Metamorphic Testing for Machine Translations: MT4MT

Conference Paper · November 2018

DOI: 10.1109/ASWEC.2018.00021

CITATIONS
11

READS
335

2 authors:

Zhi Quan Zhou
University of Wollongong
69 PUBLICATIONS **1,286** CITATIONS

SEE PROFILE

Liqun Sun
University of Wollongong
9 PUBLICATIONS **123** CITATIONS

SEE PROFILE

2018 25th Australasian Software Engineering Conference (ASWEC)

Metamorphic Testing for Machine Translations: MT4MT

Liqun Sun
School of Computing and Information Technology
University of Wollongong
Wollongong, NSW 2522, Australia
ls168@uowmail.edu.au

Zhi Quan Zhou *
School of Computing and Information Technology
University of Wollongong
Wollongong, NSW 2522, Australia
zhiquan@uow.edu.au

***Abstract*—Automated machine translation software and services have become widely available and increasingly popular. Due to the complexity and flexibility of natural languages, automated testing and quality assessment of this type of software is extremely challenging, especially in the absence of a human oracle or a reference translation. Furthermore, even if a reference translation is available, some major evaluation metrics, such as BLEU, are not reliable on short sentences, the type of sentence now prevailing on the Internet. To alleviate these problems, we have been using a metamorphic testing technique to test machine translation services in a fully automatic way without the involvement of any human assessor or reference translation. This article reports on our progress, and presents some interesting preliminary experimental results that reveal quality issues of English-to-Chinese translations in two mainstream machine translation services: Google Translate and Microsoft Translator. These preliminary results demonstrate the usefulness and potential of metamorphic testing for applications in the natural language processing domain.**

***Index Terms*—Machine translation, software testing, quality evaluation, oracle problem, metamorphic testing, MT4MT.**

I. Introduction

Machine translation services provide automatic translation of text or speech from one language to another. This type of software must be thoroughly tested and evaluated so that developers can understand the advantages and disadvantages of different algorithms as well as locate implementation bugs, and that users can compare different translation services for their information needs. As a result, better evaluation metrics for machine translations are on demand [1].

The determination of machine translation quality normally depends on the judgment of the human assessor [2]. This means that the evaluation is both *subjective* and *expensive*. Furthermore, in the context of automatic quality assessment, an equivalent target language text (reference translation, or oracle translation) is normally needed when it comes to quantitative quality indicators for the translation. However, given the complexity and flexibility of natural languages, using one reference translation as "the standard answer" is obviously not the best solution (except in some highly technical domains), not to mention that a reference translation is often unavailable in practical situations [2], [3].

*All correspondence should be addressed to Zhi Quan Zhou.

The above difficulty is generally known as the *oracle problem*, where an *oracle* refers to a mechanism against which the tester can decide whether the outcomes of test case executions are correct. The oracle problem is a fundamental challenge in software testing, which refers to situations where an oracle is unavailable, or is theoretically available, but practically too difficult or expensive to be applied [4], [5].

A major approach to addressing the oracle problem is known as *metamorphic testing* (MT) [6], [7], a property-based software testing and quality assurance paradigm. MT has been studied by a growing body of research [4], [8]–[10], and has been adopted by industry and organizations, such as Adobe [11], [12], NASA [13], [14], Accenture (including both a research paper [15] and a patent titled "Verifying Machine Learning through Metamorphic Testing" [16, p. 12]), and the US National Institute of Standards and Technology [17]. Examples of successful applications of MT to real-life critical systems include the detection of previously unknown bugs in various compilers [18]–[20], commercial obfuscators [21] and, more recently, self-driving cars [22]. In Aug 2018, Google acquired GraphicsFuzz [23], a spinout company from Imperial College London, to apply metamorphic testing to graphics drivers [20], [24], [25].

To explore the usefulness of *metamorphic testing for machine translations* (**MT4MT**), we have previously reported some promising results [2], where a Monte Carlo method for MT4MT has been developed. We recognize that natural language processing is a complex task and, hence, one testing approach is not enough. In the present research, we continue to investigate MT4MT beyond Monte Carlo approaches. Our research question is stated as follows: *Can new MT4MT techniques be developed to automatically detect real-life defects in major machine translation services?*

The rest of this paper is organized as follows: Sec. II introduces the background and key concepts of this research. Sec. III describes a main component of our testing method. Sec. IV explains the design of our experiments, and Sec. V analyzes the experimental results. Sec. VI compares our research with related work. Sec. VII concludes the paper and points out future research directions.

2377-5408/18/$31.00 ©2018 IEEE
DOI 10.1109/ASWEC.2018.00021

II. Background

Compared with conventional testing techniques, metamorphic testing has a unique characteristic: Instead of verifying the correctness of each individual output, it examines the *relations* among the inputs and outputs of *multiple* executions of the program under test. Such relations are called *metamorphic relations* (MRs), which are necessary properties of the intended program's functionality [9].

Consider the testing of machine translation software. A straightforward MR is known as *round-trip translation* (RTT) [26]–[29]: First, translate a string L_1 from the original language O to the target language T, resulting in a string L_2; then, translate L_2 back to the original language O, resulting in a string L_3; finally, assess the translation quality by comparing L_1 and L_3: the closer, the better. The main issue with RTT is that it involves two-way translations: the forward translation (FT) and the back translation (BT). Therefore, it could be hard to assess the quality of one-way translations. Nevertheless, it has been reported that RTT could still be useful. For example, Aiken and Park [29] stated that: "*RTT is **the only** technique that can be used when no human fluent in the target language or equivalent text is readily available.*"

To address the above problem, we developed a fully automated **non-RTT** technique [2] that can be used without an equivalent target language text (reference translation), or proficient (fluent) target language user. In this approach, a one-way evaluation method was implemented without referring back to the source language. The general idea was that a *perfect* translator should produce the same translation "when translating either directly (from a *source* language to a *target* language) or indirectly (from the source language to an *intermediate* language and then from the intermediate language to the target language)" [2]. To implement this MR, a Monte Carlo method was used to measure the quality (consistency) of translation, with the help of multiple intermediate languages. The testing process is depicted in Fig. 1, where, for illustration purposes, English, Chinese, and Japanese are the source, target, and (randomly chosen) intermediate languages. In the example shown in Fig. 1, two English-Chinese translations are collected: one direct (English-Chinese) translation, and one indirect (English-Japanese-Chinese) translation. The similarity of these two translations (in Chinese) are calculated using standard text similarity metrics to indicate the translation quality: the higher, the better.

The above method requires the existence of intermediate languages and a large number of test runs to make the results statistically meaningful. In the present research, we aim to develop a simpler metamorphic relation that does not involve intermediate languages.

III. Our Metamorphic Relation

Our metamorphic relation is named $\text{MR}_{\text{replace}}$: In some situations, if we change the value of a relatively independent component in a system's input, then only a small part (or no part) in the system's output should be changed. In other words, $\text{MR}_{\text{replace}}$ observes that some changes to the input

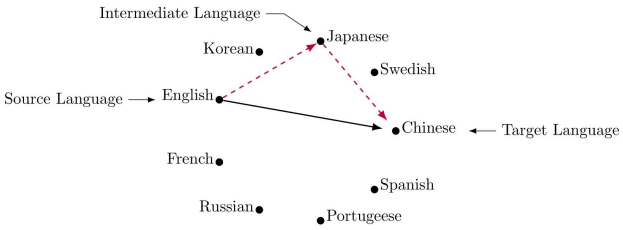

Fig. 1: Pesu et al.'s [2] metamorphic relation that compares direct and indirect translations. The black line shows the direct translation and the coloured line shows the indirect translation involving an intermediate language. A total of eight intermediate languages were used in the Monte Carlo metamorphic testing approach. This figure is taken from [2].

should not have an impact on the overall structure of the output. For example, consider the obstacle perception module [22] of a self-driving car — changing the colour of a nearby truck should not affect the detection of this obstacle — the self-driving car should always detect this truck regardless of whether the truck is red, green, or blue. Therefore, in metamorphic testing, we can change the colour of surrounding vehicles and check whether the self-driving car has the same behaviour. In this situation, $\text{MR}_{\text{replace}}$ is used to replace a vehicle's colour (e.g. red) with a different value (e.g. white).

We can apply $\text{MR}_{\text{replace}}$ to *automatically* test English-to-Chinese translation services. As we know both languages well, we can design rules to generate valid test cases. For example, the Chinese translations for "A_1 loves B" and "A_2 loves B" (where A_1, A_2, and B are nouns) should have a very similar structure except that, in the latter translation, the Chinese word corresponding to A_1 should be replaced by another Chinese word corresponding to A_2. Fig. 2 shows a violation of this MR detected in Google Translate: Changing the subject from "Mike" to "Mouse" has led to an inconsistent Chinese translation for *KFC* (the translation of KFC has changed from 肯德基 to 肯德鸡). Even without a test oracle to tell which one is the authoritative Chinese name of KFC, we have detected an inconsistency issue in the translations. All translation inconsistencies reported in this paper were repeatable at the time of experimentation.

It may be argued that a good translation should consider the context, and that just looking at a single short sentence is not enough to assess the translation quality. On today's Internet, however, the demand for evaluating translations of short sentences is rapidly increasing with the growing popularity of social network sites. For example, it has been reported that a Palestinian man was arrested by police after posting "Good morning" in Arabic which was wrongly translated as "attack them" by Facebook [30], [31]. A screenshot of the news is shown in Fig. 3. Furthermore, it has been reported that some evaluation metrics, including BLEU and NIST, do not perform well at the individual sentence level [32]. In this research, therefore, we focus on the translation quality of short sentences without using these metrics.

(a) "KFC" is translated into "肯德基" in Chinese.

(b) "KFC" is translated into "肯德鸡" in Chinese

Fig. 2: Violation of $MR_{replace}$ in Google Translate. (Note that the word "love" instead of "loves" is used to test the robustness of the translator in dealing with minor grammatical errors.)

The man uploaded a picture of himself leaning against a bulldozer at the Israeli settlement of Beitar Ilit, where he works, in the occupied West Bank

In the caption, he wrote an Arabic term meaning 'good morning', but a software malfunction translated it to mean 'attack them' in Hebrew and 'hurt them' in English

Fig. 3: Facebook translation error leading to arrest [30].

IV. Design of Experiments

The systems under test (SUTs) are (1) Google Translate (https://translate.google.com.au) and (2) Microsoft Translator (https://www.bing.com/translator). These two SUTs are selected because of their popularity. We have developed a tool in Python to implement our testing method, by calling the translation APIs to translate English sentences into Chinese. All tests were conducted in March 2018.

We have constructed very short sentences using the *subject-verb-object* structure. We have tested two verbs: *likes* and *hates*. More specifically, to generate a subject-verb-object sentence and to ensure that the subject noun is unrelated to the object noun, we selected the top 100 most popular US female names (for the year of birth: 2016) from the US Social Security Administration website [33] as the subject nouns, and the top 100 brands (in the year 2015) [34] as the object nouns.

TABLE I: Test results.

translator	Google Translate		Microsoft Translator	
verb	likes	hates	likes	hates
number of unique comparisons	990000	990000	990000	990000
noun inconsistency	137438	159044	34670	36779
inconsistency rate (%)	13.88	16.07	3.50	3.72

Hence, for Google Translate, $100 \times 100 = 10,000$ English-to-Chinese translations were performed for the verb "likes," and another series of $100 \times 100 = 10,000$ English-to-Chinese translations were performed for the verb "hates," giving a total of $20,000$ translation results. The same $20,000$ English sentences were also translated into Chinese using Microsoft Translator. This gave $20,000 + 20,000 = 40,000$ translations (calls to the translation APIs) conducted by Google Translate and Microsoft Translator in total.

V. Analysis of Test Results

We divide the $40,000$ translations into four groups: Google-likes, Google-hates, Microsoft-likes, and Microsoft-hates. Each group contains $100 \times 100 = 10,000$ translations in Chinese. Within each group, we conduct pairwise comparisons between the Chinese translations whose source English sentences **differ in only one noun**. For example, if there are only two subject nouns A and B, and two object nouns X and Y, then for the verb "likes," there will be four English source sentences (test cases): (1) A likes X. (2) A likes Y. (3) B likes X. (4) B likes Y. Let their respective Chinese translations be C_1, C_2, C_3, and C_4. To evaluate these translations, we compare C_1 with C_2, C_1 with C_3, C_2 with C_4, and C_3 with C_4. Note that we do not compare C_1 with C_4, because their respective English source sentences differ in more than one word. For the same reason, we do not compare C_2 with C_3. Neither do we compare translations whose source English sentences contain different verbs.

Because each of the four groups contains 100×100 translations, we conduct a total of $C_{100}^2 \times 100 \times 2 = 990,000$ unique pairwise comparisons within each group. A violation (inconsistency) is detected if the pair of translations under comparison differ in more than one place.

A. Results of Experiments

The comparison results are summarized in Table I. For each of the two translators, its likes- and hates-inconsistency rates were similar, with the former being slightly lower. Compared with Google Translate (13.88% likes- and 16.07% hates-inconsistency rates), Microsoft Translator had much better performance: The respective inconsistency rates were 3.50% and 3.72%.

B. Examples of the Detected Translation Issues

The first type of translation inconsistency issue (that is, violation of $MR_{replace}$) is that the same English name was translated into different Chinese names. For example, Fig. 4 shows that Microsoft Translator translated the English name "Nora" (in the sentence "Nora likes John Deere") into the Chinese name 娜拉, but translated the same "Nora" (in

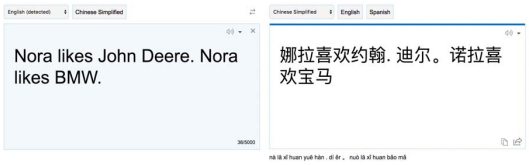

Fig. 4: An Example of translation inconsistency: The same "Nora" was translated into two different Chinese names by Microsoft Translator.

(a) Translation failure for "Layla."

(b) Translation failure for "Mini."

Fig. 5: Google Translate failures: The same names were sometimes translated and sometimes not translated.

the sentence "Nora likes BMW") into a different Chinese name 诺拉. Each of these two translations, when examined independently of each other, is acceptable; however, when we compare them using $MR_{replace}$, a translation defect is revealed: The same "Nora" should not have been translated into two different Chinese names, which represent two different people — these translations will definitely cause misunderstanding among Chinese readers.

The second type of $MR_{replace}$ violation is that the same English name was sometimes translated and sometimes not translated. For example, Fig. 5a shows that Google Translate translated the English name "Layla" (in the sentence "Layla likes Ford") into the Chinese name 莱拉, but failed to translate the same "Layla" in the second sentence ("Layla likes Google") — Google Translate directly pasted the string "Layla" into the Chinese sentence without translation. This is a translation failure, causing misunderstanding among Chinese readers as they would consider these two names to represent two different people.

Fig. 5b shows a similar failure, where Google Translate successfully translated the brand name "Mini" in the first sentence into Chinese, but failed to translate the same "Mini" in the second sentence.

VI. Related work

Zheng et al. [3] developed two algorithms that can be used to detect under-translations (where some words from the text in the original language are missing after translation) and over-translations (where some words from the text in the original language are unnecessarily translated two or more times) in neural-network-based machine translations. Their approach detects the target translation issues without referring to a reference translation, and thus is called "oracle-free detection." Zheng et al. [3] reported that their approach had been deployed in both the development and production environments of WeChat by Tencent Inc. to help eliminate numerous defects of their neural machine translation model.

Our approach presented in this paper is not limited to the detection of under- or over-translations, but instead we examine the consistency among *multiple* translations. Therefore, our approach and Zheng et al.'s approach [3] complement each other.

VII. Conclusion

In this paper, we followed up on our previous work in applying metamorphic testing for machine translations (MT4MT) without the need for a test oracle [2]. The main contributions of this paper include the identification of a metamorphic relation $MR_{replace}$, and the empirical results using $MR_{replace}$.

A limitation of this study is that we did not consider the effect of constructing test sentences using names that are also nouns or verbs (such as Bill, Harry, and Sue). Furthermore, the empirical results are only with respect to the consistency metric rather than the correctness of the translations (which would require human evaluations). The metamorphic relation, $MR_{replace}$, provides a guideline, or a pattern, but cannot be used in any absolute sense, because it is possible that similar English sentences (including idioms) do not have similar Chinese translations.

There is a major threat to the external validity of our results: The scale of our experiments was quite small ($40,000$ translations in total). In particular, our experiments involved only two verbs: "like" and "hate." Nevertheless, within this limited scale, we have made interesting findings, and have successfully addressed the research question raised in Sec. I. Our findings are summarized as follows.

First, major translation services, including Google Translate and Microsoft Translator, can fail to translate extremely simple sentences, or can translate them in a very inconsistent way, causing serious misunderstanding among users. This finding demonstrates the usefulness of $MR_{replace}$ (and, more generally, metamorphic relations) in the natural language processing domain.

Second, the empirical results show that Microsoft Translator strongly outperformed Google Translate. This result is **inconsistent** with that reported in our previous work [2] where we found that Google Translate outperformed Microsoft Translator. We believe that the reason for this inconsistency is that different types of source sentences were used during testing: In our previous work [2], we used relatively long English sentences taken from Wikipedia; whereas in the present study, we used extremely short, simple, and synthetic sentences (with the structure subject-verb-object) involving

only the word "likes" or "hates." This observation means that the quality assessment of machine translation services must consider multiple dimensions and multiple types of user inputs, and that different translators can have very different performance when processing different types of inputs.

Future work will include the identification of a set of *diverse* metamorphic relations, the analyses of different types of inputs, and experimentation at a larger scale.

ACKNOWLEDGMENTS

This work was supported in part by a linkage grant of the Australian Research Council (project ID: LP160101691). We would like to thank Suzhou Insight Cloud Information Technology Co., Ltd for supporting this research.

REFERENCES

[1] C. Liu, D. Dahlmeier, and H. T. Ng, "Better evaluation metrics lead to better machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 375–384.

[2] 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.

[3] W. Zheng, W. Wang, D. Liu, C. Zhang, Q. Zeng, Y. Deng, W. Yang, and T. Xie, "Oracle-free detection of translation issue for neural machine translation," *ArXiv e-prints*, Jul. 2018. [Online]. Available: https://arxiv.org/abs/1807.02340

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] S. Segura and Z. Q. Zhou, "Metamorphic testing 20 years later: A hands-on introduction," in *Proceedings of the IEEE/ACM 40th International Conference on Software Engineering (ICSE '18 Companion)*. ACM, 2018, presentation slides available online at http://doi.org/10.5281/zenodo.1256230.

[11] D. C. Jarman, Z. Q. Zhou, and T. Y. Chen, "Metamorphic testing for Adobe data analytics software," 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. 21–27.

[12] 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.

[13] M. Lindvall, D. Ganesan, R. Árdal, and R. E. Wiegand, "Metamorphic model-based testing applied on NASA DAT – an experience report," in *Proceedings of the IEEE/ACM 37th International Conference on Software Engineering (ICSE '15)*, 2015, pp. 129–138.

[14] J. Rothermel, M. Lindvall, A. Porter, and S. Bjorgvinsson, "A metamorphic testing approach to NASA GMSEC's flexible publish and subscribe functionality," 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. 18–25.

[15] 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.

[16] Accenture. (2018) Quality engineering in the new: A vision and R&D update from Accenture Labs and Accenture Testing Services. [Online]. Available: https://www.accenture.com/t20180627T065422Z__w__/us-en/_acnmedia/PDF-81/Accenture-Quality-Engineering-POV.pdf

[17] N. Mouha, M. S. Raunak, D. R. Kuhn, and R. Kacker, "Finding bugs in cryptographic hash function implementations," *IEEE Transactions on Reliability*, in press.

[18] V. Le, M. Afshari, and Z. Su, "Compiler validation via equivalence modulo inputs," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'14)*, 2014, pp. 216–226.

[19] J. Regehr, "Finding compiler bugs by removing dead code," http://blog.regehr.org/archives/1161, June 20, 2014.

[20] A. F. Donaldson, H. Evrard, A. Lascu, and P. Thomson, "Automated testing of graphics shader compilers," *Proceedings of the ACM on Programming Languages*, vol. 1, no. OOPSLA, pp. 93:1–93:29, 2017.

[21] T. Y. Chen, F.-C. Kuo, W. Ma, W. Susilo, D. Towey, J. Voas, and Z. Q. Zhou, "Metamorphic testing for cybersecurity," *Computer*, vol. 49, no. 6, pp. 48–55, 2016.

[22] Z. Q. Zhou and L. Sun, "Metamorphic testing of driverless cars," *Communications of the ACM*, in press.

[23] GraphicsFuzz homepage. [Online]. Available: https://www.graphicsfuzz.com

[24] A. F. Donaldson and A. Lascu, "Metamorphic testing for (graphics) compilers," in *Proceedings of the IEEE/ACM 1st International Workshop on Metamorphic Testing (MET '16), in conjunction with the 38th International Conference on Software Engineering (ICSE '16)*. ACM, 2016, pp. 44–47.

[25] GraphicsFuzz. How it works. [Online]. Available: https://www.graphicsfuzz.com/howitworks.html

[26] H. Somers, "Round-trip translation: What is it good for?" in *Proceedings of the Australasian language technology workshop*, 2005, pp. 127–133.

[27] P. Koehn and C. Monz, "Manual and automatic evaluation of machine translation between European languages," in *Proceedings of the Workshop on Statistical Machine Translation (StatMT '06)*. Association for Computational Linguistics, 2006, pp. 102–121.

[28] T. Shigenobu, "Evaluation and usability of back translation for intercultural communication," in *Proceedings of the 2nd International Conference on Usability and Internationalization, Lecture Notes in Computer Science, vol 4560*. Springer-Verlag, 2007, pp. 259–265.

[29] 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

[30] G. Davies, "Palestinian man is arrested by police after posting 'good morning' in Arabic on Facebook which was wrongly translated as 'attack them'," *DailyMail*, 2017. [Online]. Available: http://www.dailymail.co.uk/news/article-5005489/Good-morning-Facebook-post-leads-arrest-Palestinian.html

[31] A. Hern, "Facebook translates 'good morning' into 'attack them', leading to arrest," *The Guardian*, 2017. [Online]. Available: https://www.theguardian.com/technology/2017/oct/24/facebook-palestine-israel-translates-good-morning-attack-them-arrest

[32] D. Coughlin, "Correlating automated and human assessments of machine translation quality," in *MT Summit IX: Proceedings of the Ninth Machine Translation Summit, New Orleans, USA, September 23-27, 2003*. Association for Machine Translation in the Americas, 2003, pp. 63–70.

[33] Social Security, USA. (2018) Popular baby names. [Online]. Available: https://www.ssa.gov/cgi-bin/popularnames.cgi

[34] B. Chapman. (2016) The top 100 brands in the world have been revealed. [Online]. Available: https://www.independent.co.uk/news/business/news/apple-most-valuable-brand-iphone-7-google-coca-cola-a7345501.html