqun Sun
University of Wollongong, ls168@uowmail.edu.au

Zhi Q. Zhou
University of Wollongong, zhiquan@uow.edu.au

Abstract

We propose a 'noise' metamorphic relation pattern (MRP), which is a sub-pattern under the more general MRP 'symmetry.' We conduct case studies with real-life systems in three different application domains (obstacle perception in autonomous systems, machine translation, and named entity recognition) to show the usefulness of the 'noise' MRP for software verification and validation.

The Impact of a Dot: Case Studies of a Noise Metamorphic Relation Pattern

Chaohua Wu, Liqun Sun, and Zhi Quan Zhou*
School of Computing and Information Technology
University of Wollongong
Australia

***Abstract*—We propose a "noise" metamorphic relation pattern (MRP), which is a sub-pattern under the more general MRP "symmetry." We conduct case studies with real-life systems in three different application domains (obstacle perception in autonomous systems, machine translation, and named entity recognition) to show the usefulness of the "noise" MRP for software verification and validation.**

Keywords: Metamorphic testing, metamorphic relation pattern, oracle problem, noise, machine translation, machine recognition, artificial intelligence.

I. Introduction

Testing is the most widely used approach for software verification and validation. A key component of testing is the mechanism to determine whether the outcomes of test case executions are correct. Such a mechanism is called a *test oracle*. Sometimes, however, a test oracle is unavailable or is too expensive to be applied—a situation known as the *oracle problem* [1], [2]. For example, due to the sheer volume of data, software for big data analytics is difficult to test [3].

A growing body of research has examined the concept of *metamorphic testing* (MT) [4], [5], and proven it highly effective for addressing the oracle problem and automated test case generation problem [1], [6], [7], [8]. MT was originally proposed as a *verification* technique, which can be adopted by both development organizations [9] and end-user programmers [10]. Xie et al. [11] found that MT could also be used for *software validation*, and Zhou et al. [12] further developed MT into a unified framework for software verification, validation, and other types of quality assessment.

In MT, the *software under test* (SUT) is checked against prescribed *metamorphic relations* (MRs). MRs are expected relations among the inputs and outputs of *multiple* executions of the SUT [7]. Because MRs are necessary properties of the software's intended functionality, if an MR is violated for certain test cases during testing then the SUT must be at fault.

To facilitate systematic identification of useful MRs, a concept of metamorphic relation "patterns" has been proposed from which multiple concrete metamorphic relations can be derived [13], [14]. Zhou et al. [14] defined a metamorphic relation pattern (MRP) as an abstraction that characterizes a set of (possibly infinitely many) metamorphic relations, and they also identified a universal MRP, *symmetry*. In the present research, we propose a *noise* MRP, which is a sub-pattern under *symmetry*, and show its applications using real-life software systems in different domains. The rest of this paper is organized as follows: Section II introduces the concept of metamorphic relation patterns, and proposes a *noise* pattern. Section III revisits previous work from the perspective of the *noise* pattern. Section IV shows the application of the *noise* pattern in the context of machine translation. Section V goes on with a case study in the area of named entity recognition. Section VI concludes the paper.

* All correspondence should be addressed to Dr. Z.Q. Zhou, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia. Email: zhiquan@uow.edu.au

II. Metamorphic Relation Pattern

In the early days of MT research, researchers usually identified MRs from scratch for each individual problem under study. To make this process more systematic, Zhou et al. were the first to propose an idea of using an abstract form of MR to derive multiple concrete MRs, and they called this abstract form of MR a "general metamorphic relation" [15, p. 3]. In a follow-up study, Zhou et al. further identified another type of abstract relation, which is a *subset* relation among the source and follow-up *outputs*—they called this a "general relation" [16, p. 223]. Their empirical results demonstrated that concrete MRs derived from the above abstract forms of MRs had a strong fault-detection capability [15], [16].

Recently, Segura et al. [17] introduced the term *metamorphic relation output pattern* (MROP), which they defined as an abstract relation among the source and follow-up outputs from which multiple concrete metamorphic relations can be derived. Their work opened a new MT research direction on "metamorphic relation patterns," in a broad sense, as foreseen by Segura in his keynote at the third International Workshop on Metamorphic Testing (ICSE MET '18) [13].

All the above studies on abstract forms of MRs, when introduced, were limited to their specific application domains (that is, search functions [15], [16] and RESTful web APIs [17]).

More recently, Zhou et al. [14] further investigated the notion of "patterns" and formally defined the general concept of a *metamorphic relation pattern* (MRP) as "an abstraction that characterizes a set of (possibly infinitely many) metamorphic relations." Zhou et al. also defined a concept of a *metamorphic relation input pattern* (MRIP) as "an abstraction that characterizes the relations among the source and follow-up inputs of a set of (possibly infinitely many) metamorphic relations." After proposing these basic concepts, Zhou et al. [14] identified a universal *symmetry* MRP, which "refers to the existence of different viewpoints from which the system appears the same"—this definition borrows from the notion given by Philip W. Anderson, Nobel laureate in Physics, who said: "**By symmetry we mean the existence of different viewpoints from which the system appears the same**" and that "it is only slightly overstating the case to say that physics is the study of symmetry" [18, p. 394]. In a *symmetry* MRP, the word "system" can refer to not only a physical system, but also to a computer system. The *symmetry* MRP is not limited to any specific application domain, but rather is general enough to be applicable to various areas. Also note that, in the definition of the *symmetry* MRP, "the system appears the same" does not mean that the software system's (source and follow-up) outputs must have an equality or equivalence relation [14].

Using the *symmetry* MRP, and a "change direction" MRIP, Zhou et al. [14] conducted case studies in a variety of different application domains, including commercial websites, navigation software, location-based search, face recognition, and video analysis. The results showed that their patterns can help users to (i) detect previously unknown failures efficiently and effectively, and (ii) obtain more desirable computation results in spite of the failures, even when the users do not fully understand the implementation of the software.

In the present paper, we propose a *noise* MRP, defined as follows:

Definition 1: The *noise MRP* refers to the requirement that a reliable system should be able to perform its functions when a low level of interference (noise) is present.

Remark 1: Definition 1 means that some noise in the input data or environment should not have a strong impact on the program's output if the program is reliable. A tester can therefore test the SUT by first running a normal input, and then running it with some injected noise, and finally comparing the outputs with each other.

Remark 2: To achieve generality, an MRP is defined at a higher level of abstraction than a concrete MR. The above definition, therefore, does not need to include an explanation of the exact meaning of "perform its functions," "a low level of interference," and "noise." These terms can be interpreted in different ways when the MRP is instantiated in specific application domains.

Remark 3: In the literature of metamorphic testing, the concept of "noise" has already been used by different researchers for the development of metamorphic relations in their application areas. The "noise" concept itself, therefore, is not new. Nevertheless, documenting it in the form of an MRP to enhance generality and reusability is beneficial.

Remark 4: As pointed out by Zhou et al. [14], it is possible for many MRPs to form a hierarchy, with MRPs at higher levels being more abstract, and those at lower levels being more concrete. Obviously, the *noise* MRP is a sub-pattern of the *symmetry* MRP, as the latter is more general (abstract).

Remark 5: To study the relationships among different MRPs and to construct family trees for them will be an important future research direction. Researchers in software patterns and pattern languages have developed approaches for structuring and visualizing relationships among patterns, such as *abstract security patterns* [19]. Some of those approaches could be adopted or adapted for MRP research.

III. The Noise MRP for Autonomous Systems

A trend has recently emerged for applying MT to machine learning and autonomous systems [20], [21], [22], [23]. In particular, Zhou and Sun [24] combined MT and fuzzing and detected previously unknown fatal defects in the LiDAR obstacle-perception module of the real-life self-driving system Baidu Apollo. In this section, we revisit Zhou and Sun's work [24] from the "pattern" perspective, and show that the approach used in their work is an application, or instance, of the *noise* MRP.

A. Background

At about 10 pm of March 18, 2018, an autonomous Uber SUV hit Elaine Herzberg in the street of Tempe, Arizona. The death of Herzberg was the first recorded case of a pedestrian fatality involving a self-driving vehicle. Subsequently, experts expressed doubts about Uber's LiDAR technology [25]. LiDAR stands for "Light Detection and Ranging," which enables an autonomous vehicle to see its surroundings hundreds of feet away. The LiDAR supplier, Velodyne, said that "our LiDAR is capable of clearly imaging Elaine and her bicycle in this situation. However, our LiDAR doesn't make the decision to put on the brakes or get out of her way" [26], and that "our LiDAR can see perfectly well in the dark, as well as it sees in daylight, producing millions of points of information. However, it is up to the rest of the system to **interpret and use** the data to make decisions. We do not know how the Uber system of decision-making works" [27].

Before the Uber accident, Zhou and Sun had already started an investigation into the question "Are there situations where a driverless car's on-board computer system could incorrectly interpret and use the data sent from a sensor such as a LiDAR sensor, making the car unable to detect a pedestrian or an obstacle on the roadway?" They did not have access to the Uber system, but managed to test Baidu Apollo, a famous real-life self-driving software system controlling many autonomous vehicles on the road (http://apollo.auto). Using a combination of metamorphic testing and fuzzing, Zhou and Sun found a fatal software fault in Apollo's LiDAR Obstacle Perception (LOP) module (which takes as input the 3D *point cloud* data generated by Velodyne's HDL64E LiDAR sensor, exactly the same type of LiDAR involved in the Uber accident [28]). The fault could make the system unable to detect some obstacles. Zhou and Sun reported this issue to the Baidu Apollo self-driving car team on March 10, 2018, MST (UTC -7), *eight days before the Uber accident*. They did not receive a response until 10:25 pm, March 19, 2018, MST (24 hours after the Uber accident), in which the Apollo *perception team* confirmed the error [24].

B. Testing Method: A Concrete Instance of the Noise MRP

Zhou and Sun [24] identified the following MR, where the software under test is the LOP module, A and A' represent two inputs to LOP, and O and O' represent the LOP's outputs for A and A', respectively:

> ***MR***

$\boldsymbol{MR_{LiDAR}}$: Let A and A' be two frames of 3D point cloud data that are identical except that A' includes a small number of additional LiDAR data points (which could represent tiny particles in the air or some possible noise from the sensor) randomly scattered in regions outside the driving area. Let O and O' be the sets of obstacles in the driving area identified by LOP for A and A', respectively. Then, the following relation must hold: $O \subseteq O'$.

Remark 1: MR_{LiDAR} means that the existence of some particles in the air, or some noise points, far away outside the driving area should not cause an obstacle inside the driving area to become undetectable. Obviously, MR_{LiDAR} is a concrete instance of the *noise* MRP defined in the present paper.

Remark 2: While we use the LiDAR image as an example to illustrate our approach, the idea of the *noise* MRP is generally applicable to almost all types of sensors and signals including videos, sound, speed, temperature, pressure, positions, angles, and so on.

Remark 3: For mission-critical systems, lack of robustness in dealing with erroneous sensor data could cause catastrophic consequences including aircraft crashes [29].

C. Test Results

Figs. 1a and 1b show a real-life example of Zhou and Sun's findings [24], where a pedestrian inside the driving area (the Apollo system depicted this pedestrian using the small pink mark as shown in Fig. 1a) could not be detected after only 10 random points were placed outside the driving area (as shown in Fig. 1b, the small pink mark is missing). Through a series of experiments with the Apollo system, Zhou and Sun found that the probability of this type of failure (violation of MR_{LiDAR}) was as high as 2.7% when only 10 random points were added [24].

IV. The Noise MRP for Machine Translation

Can the *noise* MRP be applied to domains beyond signal processing? Our answer is affirmative. This section shows such an example in the natural language processing domain. We consider the testing of machine translation.

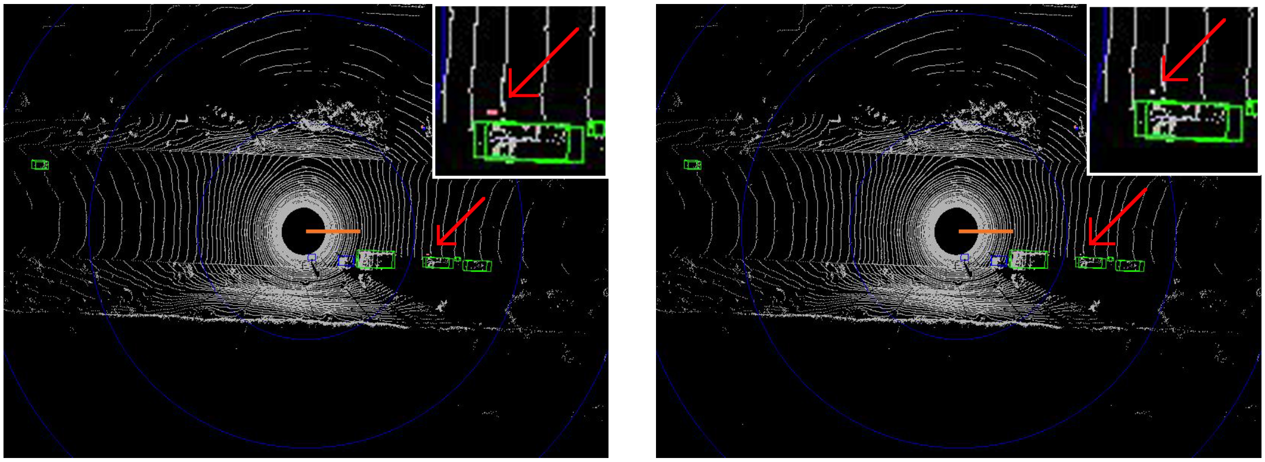
A. Related Work

Generally speaking, manual assessment of machine translation quality by a human assessor is both expensive and subjective [30]. A method that alleviates this problem is known as *round-trip translation* (RTT) [31] (that is, translate the original sentence to the target language and back to the original language, then compare the difference). RTT does not test one system, but two systems: the forward translation and the back translation. In spite of this limitation, it was claimed that "RTT is **the only** technique that can be used when no human fluent in the target language or equivalent text is readily available" [32].

Pesu et al. [30] were the first to develop an automatic **non-RTT** technique that can be used to assess the quality of machine translation without the need for an equivalent target language text, or proficient (fluent) target language user. Their approach used a Monte Carlo method and was based on metamorphic testing. Sun and Zhou [33] extended the study of *metamorphic testing for machine translation* (MT4MT) beyond Monte Carlo approaches. They named their metamorphic relation pattern $MR_{replace}$ (which belongs to the *symmetry* MRP). As an example of detected failures with $MR_{replace}$, they observed that Google translated "Emma likes Mini" into the correct Chinese sentence "艾玛喜欢迷你," but translated "Victoria likes Mini" into "维多利亚喜欢Mini" where "Mini" was not translated into Chinese.

B. Our Findings with the "Replace" and "Noise" MRs

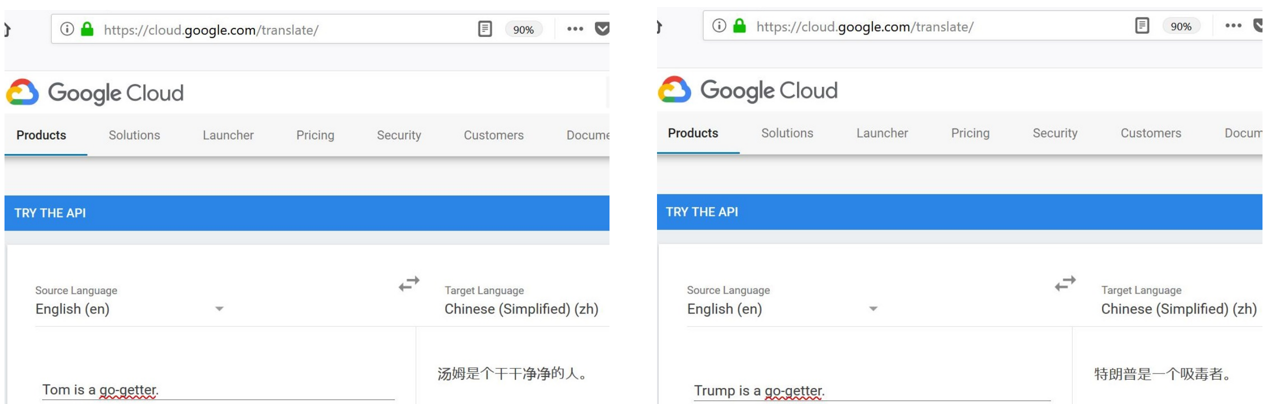
In this section, we first apply $MR_{replace}$ to the Google Cloud Translation API (https://cloud.google.com/translate) to show a translation failure, and then go on to apply a *noise* MR to show further failures. Although the theme of this paper is on the *noise* MRP, we include $MR_{replace}$ in our experiment to show that *multiple* MRs can be applied



(a) Original (104,251 LiDAR data points; the small pink mark was generated by the Apollo system to depict a detected pedestrian).

(b) After adding only 10 random data points outside the driving area, the pedestrian inside the driving area could no longer be detected.

Figure 1. A real-life fatal error in LiDAR point-cloud data interpretation in the Apollo *perception* module: a missing pedestrian. The black circle in the middle of each subfigure is the location of the autonomous car, which was driving towards the right. The boxes in different colors were generated by the Apollo system to depict different types of detected obstacles. This figure was taken from Zhou and Sun [24].



(a) Google translated “Tom is a go-getter.” into a Chinese sentence “汤姆是个干干净净的人。” (which means “Tom is a clean person.”)

(b) Google translated “Trump is a go-getter.” into a Chinese sentence “特朗普是一个吸毒者。” (which means “Trump is a drug addict.”)

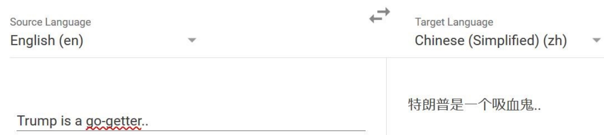
Figure 2. Google Cloud Translation API failure detected by $MR_{replace}$.

in practical situations, and that they may complement each other for the generation of more informative test results.

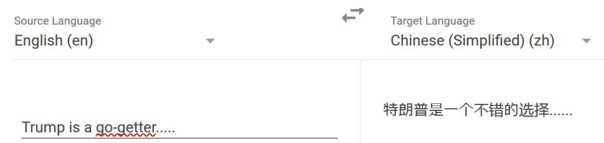
Fig. 2 shows that Google translated “Tom is a go-getter.” and “Trump is a go-getter.” into the Chinese sentences “汤姆是个干干净净的人。” (which means “Tom is a clean person.”) and “特朗普是一个吸毒者。” (which means “Trump is a drug addict.”) respectively. This inconsistency was detected when we run our automated test driver that implemented $MR_{replace}$ with random test case generation. It was illogical that the change of a personal name from “Tom” to “Trump” could have changed the meaning of the entire translation.

Based on the above results, we further defined a *noise* MR, hoping to detect more failures. In this MR, the “noise” was some periods that appear at the end of a sentence. Fig. 3 shows that Google translated “Trump is a go-getter..” (two periods) and “Trump is a go-getter.....” (five periods) into the Chinese sentences “特朗普是一个吸血鬼..” (which means “Trump is a vampire”) and “特朗普是一个不错的选择.....” (which means “Trump is a good choice”) respectively.

The above example shows that, every time the sentence “Trump is a go-getter” was translated, it was given a completely different meaning, only because of the different number of periods in the original sentence: When there was one period (Fig. 2b), Trump was “a drug addict”; when there were two periods (Fig. 3a), Trump was “a vampire”; when there were five periods (Fig. 3b), Trump became “a good choice.” These translation inconsistencies (failures) were repeatable for a long period of time when we conducted the experiment in 2018, and have now been corrected.



(a) Google translated “Trump is a go-getter..” (two periods at the end of the sentence) into a Chinese sentence “特朗普是一个吸血鬼..” (which means “Trump is a vampire.”)



(b) Google translated “Trump is a go-getter.....” (five periods at the end of the sentence) into a Chinese sentence “特朗普是一个不错的选择......” (which means “Trump is a good choice.”)

Figure 3. A further Google Cloud Translation API failure detected by a *noise* MR where the “noise” was defined as periods added to the end of a sentence.

V. The Noise MRP for Named Entity Recognition

In this section, we conduct a case study of the *named entity recognition* (NER) feature of LingPipe, which is a tool kit for processing text using computational linguistics (http://alias-i.com/lingpipe).

NER is the process of finding mentions of specified things in text. For instance, in the sentence *John J. Smith lives in Seattle*, a named entity recognizer might find the person mention *John J. Smith* and the location mention *Seattle* (http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html). As explained in its website, the NER feature of LingPipe “involves the supervised training of a statistical model or more direct methods like dictionary matching or regular expression matching. All these methods are designed to work together smoothly.” This tool is often used to identify biomedical entities (such as genes, organisms, malignancies, chemicals, and so on).

While NER can perform both the *first-best* and the *n-best* named entity chunking, we decided to test the former because the latter always produced a large amount of complicated output that was time-consuming to comprehend. For example, using the *n-best* analysis, a simple text input “How are you today” could yield an analysis report of more than 34 lines.

Metamorphic testing of the LingPipe NER tool was previously studied in [34]; however, we decided not to adopt the MRs used in [34] because those MRs may not be valid when the test data is arbitrary text. Instead, we used the *replace* and the *noise* MRPs to explore this system. As explained earlier, although the focus of this paper is on the *noise* MRP, we included the *replace* MRP ($MR_{replace}$) in the experiment to show that multiple MRs can be applied together to generate more informative test results in practical situations.

We defined the “noise” to be a period added to the end of a sentence or word. Fig. 4 (line 1) shows that, when the input text message was “bbagrm” the LingPipe tool successfully identified this string as a biomedical entity. In Fig. 4 (line 1), “bbagrm” is the input text, “0-6” is the software output that indicates the starting and ending positions of the identified entity. After a period was added to the string (line 2), however, the system failed to identify

```
bbagrm : [0-6:GENE@-Infinity]

bbagrm. : []

Okazaki Fragment : [0-16:GENE@-Infinity]

Okazaki Fragment. : []
```

Figure 4. Anomaly detected by a *noise* MR (where a full stop “.” was used as the noise): The LingPipe tool successfully identified *bbagrm* (line 1) as a biomedical entity but failed to do so when the string was followed by a full stop (line 2). Likewise, the tool successfully identified *Okazaki Fragment* as a biomedical entity (line 3) but failed to do so when the phrase was followed by a full stop (line 4).

any biomedical entity (as represented by the empty symbol “[ ]”). Likewise, lines 3 and 4 show that the phrase “Okazaki Fragment” was identified but, after a full stop was added to the end of the phrase, this entity could no longer be identified.

The anomaly described above may not necessarily indicate a bug in the software, because the addition of a period to the string might have changed the confidence level calculated by the LingPipe NER tool. Nevertheless, if we consider the word identification task from a user’s perspective, these inconsistencies are obviously unacceptable because a full stop is a normal and integral part of a sentence and should not have a negative impact on the named entity recognition. From a user *validation* perspective, therefore, the software failed the *noise* test.

Fig. 5 shows another recorded anomaly (detected by $MR_{replace}$). This observation means that both the *noise* and the *replace* MRs are effective for the software under test.

VI. Conclusion and Future Work

We have proposed a *noise* metamorphic relation pattern (MRP), which is a sub-pattern of the *symmetry* MRP. We have conducted case studies in three different domains: LiDAR image analysis for self-driving vehicles, machine translation, and named entity recognition, where all studies were performed with real-life software systems. We have

```
We present data demonstrating the
  extensive purification of two
  dexamethasone-binding proteins,
  corresponding in their characteristics
  to receptors DE-2 and DE-3. :
  [148-152:GENE@-Infinity,
  157-161:GENE@-Infinity]

It is evident that the major protein
  contained in the DE-2 fraction has a
  molecular weight of approximately 45000
  whereas fraction DE-3 is mainly
  composed of a protein with a molecular
  weight of about 90000. :
  [23-36:GENE@-Infinity,
  54-58:GENE@-Infinity]

We present data demonstrating the
  extensive purification of two
  dexamethasone-binding proteins,
  corresponding in their characteristics
  to receptors DE-2 and DE-3 and so on. :
  [148-152:GENE@-Infinity,
  157-161:GENE@-Infinity]
```

Figure 5. The first sentence (upper) was taken from a biochemistry article [35], for which the LingPipe tool identified both "DE-2" and "DE-3" as named entities. The third sentence (lower) was created by inserting " and so on" to the end of the first sentence, for which the LingPipe tool returned the same result (which satisfied $MR_{replace}$). The second sentence (middle) was taken from the same article, for which the LingPipe tool identified "major protein" and "DE-2" (**but missed out "DE-3"**) as named entities. The missing "DE-3" was detected by $MR_{replace}$ which, for the NER systems, states that if a term is identified in one sentence then it should also be identified in another (similar) sentence, especially if both sentences are from the same article.

shown the various issues detected in these systems. Although these case studies were at a relatively small scale, they are useful for illustrating the proposed concept and its potential effectiveness for a wide range of application areas. In future research, we will continue the investigation of the *noise* MRP at a larger scale.

Acknowledgments

This research was supported in part by a linkage grant of the Australian Research Council (Project ID: LP160101691) and an Australian Government Research Training Program scholarship. We would like to thank Morphick Solutions Pty Ltd for supporting this work. We wish to thank Sergio Segura for his discussions and comments on this paper.

References

[1] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, "The oracle problem in software testing: A survey," *IEEE Transactions on Software Engineering*, vol. 41, no. 5, pp. 507–525, 2015.

[2] Z. Q. Zhou, D. Towey, P.-L. Poon, and T. H. Tse, "Introduction to the special issue on test oracles," *Journal of Systems and Software*, vol. 136, pp. 187–187, 2018, Editorial.

[3] C. E. Otero and A. Peter, "Research directions for engineering big data analytics software," *IEEE Intelligent Systems*, vol. 30, no. 1, pp. 13–19, 2015.

[4] T. Y. Chen, S. C. Cheung, and S. M. Yiu, "Metamorphic testing: A new approach for generating next test cases," Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, Tech. Rep. HKUST-CS98-01, 1998.

[5] T. Y. Chen, T. H. Tse, and Z. Q. Zhou, "Fault-based testing without the need of oracles," *Information and Software Technology*, vol. 45, no. 1, pp. 1–9, 2003.

[6] S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Cortés, "A survey on metamorphic testing," *IEEE Transactions on Software Engineering*, vol. 42, no. 9, pp. 805–824, 2016.

[7] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T. H. Tse, and Z. Q. Zhou, "Metamorphic testing: A review of challenges and opportunities," *ACM Computing Surveys*, vol. 51, no. 1, pp. 4:1–4:27, 2018.

[8] S. Segura, D. Towey, Z. Q. Zhou, and T. Y. Chen, "Metamorphic testing: Testing the untestable," *IEEE Software*, DOI: 10.1109/MS.2018.2875968.

[9] Z. Wang, D. Towey, Z. Q. Zhou, and T. Y. Chen, "Metamorphic testing for Adobe Analytics data collection JavaScript library," in *Proceedings of the IEEE/ACM 3rd International Workshop on Metamorphic Testing (MET '18), in conjunction with the 40th International Conference on Software Engineering (ICSE '18)*. ACM, 2018, pp. 34–37.

[10] T. Y. Chen, F.-C. Kuo, and Z. Q. Zhou, "An effective testing method for end-user programmers," in *ACM SIGSOFT Software Engineering Notes 30 (4), Proceedings of the 1st Workshop on End-User Software Engineering (WEUSE I)*. ACM Press, 2005, pp. 1–5.

[11] X. Xie, J. W. K. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen, "Testing and validating machine learning classifiers by metamorphic testing," *Journal of Systems and Software*, vol. 84, pp. 544–558, 2011.

[12] Z. Q. Zhou, S. Xiang, and T. Y. Chen, "Metamorphic testing for software quality assessment: A study of search engines," *IEEE Transactions on Software Engineering*, vol. 42, no. 3, pp. 264–284, 2016.

[13] S. Segura, "Metamorphic testing: Challenges ahead (keynote speech)," in *Proceedings of the IEEE/ACM 3rd International Workshop on Metamorphic Testing (MET '18), in conjunction with the 40th International Conference on Software Engineering (ICSE '18)*, May 27, 2018, pp. 1–1.

[14] Z. Q. Zhou, L. Sun, T. Y. Chen, and D. Towey, "Metamorphic relations for enhancing system understanding and use," *IEEE Transactions on Software Engineering*, in press. [Online]. Available: https://doi.org/10.1109/TSE.2018.2876433

[15] Z. Q. Zhou, T. H. Tse, F.-C. Kuo, and T. Y. Chen, "Automated functional testing of web search engines in the absence of an oracle," Department of Computer Science, The University of Hong Kong, Tech. Rep. TR-2007-06, 2007.

[16] Z. Q. Zhou, S. Zhang, M. Hagenbuchner, T. H. Tse, F.-C. Kuo, and T. Y. Chen, "Automated functional testing of online search services," *Software Testing, Verification and Reliability*, vol. 22, no. 4, pp. 221–243, 2012.

[17] S. Segura, J. A. Parejo, J. Troya, and A. Ruiz-Cortés, "Metamorphic testing of RESTful web APIs," *IEEE Transactions on Software Engineering*, vol. 44, no. 11, pp. 1083–1099, November 2018.

[18] P. W. Anderson, "More is different," *Science*, vol. 177, no. 4047, pp. 393–396, 1972.

[19] E. B. Fernandez, H. Washizaki, and N. Yoshioka, "Abstract security patterns," in *Proceedings of the 15th Conference on Pattern Languages of Programs (PLoP '08)*, Nashville, Tennessee, USA, Oct. 18–20, 2008.

[20] M. Lindvall, A. Porter, G. Magnusson, and C. Schulze, "Metamorphic model-based testing of autonomous systems," in *Proceedings of the IEEE/ACM 2nd International Workshop on Metamorphic Testing (MET '17), in conjunction with the 39th International Conference on Software Engineering (ICSE '17)*, 2017, pp. 35–41.

[21] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the IEEE/ACM 40th International Conference on Software Engineering (ICSE '18)*. ACM, 2018, pp. 303–314.

[22] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE '18)*. ACM, 2018, pp. 132–142.

[23] A. Dwarakanath, M. Ahuja, S. Sikand, R. M. Rao, R. P. J. C. Bose, N. Dubash, and S. Podder, "Identifying implementation bugs in machine learning based image classifiers using metamorphic testing," in *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '18)*. ACM, 2018, pp. 118–128.

[24] Z. Q. Zhou and L. Sun, "Metamorphic testing of driverless cars," *Communications of the ACM*, vol. 62, no. 3, pp. 61–67, March 2019. [Online]. Available: https://doi.org/10.1145/3241979

[25] S. Levin, "Uber crash shows 'catastrophic failure' of self-driving technology, experts say," https://www.theguardian.com/technology/2018/mar/22/self-driving-car-uber-death-woman-failure-fatal-crash-arizona, March 23, 2018.

[26] A. Ohnsman, "Lidar maker velodyne 'baffled' by self-driving uber's failure to avoid pedestrian," https://www.forbes.com/sites/alanohnsman/2018/03/23/lidar-maker-velodyne-baffled-by-self-driving-ubers-failure-to-avoid-pedestrian, March 23, 2018.

[27] D. Lee, "Sensor firm velodyne 'baffled' by uber self-driving death," http://www.bbc.com/news/technology-43523286, March 23, 2018.

[28] M. Posky, "Lidar supplier defends hardware, blames uber for fatal crash," http://www.thetruthaboutcars.com/2018/03/lidar-supplier-blames-uber/, March 23, 2018.

[29] A. C. Madrigal, "The FAA rigorously tested the Boeing 737's software. So how did a problem slip through?" The Atlantic, March 14, 2019. [Online]. Available: https://www.theatlantic.com/technology/archive/2019/03/boeing-737-max-8-safe-how-faa-tested-its-software/584848/

[30] D. Pesu, Z. Q. Zhou, J. Zhen, and D. Towey, "A Monte Carlo method for metamorphic testing of machine translation services," in *Proceedings of the IEEE/ACM 3rd International Workshop on Metamorphic Testing (MET '18), in conjunction with the 40th International Conference on Software Engineering (ICSE '18)*. ACM, May 27, 2018.

[31] H. Somers, "Round-trip translation: What is it good for?" in *Proceedings of the Australasian Language Technology Workshop*, 2005, pp. 127–133.

[32] M. Aiken and M. Park, "The efficacy of round-trip translation for MT evaluation," *Translation Journal*, vol. 14, no. 1, 2010. [Online]. Available: http://translationjournal.net/journal/51reverse.htm

[33] L. Sun and Z. Q. Zhou, "Metamorphic testing for machine translations: MT4MT," in *Proceedings of the 25th Australasian Software Engineering Conference (ASWEC 2018)*. IEEE, 2018, pp. 96–100.

[34] M. Srinivasan, "Prioritization of metamorphic relations based on test case execution properties," in *Proceedings of the International Symposium on Software Reliability Engineering Workshops*. IEEE, 2018, pp. 162–165.

[35] M. V. Govindan and C. E. Sekeris, "Purification of two dexamethasone-binding proteins from rat-liver cytosol," *European Journal of Biochemistry*, vol. 89, pp. 95–104, 1978.