Metamorphic Testing of Machine Translation Models using Back Translation

Wentao Gao
The University of Melbourne
Melbourne, Australia
wentaog1@student.unimelb.edu.au

Jiayuan He
RMIT University
Melbourne, Australia
jiayuan.he@rmit.edu.au

Van-Thuan Pham
The University of Melbourne
Melbourne, Australia
thuan.pham@unimelb.edu.au

***Abstract*—Machine translation software has been widely adopted in recent years. The recent advance in deep learning research has massively improved the accuracy and fluency of the translated output. However, incorrect translations may still occur, which causes misunderstandings, and even more detrimental consequences when applying these systems for crucial applications, such as translating legal and medical documents. This calls for methods that can test the correctness of machine translation software efficiently and effectively. In this paper, we propose a method that uses back-translation as a reference for machine translation testing, minimizing the knowledge and use of the NLP tools in the target language so that the same workflow can be applied to test systems translating English to multiple languages. We build a metamorphic testing method using our proposed concept called *contextual referentially transparent input* (CRTI). A CRTI is a piece of text that should have a similar meaning under a certain context in any given language. Our method detects inconsistency between a CRTI in the original sentence and the back-translation to report translation errors. To evaluate our method, we translate 200 sentences using Google Translate. Our method reports 57 suspicious issues with a precision of 74% in Chinese translation and 22 suspicious issues with a precision of 82% in Vietnamese translation.**

***Index Terms*—metamorphic testing, machine translation testing, back-translation**

I. Introduction

Machine translation software aims to automatically translate texts from a source language to a target language. The state-of-art machine translation platforms are supported by neural machine translation models with much-improved accuracy compared with old models. Machine translation has been widely used in daily life for international trading and communication. However, errors still exist in machine translation software, causing misunderstandings and conflicts. Thus, automatic methods, which can automatically detect the errors made by machine translation software, are of great interest.

However, it is challenging to identify errors in machine translation software. One of the biggest challenges is in generating a clear test oracle because an optimal translation does not exist, and a human oracle is both expensive and error-prone. The key insight here is that instead of getting a correct translation output from a single input, researchers would create a new test oracle to clearly pick up an error without knowing the ground truth. Metamorphic testing is a commonly adopted testing method to create this test oracle. Researchers would create or collect a group of inputs and check the relation among the inputs and outputs in multiple executions of the system that is under test [19]. This is called *metamorphic relation*.

Recently, He et al. [7] identified an effective metamorphic relation based on the concept of *referentially transparent inputs* (RTI) to test the correctness of machine translation models. The concept of RTI is taken from the term *referential transparency* [15]: an expression is called referentially transparent if it can be replaced with its corresponding value (and vice-versa) without changing the program's behavior. In translation testing, an RTI refers to a piece of text that should always have similar meanings in any context. Based on this metamorphic relation, He et al. designed a novel test oracle: a translation is correct if the stand-alone translation of an RTI appears in the translation of the whole sentence. For example, if we consider the phrase "Another three decades" as an RTI, a sentence such as "He served his country for another three decades" should have the translation of the phrase "another three decades" in the target language. If the translation means "three years" instead of "three decades", a translation error is reported. This approach successfully found many bugs in Google Translate and Microsoft Bing Translator.

TABLE I
Limitations in the testing method using RTI without context in He et al.'s work [7]

RTI pair	Translations	Translation meaning
I read the Wall Street Journal story	我读了华尔街日报的新闻	I read the Wall Street Journal news report
Wall Street Journal story	华尔街日报故事	Wall Street Journal's anecdote

However, we have identified two key limitations in the current state-of-the-art metamorphic testing approaches for machine translation models [6], [7], [16], [17]. While the first limitation is specific to the RTI-based metamorphic relation proposed by He et al. [7], the second one is a general problem.

First, the assumption that RTIs should be translated to similar results —with any context—may not be true. Table I gives an example of an RTI translated with and without its context. The phrase "Wall Street Journal story" is identified as an RTI. When translating with context in row 1, we have high confidence that the meaning of "story" is closer to "news item". The meaning is also correctly captured by the Chinese translation. However, if we feed the phrase "Wall Street Journal story" to the translation engine alone, the Chinese translation result becomes "Wall Street Journal's anecdote", which is different from the meaning of the phrase under context. In this case, a short phrase may not have a correct translation, it could be interpreted in multiple ways. However, [7] reports this as an error in translation. We argue that this can be considered a false positive because the translation of the full sentence is indeed correct. Having too many false positives in a testing method could waste developers' time doing verification. To address this issue, we introduce a new concept called *contextual referentially transparent inputs* (CRTI). CRTI is a piece of text that should have a similar meaning under a certain context in any given language. Unlike [7], we will check the correctness of CRTI translations in context (i.e., in full sentences). We create two types of CRTI, named entities and general noun phrases.

The second limitation is that current approaches rely on knowledge of the target language and supporting tools (e.g., constituent parsing [8], part-of-speech tagging [2]) to apply their testing methods. A developer/tester would need to know detailed knowledge of the target language if they want to reason on errors reported or check the performance of the testing method. Moreover, the reliance on supporting tools in the target language makes testing translations for low-resource languages more difficult. To address this general problem of current state-of-the-art metamorphic testing approaches to checking correctness of machine translation models, we introduce a back-translation step to the workflow. Back-translation is the "re-translation" of the translated sentence. For example, in Table I, the "translation meaning" column is the back-translation. Instead of processing the target language, we use the back-translation as a reference and process it, which is in the same language as the source sentence. This solution also helps us apply the novel concept of CRTI because CRTIs require context, and they must be put in full sentences (i.e., original sentence and its back translation).

Using the same example in Table I, our approach identifies the phrase "Wall Street Journal story" as a CRTI. Instead of translating it only from the source to the target language, we first generate the first row in Table I, which includes the full original sentence (first column), its translated version in the target language (second column) and its back-translation (third column). And then, we try to match the closest constituent (i.e. a part of a sentence) in the back-translation to our CRTI phrase. After vectorizing and comparing distances, we identify "Wall Street Journal news report" to be the corresponding CRTI in the back-translation sentence. Since they are similar in meaning, we report the translation to be correct. It is worth noting that since both the original sentence and its back-translation version are in the same language (English in this case), we can leverage the same supporting tools for constituent parsing or part-of-speech tagging etc.

We implemented and evaluated our approach on the Google Translate service. To show that our approach can generalize to different target languages, we tested two translation systems: English to Chinese and English to Vietnamese. The corpus we used is 200 sentences from CNN news which is collected by He *et al.* [6]. In total, our approach identified 57 and 22 suspicious issues for the two translation systems, respectively. We benchmarked the method using RTI by He *et al.* [7] using the updated version of Google translation. Our precision of error detection has improved to 74%, which is 30% higher than the previous research. Our method can report more true errors with higher precision. We produce more false negative examples compared to the previous research, but it is beneficial for developers without knowledge of the target language to have an initial check on the accuracy of the translation system before handling it to experts in a specific language. We would also suggest developers use a combined approach. Our method doesn't require knowledge of the target language and provides a lower bound of the number of errors with greater confidence. The previous method [7] reports more suspicious issues that require scrutiny from language-specific experts.

The types of error we detected are miss-translation, under-translation, over-translation, incorrect-modification, under-logic and over-inference. Most significantly, the over-inference error, which is closely related to our proposed method to compare named entities, is discovered and defined. To the author's limited knowledge, this is the first research to look into this type of translation error. We couldn't find a comprehensive way to report all errors we found to Google, but we manually gave feedback on the Google translate user interface correcting the miss-translated results.

The main contributions of this paper are as follows:

- We propose the use of back-translation as a machine translation correctness testing method without depending on previous knowledge or NLP tools for the target language. This allows developers to investigate translation errors without knowledge of the target language.
- We propose the concept of CRTI to build metamorphic relations using noun phrases and named entities for machine translation testing.
- We implemented the proposed method to test Google translation. The results show that our approach improves the precision of error detection to 74%, which is 30% higher than the previous research.

II. Background and Related Work

Machine learning testing refers to any activity that reveals machine learning bugs [18]. Multiple properties are being tested including correctness, robustness, efficiency, fairness and security. Most studies in machine translation testing work on testing correctness. Correctness testing focuses on the accuracy of a machine learning system.

Classical correctness testing methods use data with ground truth as test input. In machine translation testing, a parallel corpus (i.e. datasets consisting of paired sentences in both source and target language) is the ground truth [4]. To evaluate the accuracy of machine translation with a parallel corpus, several metrics exist. The BLEU score [12] is the most widely-adopted measurement to check translation quality.

However, in most cases, there may not exist a ground truth translation for monolingual corpus. Metamorphic testing [3] is often used in this case. Metamorphic testing creates a metamorphic relation between a pair of inputs. After feeding this pair of inputs to the system under test and collecting outputs, it assumes that this relation remains in the output pair.

Several research works on machine translation testing with metamorphic testing. They proposed several properties in a pair of sentences that should remain consistent after translation. *TransRepair* [17] and *SIT* [6] are testing methods that make synthesised sentences by replacing words in them. Authors then assume the synthetic sentences have the same structural and grammatical features after translation to test machine translation systems. In a system called *Purity* [7], authors assume a noun phrase should be translated to a similar result in the target translations despite the context. They compare the stand-alone translation of noun phrases with the translation of them under different contexts to test machine translation systems.

Back-translation is a concept used in natural language processing and linguistics. In NLP, it is the process of training an intermediate system on the parallel data and then translating monolingual corpora in the target language back to the source language [4]. Initially proposed to generate synthetic sentences for training machine translation models [14], back-translation can augment the parallel corpus by adding monolingual corpus in training and improve model accuracy. In linguistics, back-translation is the "re-translation" of the translated corpus into the original language. Researchers have used back-translation in questionnaire translation by comparing the original and back-translated questionnaires to assess translation quality [1].

III. Testing ML-based Translation Models with Back-translation and Contextual RTI

A. Back-translation

The reason for involving back-translation to machine translation testing is that most current methods rely on domain knowledge for the target language or the NLP tools developed for it. Although it remains possible to design evaluation algorithms based on models/tools that are specific to the target language, the accuracy of such models may not be reliable, and hence, affect the overall evaluation accuracy, especially if the target language is a low-resource language. Moreover, most of the state-of-the-art language models demonstrate much better performances for English corpus.

In this paper, we demonstrate that introducing back-translation can obtain better testing performances, even for non-low-resource languages. The key idea is to design a testing method that operates on a single language. Therefore, we generate a reference sentence that is almost equivalent to the translated sentence but in the source language. We accomplish this by using back-translation in the workflow. Suppose we have a sentence A in language a. The forward translation of A to a target language b is B. We then translate sentence B back to the original language a, to get a back-translation A'.

B. Contextual Referentially Transparent Inputs

We built upon the concept of RTI by He *et al.* [7] for metamorphic testing in machine translation testing. An RTI refers to a piece of text that should always have similar meanings in any context. He *et al.* [7] then argue that noun phrases under certain constraints are all considered to be RTIs. A noun phrase can be determined by using constituency parsing tools. However, we found two conditions that can undermine this assertion.

First, a noun phrase receives better word sense disambiguation with a longer context. Word sense disambiguation is the process of identifying which sense of a word is activated by its use in a particular context. [11] For example, in a given sentence "River Bank is a financial service company.", we want to make sure bank means "a financial establishment" here, but without the context, one may argue that bank means "land alongside a river". The difference between the two word senses is too far to be similar.

Also, shorter phrases cannot be disambiguated without context, making it difficult to tell the right or wrong of a translation. Take the same example of the word "bank". A short phrase such as "three meters from the bank" is difficult to disambiguate, even for humans. We actually guess the actual meaning based on word frequency.

To create a more robust feature for constructing metamorphic relations, we proposed a concept called Contextual Referentially Transparent Input (CRTI). CRTI is defined as a piece of text that should have a similar meaning in any given language under a certain context. For example, in Figure 1, the meaning of "story" in the phrase "Wall Street Journal story" is closer to "news item", given the full sentence. If we know the abstract meaning of the phrase, we can find equivalent text in the target language for "Wall Street Journal" and "news item". The key insight here is that a highly abstract meaning of something is intellectual in most languages. Imagine a person drawing following a set of instructions. The more detailed instructions normally lead to a more precise drawing. The key improvement of CRTI over RTI is that a sentence with sufficient contextual information can better formulate its abstract meaning.

C. Metamorphic Relations in Machine Translation Testing

The metamorphic relation that we identified is as follows: a pair of CRTIs should be contained in both the source sentence and the back-translation, respectively. However, we assume that a CRTI should be translated with context. We do not consider feeding the stand-alone CRTI to the translation system. Instead, we reconstruct the metamorphic pairs by extracting corresponding constituents from the back-translation. First, we construct a metamorphic pair of CRTI and the full sentence

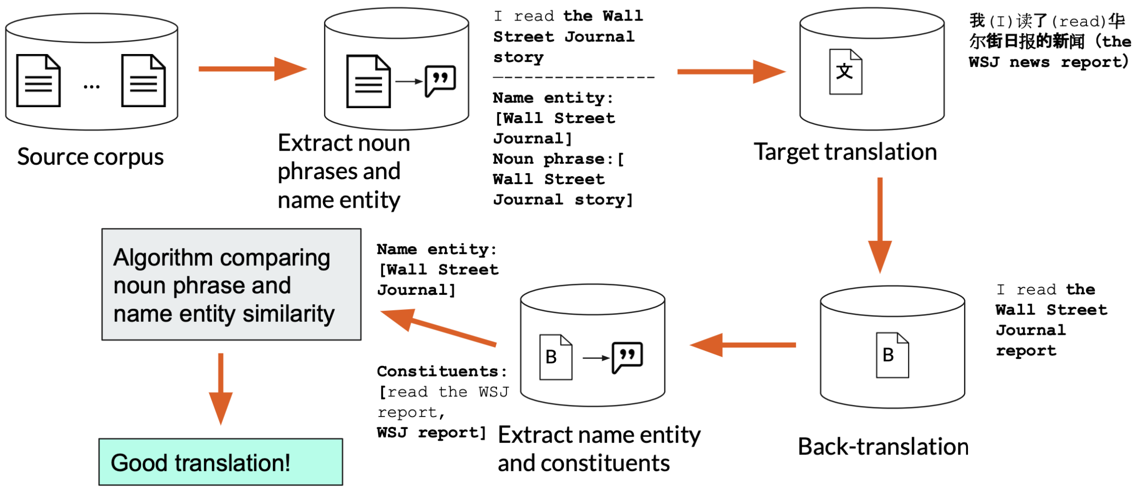
Source corpus
Extract noun phrases and name entity
I read the Wall Street Journal story
Name entity: [Wall Street Journal]
Noun phrase:[Wall Street Journal story]
文
Target translation
我(I)读了(read)华尔街日报的新闻 (the WSJ news report)
B
Back-translation
I read the Wall Street Journal report
B
Extract name entity and constituents
Name entity: [Wall Street Journal]
Constituents: [read the WSJ report, WSJ report]
Algorithm comparing noun phrase and name entity similarity
Good translation!

Fig. 1. An overview workflow of our method, with examples for each step

containing that CRTI. Then, we only translate and back-translate the full sentence. Finally, we use a similarity measurements function *Similarity* to match the closest constituent of the back-translation with the original CRTI. Suppose P is a CRTI we identified in sentence A in language a, the back-translation is denoted as A'. P' is the most similar constituent we can extract from sentence A'. The metamorphic relation here is that

$$Similarity(P, P') > t \tag{1}$$

where P is a sub-sequence of A and P' is a sub-sequence of A'; t is a threshold which can be different depending on the type of P we identified.

The next question to answer is which part of the sentence can be considered a CRTI. He *et al.* [7] argued that noun phrases in a sentence are RTIs. After considering common translation errors, we would also consider noun phrases as CRTIs. However, we also divided noun phrases into two categories, named entities and others. A named entity is a real-world subject, such as a person, location, organisation or product, that can be denoted with a proper name [10]. To construct a similarity comparison threshold in Equation 1, a stricter criterion (i.e. bigger threshold t) is set for named entities, while a relaxed comparison (i.e. smaller threshold t) is performed to compare inconsistency between other noun phrases. We assume that CRTIs coming from named entities are more similar to each other, the part-of-speech or phrase structure remains relatively consistent. However, the difference should be bigger for other noun phrases because there may exist a way to rephrase them to another grammar structure or use other words, but the meaning is preserved. For example, “a three-year study” is a noun phrase, but “a study for three years” is a noun phrase plus a prepositional phrase. These two phrases have very similar meanings. Therefore, unlike named entities, we would consider any constituent in the back-translation as a candidate to compare similarity with general noun phrases, and hence reduce false negatives in error detection.

IV. Algorithms Design and Implementation

This section introduces our implementation to test machine translation systems based on back-translation. If the metamorphic relation in Equation 1 is violated, we report a suspicious translation error. This suspicious error has two main causes. The first type of error comes from the translation of the context. This error makes the machine translation system unable to get a precise meaning of the CRTI. The second type of error is a translation error that happens to the CRTI even though the context is correct. Moreover, the key benefit of this metamorphic relation is that the suspicious issue can always be located in the full sentence. Because we only execute the system under test once using the full sentence. The metamorphic pair is reconstructed in back-translation. We directly report that the translation of the full sentence is incorrect. Note that there are two transactions to generate the back-translation(i.e. language a to b, then back to a). Our method hypothetically reports an error in either of the systems. Fig 1 shows the workflow of our method, which has the following five steps.

1) **Extract CRTIs**. For each source sentence, we perform constituency parsing and named entity recognition (NER) to extract CRTIs.
2) **Generate translation**. Feed the source sentence to the machine translation engine that we want to test and gather its translation as the target sentence.
3) **Generate back-translation**. Feed the target sentence generated by previous steps to the same machine translation engine and call the reverse way (i.e. translate the target language to the source language) to gather its back-translation
4) **Reconstruct CRTI pairs**. In back-translation, find the constituent that has the closest contextual representation as the original CRTI.

5) **Report suspicious errors**. Calculating the similarity between the original CRTI and the one in back-translation. If the similarity is smaller than a threshold, we report that the source sentence is wrongly translated.

The tools used in each step and their detailed process of them will be described in the following sections.

A. Extract CRTI

As mentioned in section III-C, we extract two types of CRTIs, named entities and noun phrases. The following two sections describe how we extract and post-process extracted named entities or noun phrases, respectively.

1) Identify Named Entity: We used one of the state-of-the-art NER tools [13] to extract named entities. A NER tool takes a corpus as input and produces identified named entities with attributes such as entity type and confidence score. We filtered out named entities that are categorised as type "miscellaneous". These are named entities detected by the tool to be proper nouns. They may not have a standard translation in a target language. In Figure 1, a list of named entities is extracted, such as "Wall Street Journal" and "Wall Street".

2) Identify Noun Phrase: Next, a series of noun phrases are identified based on their part-of-speech tag. We used the Berkeley Parser [9], which is one of the state-of-art constituency parsers to predict the part-of-speech tag of the input sentence. Phrases marked as noun phrases are filtered out. We adopted a set of tuned parameters in [7] to post-process noun phrases. This step is to make sure that most noun phrases carry a single meaning and reduce false positives. In Figure 1, we have a list of noun phrases, including "the Wall Street Journal story".

B. Generate Translation and Generate Back-translation

The next step is to gather a translation of the original sentence and the back-translation of the translated sentence. We called the API of the translation engine under test (We tested Google translation) twice to generate a forward translation function and a backward translation function.

C. Reconstruct CRTI Pairs

In this step, we want to locate a part of text that most correspond to each of the CRTIs in the original sentence. The first step is to extract a pool of candidates matching either a named entity or a noun phrase. Then a similarity measure is adopted to link one CRTI from the original sentence to a specific one extracted from the back-translation.

We used a pre-trained model to do contextual representation that converts each noun phrase to a vector and calculates cosine similarity between them. The pre-trained model comes from [5]. The key benefit of using this work is that its sentence embedding amplifies the difference between contradictive sentences, even when the edit distance is small.

To match named entities in the back-translation, we use the same NER tool to extract named entities to be the pool of candidates. To match other noun phrases, the criteria are relaxed, and any constituent of the sentence is accepted as a candidate.

Finally, we build a CRTI pair including the original CRTI and the most similar candidate. The similarity score is also collected for error detection.

D. Report Suspicious Errors

We compare and report suspicious errors based on reconstructed CRTI pairs and similarity scores. The following two equations expand Equation 1 to denote the test oracle of both types of CRTIs. Multiple tests are conducted depending on the number of CRTI pairs extracted from a source sentence, if one of the pairs violates its own test oracle, the target sentence will be reported by the system as a suspicious error.

$$cosine\_sim(\mathbf{V}_{NE}, \mathbf{V'}_{NE}) > t_1 \tag{2}$$

$$cosine\_sim(\mathbf{V}_{NP}, \mathbf{V'}_{con}) > t_2 \tag{3}$$

Where $\mathbf{V}_{NE}$ and $\mathbf{V}_{NP}$ denote the embedding vector of named entities and noun phrases in the source sentence, respectively. $\mathbf{V'}_{NE}$ and $\mathbf{V'}_{NP}$ denote the corresponding named entities and constituents in the back-translation. The first equation denotes the test oracle for named entities, a threshold t_1 is set. The second equation denotes the test oracle for other general noun phrases. By comparing noun phrases in the original sentence and constituents in back-translation, a threshold t_2 is set. t_1 is bigger than t_2 because we expect a higher consistency among named entities. We manually labelled the correctness of all the translations in our dataset. Therefore the threshold is essentially for a binary classification using one feature only. The threshold is learnt from a 10-fold cross-validation.

V. Evaluation

In this section, we evaluate the performance of our implementation to test Google translation. Specifically, this section aims to answer three research questions:

RQ-1: Can back-translation be used for metamorphic testing of Machine Translation models and what is the accuracy of this approach?

RQ-2: Does named entity recognition help noun phrase discovery?

RQ-3: Is it possible to test machine translation without the knowledge of the target language?

These three research questions are explored for the following reasons. As discussed in the introduction, we identified two limitations of the previous research: high reliance on knowledge of the target language and ambiguous meaning of noun phrases without context.

For RQ-1, we want to see if back-translation helps with both limitations. Because back-translation ensures we only operate on a single language, and we ensure all noun phrases of interest are translated under a fixed context. For RQ-2, we want to see if named entity recognition helps with the second limitation, whether we can discover strong noun

TABLE II
PRECISION OF ERROR DETECTION, ON TWO DATASETS, COMPARED WITH THE BASELINE

Precision	Proposed method	Baseline - RTI
Google - Politics	21/27 (**77.8%**)	23/47 (48.9%)
Google - Economics	21/30 (**70.0%**)	12/34 (35.3%)

phrases with no meaning shift. For RQ-3, we want to answer whether our method helps with the first limitation and could be generalisable.

A. *Experiment and Datasets*

We tested translation systems for English-to-Chinese translation and English-to-Vietnamese translation. In particular, we perform detailed error analysis for English-to-Chinese translation tasks to further understand our proposed evaluation framework. English-to-Vietnamese translation is evaluated mainly based on the precision of error detection as an indication to answer RQ-3. The workflow for the two different target languages is the same, our proposed method can report suspicious issues in both languages only by changing the language variable in Google translation API. The only modification made is that a different threshold is used for t_2 (i.e. threshold comparing noun phrase constituency). t_2 is individually tuned for the two languages to achieve a better result.

The framework *Purity* proposed in [7] using RTI to test English to Chinese translation serves as a baseline. Because we can not retrieve the old version of Google translation API, the translation results may be different from the time when this research was conducted. As a result, we evaluate their method based on new translation results. Therefore, the accuracy of the baseline method is different to the original paper.

The experiments will use real-world sentences without lexical and syntactical errors as input to machine translation software. The dataset is collected in [6]. using news from CNN. This dataset consists of two corpora: Politics and Business, with each corpus having 100 sentences.

B. *Performance Evaluation*

1) Chinese Translation: In this section, we evaluate the performance of our method testing Chinese translation. Three metrics are included in the following discussion, the precision, the recall rate and common false positive instances of our method.

Precision. The precision is presented in Table II. It can be interpreted that our proposed method on Google translation for Political news reported 27 suspicious errors, with 21 of them being true errors labelled by humans. As we can see from the table, our method achieves around 74% precision. Our method using back-translation improved accuracy by around 30% compared to the previous baseline method. In terms of the number of true errors detected, we have seen a large improvement in the Economic news category.

Next, we constructed two different types of CRTIs and designed their own test oracle. We also calculated the accuracy of each type. Here, we use the accuracy of general noun phrases to quantify the contribution of back-translation. The reason is that compared to the original method, the methodologies to extract noun phrases are similar. The only difference is that we operate on back-translation in the original language instead of the target language. Table III shows the result.

TABLE III
PRECISION OF ERROR DETECTION – EACH COMPONENT

Precision	Proposed method - NER only	Proposed method - noun phrase only	Baseline - RTI
Google - Politics	11/12 (**91.6%**)	13/18 (**72.2%**)	23/47 (48.9%)
Google - Economics	8/10 (**80.0%**)	15/23 (**65.2%**)	12/34 (35.3%)

As we can see in table III, the error detected by each type of our CRTI improves the precision over the baseline method. The precision considering named entity only is very high at 80% to 92%. It can answer **RQ-2** that adding NER to the workflow improves the quality of noun phrase discovery. Note that the number of error detected is relatively low. This is bacause named entities are treated as noun phrases in the baseline method. But extracting them separately and setting a stricter threshold for comparison yield very accurate and robust results. Next, if we look at the third column, the precision considering other noun phrases is relatively acceptable compared to the baseline method at 65% to 72%. But the key contribution is that we build a method that does not know the ground truth or any information in the target language. The precision still improves by around 25% compared to the baseline. This answers **RQ-1**. Back-translation can be introduced to machine translation testing. The key insight here is that machine will be better at translating sentences that are generated by itself, so we could trust the back-translation as a reference. Because the target translation can activate neurons in the language-generating module. This makes it easier for the same neural network to capture this generated sentence compared to capturing a sentence written by humans.

Recall Rate. Figure 2 is a Venn diagram to show the distribution of errors in this search space. Here, we manually identified 62 errors in the translation. Our method, which is in the yellow circle, detects 42 errors in the search space. The recall rate is around 68%. Even though the method is tuned to produce higher precision, the recall rate is still higher than the baseline method. We discovered that the error unique to our method mainly comes from two parts. Firstly, errors detected by comparing named entities may not be covered by the previous research. Also, the introduction of back-translation changes the way to compare noun phrase consistency. It also explains why the baseline method has its own unique reported issue. Two errors that we identified can not be reported by any

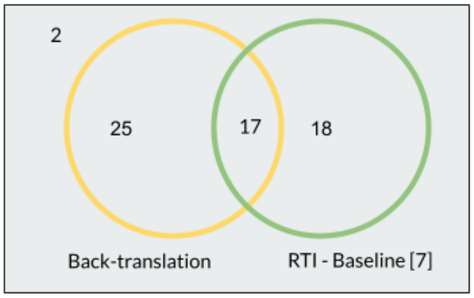

Fig. 2. Erroneous translation reported by back-translation, baseline and all errors

of our methods. These errors do not come from inconsistency in noun phrases and introduce some false negatives to our method.

TABLE IV
EXAMPLE: FALSE POSITIVES

Source	Back-translation
Chevy should keep the plant	Chevrolet should keep the plant
A very large retailer that has a large number of locations	a very large retailer that has a large number of stores

False Positives. Two types of false positives can be found in the experiment. Table IV gives an example of each type. Firstly, comparing the similarity of named entities is still difficult. One extreme case is that named entities may have nicknames, and it is difficult to relate them to the original form. As shown in the table, the translation engine is able to capture “Chevy” as a nickname of “Chevrolet”, but our method regards them as different words. The second cause is the major source of our false positives. A phrase may have multiple correct translations and therefore creates different back-translations. This causes a relatively lower similarity score between the source CRTI and constituent in the back-translation. The threshold we tuned has a “grey zone” where it is close to the decision boundary itself. Intuitively, there will be false positives and false negatives because of the threshold.

2) Vietnamese Translation: This section briefly introduces the precision of testing Vietnamese. We applied the same workflow to test English - Vietnamese translation successfully. The reported suspicious issue is then evaluated by one of the authors who is a native Vietnamese speaker.

Precision. We can see that from Table V, there are two thresholds used in this experiment, a threshold tuned in testing Chinese translation and a newly learnt threshold. However, the TP rate using the threshold adopted from Chinese is not acceptable at 63.6%. Which is shown in the left column. We performed another round of cross-validation using labels in Vietnamese to learn a new threshold, the precision improved to 81.8%. Also, Vietnamese translation results in fewer suspicious issues because it mostly keeps named entities in the original English form. Given this result, **RQ-3** can be answered. First, it’s possible to apply the workflow to a language that the author has no knowledge of. The precision in Vietnamese is acceptable using a specifically tuned threshold.

TABLE V
PRECISION OF ERROR DETECTION – VIETNAMESE

Precision	Proposed method - Threshold adopted from Chinese	Proposed method - Learnt threshold
Google - Combined	21/33 (63.6%)	18/22 (**81.8%**)

VI. DISCUSSION

A. *Types of Error Found*

The types of error our method can detect agree with other research like [6], and [17]. There are 5 types in total, namely miss-translation, under-translation, over-translation, incorrect modification and under-logic. Additionally, a new type of translation error is detected because of the introduction of NER. We define this as an over-inference error, which is discussed in detail.

TABLE VI
EXAMPLE: OVER TRANSLATION

Source	According to data from the Bureau of Labor Statistics.
Target	根据美国劳工统计局的数据
Target Meaning	According to data from the U.S. Bureau of Labor Statistics.
Back-translation	According to data from the US Bureau of Labor Statistics.

1) Over-inference in Named Entities: One notable source of error comes from named-entity-related CRTIs. We have observed an exposure bias on named entities in neural machine translation. It is biased towards the most influential country and famous people so that some names will be replaced by the name of famous people, some places will be entitled to be in an influential country. As seen in Table VI, it unnecessarily adds the word “US” to the “Bureau of Labor Statistics”. Meanwhile, the example in Table VII replaces the word “bloc” with “EU” due to the translation engine relating the date and subject of the date and subject of the sentence to think that the sentence is talking about Brexit. Such over-inference errors may unfairly relate indirect evidence within a single sentence to real-world and recent events. However, the context of these sentences may not be sufficient to infer. For example, it could be completely fictional, taken from a novel. Or, it could be a historical event rather than a recent one. It is important for translation engines to check for the level of inference and avoid adding unnecessary information to ensure accurate translations.

TABLE VII
EXAMPLE: OVER-INFERENCE

Source	The chances of the country crashing out of the bloc without a transitional deal on March 29 increase.
Target	该国在 3 月 29 日没有达成过渡性协议的情况下退出欧盟的可能性就会增加。
Target Meaning	The chances of the country leaving the EU without a transitional deal on March 29 increase.
Back-translation	The chances of the country leaving EU without a transitional deal on March 29 increase.

B. Limitations and Threats to Validity

The are three limitations of our current method. First of all, despite our method having a higher accuracy of error detection, our method may introduce more false negatives compared to the baseline approach [7]. Second, our method only identifies translation errors based on the correctness of nouns. There could be other reasons that make a translation inappropriate, which is outside the scope of our study. Third, without a comprehensive study on more target languages and translation engines, it is not guaranteed that our method can be extended to all languages.

VII. Conclusion and Future Works

A. Conclusion

In this paper, we implemented a machine translation testing method using metamorphic testing. We introduced back-translation and created the concept of contextual referentially transparent inputs (CRTI). We assume that CRTIs should be translated to the same result given a short context. We then extract a metamorphic pair for testing between the CRTI in the original sentence and the back-translation, which only operates on a single language. As a result, we reported suspicious translation errors without the knowledge of a given target language with an improved testing accuracy compared to the baseline method in [7]. We translated 200 sentences using Google Translate. Our method reports 57 suspicious issues with a precision of 74% in Chinese translation and 22 suspicious issues with a precision of 82% in Vietnamese translation. Even though the hyper-parameter needs to be tuned to achieve better testing accuracy, the workflow fits other languages.

B. Future Works

There are two directions to strengthen further or extend this research. First of all, it is important to extend the scope of our proposed method to test translation to more languages, especially to low-resource language, where no constituency parser or other NLP tool is available. In that way, We can better claim our contribution of introducing back-translation to test machine translation.

Also, a new study area to explore the bias in contextual representation towards high-influence countries and famous people is worth looking into. Several directions can be explored, from reducing bias in the training process, benchmarking the bias that we described, or repairing this kind of bias in translation results.

References

[1] Dorothée Behr. Assessing the use of back translation: The shortcomings of back translation as a quality testing method. *International Journal of Social Research Methodology*, 20(6):573–584, 2017.

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[3] Tsong Y Chen, Shing C Cheung, and Shiu Ming Yiu. Metamorphic testing: a new approach for generating next test cases. *arXiv preprint arXiv:2002.12543*, 2020.

[4] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018.

[5] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[6] Pinjia He, Clara Meister, and Zhendong Su. Structure-invariant testing for machine translation. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 961–973. IEEE, 2020.

[7] Pinjia He, Clara Meister, and Zhendong Su. Testing machine translation via referential transparency. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 410–422. IEEE, 2021.

[8] D. Jurafsky, J.H. Martin, P. Norvig, and S. Russell. *Speech and Language Processing*. Pearson Education, 2014.

[9] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[10] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[11] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[12] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[13] Stefan Schweter and Alan Akbik. Flert: Document-level features for named entity recognition, 2020.

[14] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

[15] Harald Søndergaard and Peter Sestoft. Referential transparency, definiteness and unfoldability. *Acta Informatica*, 27(6):505–517, 1990.

[16] Liqun Sun and Zhi Quan Zhou. Metamorphic testing for machine translations: Mt4mt. In *2018 25th Australasian Software Engineering Conference (ASWEC)*, pages 96–100. IEEE, 2018.

[17] Zeyu Sun, Jie M Zhang, Mark Harman, Mike Papadakis, and Lu Zhang. Automatic testing and improvement of machine translation. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 974–985, 2020.

[18] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 48(1):1–36, 2022.

[19] Zhi Quan Zhou, Liqun Sun, Tsong Yueh Chen, and Dave Towey. Metamorphic relations for enhancing system understanding and use. *IEEE Transactions on Software Engineering*, 46(10):1120–1154, 2018.