# SemMT: A Semantic-based Testing Approach for Machine Translation Systems

JIALUN CAO, The Hong Kong University of Science and Technology, China

MEIZINIU LI, The Hong Kong University of Science and Technology, China

YETING LI, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, China

MING WEN, Huazhong University of Science and Technology, China

SHING-CHI CHEUNG*, The Hong Kong University of Science and Technology, China

HAIMING CHEN, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, China

Machine translation has wide applications in daily life. In mission-critical applications such as translating official documents, incorrect translation can have unpleasant or sometimes catastrophic consequences. This motivates recent research on the testing methodologies for machine translation systems. Existing methodologies mostly rely on metamorphic relations designed at the textual level (e.g., Levenshtein distance) or syntactic level (e.g., distance between grammar structures) to determine the correctness of translation results. However, these metamorphic relations do not consider whether the original and the translated sentences have the same meaning (i.e., semantic similarity). To address this problem, in this paper, we propose SemMT, an automatic testing approach for machine translation systems based on semantic similarity checking. SemMT applies round-trip translation and measures the semantic similarity between the original and the translated sentences. Our insight is that the semantics concerning logical relations and quantifiers in sentences can be captured by regular expressions (or deterministic finite automata) where efficient semantic equivalence/similarity checking algorithms can be applied. Leveraging the insight, we propose three semantic similarity metrics and implement them in SemMT. We compared SemMT with related state-of-the-art testing techniques, demonstrating the effectiveness of mistranslation detection. The experiment results show that SemMT outperforms existing metrics, achieving an increase of 34.2% and 15.4% on accuracy and F-Score, respectively. We also study the possibility of further enhancing the performance by combining various metrics. Finally, we discuss a solution to locate the suspicious trip in round-trip translation, which provides hints for bug diagnosis.

Additional Key Words and Phrases: machine translation, metamorphic testing, testing, semantic equivalent, semantic similarity

**ACM Reference Format:**

Jialun CAO, Meiziniu LI, Yeting LI, Ming Wen, Shing-Chi Cheung*, and Haiming Chen. 2020. SemMT: A Semantic-based Testing Approach for Machine Translation Systems . In *ACM Trans. Softw. Eng. Methodol.* ACM, New York, NY, USA, 35 pages. https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

## 1 INTRODUCTION

Machine translation systems, which provide automatic translation for text and speech from a *source language* to another *target language*, are widely used in daily lives [34, 35]. However, machine translation systems can give incorrect or

* Corresponding author.

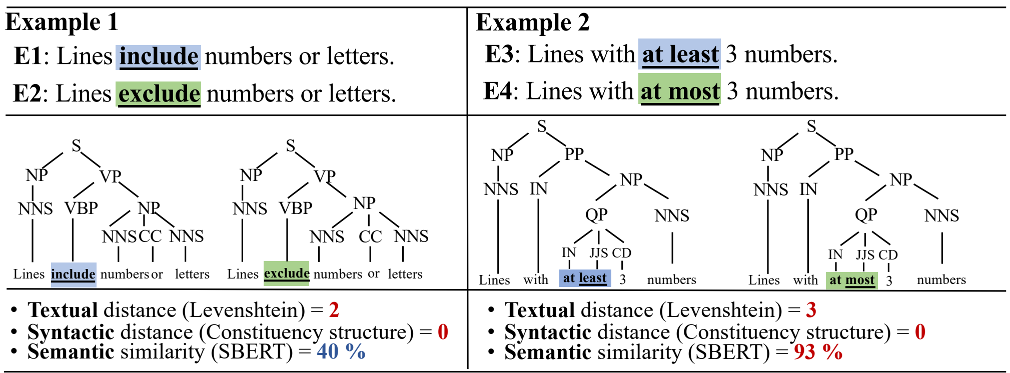



Fig. 1. **Motivating Examples.** Examples of semantic differences that cannot be captured by existing textual, syntactic or semantic similarity metrics.

inappropriate translations, which could lead to harmful consequences such as embarrassment, misunderstanding, financial loss, or political conflicts [23, 49, 55, 56]. This motivates the research on testing methodologies to assure the quality of machine translation software.

Recent works [32, 35, 60, 77, 83, 92] on machine translation testing mostly adopt the metamorphic testing approach [16, 17]. The intuition is that similar sentences should be similarly translated [35, 77, 92], while sentences with different meanings should not have the same translations [32]. Mistranslations are detected by examining the textual (e.g., Levenshtein distance) or syntactic similarities between the original sentence and the translated sentence. However, a close textual or syntactic distance between two sentences does not necessarily imply close semantic meaning, and thus cannot guarantee the correctness of translation. For example, in Fig. 1 (Example 1), the verbs "include" and "exclude" are logically opposite, though their textual/syntactic distances are the same or close. Consequently, while the textual and syntactic distances between the two sentences are close, they deliver opposite semantic meanings. Using semantic-based metrics such as the state-of-the-art SBERT [67], the similarity measured over the two sentences in Example 1 is low (i.e., only 40%), which indicates that the differences can be more precisely captured in this case. However, SBERT does not necessarily perform well in all cases. In the Example 2 of Fig. 1, SBERT gives a high semantic similarity (93%) although the phrase "at least" and "at most" in E3 and E4 are two opposite quantifiers. The two examples suggest that a better metric to measure sentence semantics is needed.

However, the semantic meaning of sentences is hard to be precisely captured, which exacerbates the challenges of measuring translation correctness. It is non-trivial to judge the semantic equivalence/similarity between two sentences even for human [91]. The judgement made by human can be subjective [60, 92], making the decision varies across different individuals. Besides, the flexibility of natural language makes this problem even more challenging [92]. For example, a token or phrase may have multiple correct translations, in such cases, modern translation software does not perform well [35].

In view of these challenges, we then approach the problem by confining the scope to the semantics that concern quantifiers and logical relations (like examples in Fig. 1), and then test translation on sentences with such semantics. For what it may worth to mention, sentences with quantifiers [62, 75] and logical relations-[18, 33, 76] are pervasive in the daily life and of central importance in linguistic semantics [6, 61, 62, 78]. According to our investigation (see Section 2), in every six sentences, there is one sentence involving quantifier or logical relations, reflecting the prevalent use of such

| | |
|---|---|
| **Source 1** | The share of symptomatic children who lose their lives to the virus in China has been estimated as 1 in 25,000, which is **30 times less than** of the middle aged and **3,000 times less than** the **elderly**. |
| **Translation 1 (Google)** <br> **Quantifier-related** | 据估计，在中国，因病毒而丧生的有症状儿童的比例为25,000人中的1人，这是中年人的**30倍**，而**中年人**的**3,000倍**。<br>(Meaning: It is estimated that the proportion of symptomatic children killed by the virus in China is 1 in 25,000, which is **30 times that of** middle-aged people and **3,000 times** that of **middle-aged people**.) |
| **Source 2** | Diarrheal disease was the cause of every tenth child's death in 2017 – more than **half a million of the 5.4** million children that died in 2017 died from diarrheal disease. |
| **Translation 2 (Google)** <br> **Quantifier-related** | 腹泻病是2017年每十分之一儿童死亡的原因-在2017年死亡的**540万**儿童中，**超过一半**死于腹泻病。<br>(Meaning: Diarrheal disease was the cause of every tenth child death in 2017-of the **5.4 million** children who died in 2017, **more than half** were due to diarrheal disease.) |
| **Source 3** | While 19.6% of household **not containing only young people** live in the northwest sub-area, only 4.8% of young person only and 10% of young person with child live in this area. |
| **Translation 3 (Google)** <br> **Logic-related** | 虽然只有19.6%的**仅包含**年轻人的家庭居住在 “西北” 子区域，但只有4.8%的年轻人和10%的有孩子的年轻人居住在该区域。<br>(Meaning: Although only 19.6% of families **containing only young people** live in the "Northwest" subregion, only 4.8% of young people and 10% of young people with children live in this area. |

Fig. 2. Examples of Mistranslation of Sentences with Quantifiers and Logical Relations.

sentences in daily life. Since they mainly specify the quantity and logical meaning of the objects, and are commonly found in legal contracts, financial statements, healthcare reports, product instructions and so on, the mistranslation regarding them can cause misunderstanding or severe consequences. For example, in Fig. 2 [1], the quantity "30 times less than" in the first sentence (i.e., Source 1) is mistranslated to "30 times" by Google translator (i.e., Translation 1). The translation mistakenly reports the children's morbidity and mortality. It can result in unnecessary public panic. Note that the original sentence is taken from a brief policy launched recently[65] on the impact of COVID-19 on children. Similarly, in the second example excerpted from a report on Diseases and Mortality [22], the number of children who suffered from disease is exaggerated after translation. Such mistranslations can cause severe consequences for the public. Apart from mistranslation of numbers, mistranslation of logical relations are also common. The third example (i.e., Source 3) taken from [48] in Fig. 2 illustrates the mistranslation of logical relation, conveying opposite meaning (i.e., exclusion and inclusion) before and after translation, resulting in a wrong cognition of the household situation in northwest. Earlier work [32] has proposed to detect mistranslations based on an intuition that semantically different sentences should not have the same translation results. However, such detection technique may not be effective because many mistranslated sentences and their semantically mutated ones do have different translation results. For example, the sentence given by Source 3 in Fig 2 is mistranslated to Chinese by Google. Yet, Google gives a different Chinese translation when the word "not" is removed from the sentence, escaping such detection of mistranslation based on the same translation results. The same situation occurs in the translation of the other two sentences in Fig 2 (i.e., replacing "less" with "more" in Source 1, and replacing reversely in Source 2). As such, it is important to design an effective test methodology for the translation of sentences that contain quantifiers and logical relations. Such testing methodologies have not been studied in prior work.

[1] The translation results were collected on December 16, 2020 on Google Translator.

In this paper, we propose SemMT, an automatic testing approach for machine translation systems based on semantic similarity checking of the concerned quantifiers and logical relations. The insight of SemMT is that the semantics concerning logical relations and quantifiers in sentences can be captured by regular expressions (or deterministic finite automata) where efficient equivalence/similarity checking algorithms are available. To be more specific, SemMT addresses the difficulty of capturing semantics similarities precisely and detecting semantic mistranslation on translation systems using the following three strategies:

**Transformation to regular expressions.** Since the semantics regarding quantifier and logical relations in sentences can be captured using regular expressions, the core step of SemMT is to transform the sentences to the semantic-equivalent regular expressions. This strategy enables the maximized precise semantic similarity capturing. As a well-explored, -tested and widely-used context-free grammar [12, 24, 80, 82, 90], the semantics over regular expression (abbrev. regex) can be evaluated under a context-free paradigm, enabling us to capture and quantify semantic similarities precisely.

**Precise semantic similarity capturing.** Based on the above strategy, the semantic similarities over regular expressions then can be captured and quantified via well-established algorithms in formal language [37, 58, 89]. If the semantic similarity quantified by these algorithms between the source and translated sentences falls beyond the similarity threshold, such translations will be considered as suspicious mistranslations.

**Semantic checking in the same language.** Semantic equivalence can hardly be measured automatically across different languages [86]. Therefore, SemMT performs testing on round-trip translation, which translates a original sentence to another language, and then translates it back. In this way, the back-translated sentence and the original sentence are in the same language, allowing their semantics to be uniformly measured and compared.

On top of that, we implemented SemMT and compared with the state-of-the-art testing techniques. The experiment results show that SemMT achieves an increase of 23% in terms of F-Score as compared with other similarity measures. SemMT outperforms the state-of-the-art techniques, achieving an improvement of 34.2% in accuracy with comparable amount of mistranlations detected. Besides, SemMT can detect 213 bugs in Google Translator as compared with 173 detected by other similarity metrics. We also study the possibilities of improving accuracy and F-Score by combining different similarity metrics. Note that a mistranlation detected by SemMT may occur at the forward translation or the backward translation [60, 74, 92]. We discuss a method to locate the translation in which the bug resides. In addition, we discuss the types of bugs detected by SemMT.

To sum up, this paper addresses the oracle problem in testing machine translation systems with respect to their semantics. Specifically, it makes the following four main contributions:

- We propose SemMT, a semantic-based machine translation testing framework. Specifically, it captures the semantics of quantifiers and logical relations during translation by semantic-equivalent regular language and detects mistranslation accordingly. To best of our knowledge, it is the first testing methodology proposed for machine translation systems based on the semantic similarity.
- We introduce an approach to determining and quantifying semantic differences. Specifically, we transform natural language to semantic-equivalent regular expressions, and then propose a metric to measure semantic similarities in formal ways.
- The experiment results show that our proposed metrics are more effective than existing similarity metrics in capturing the semantics of sentences that concern quantifiers and logical relations. Using the proposed metric, SemMT outperforms state-of-the-art techniques, achieving an increase of 34.2% in accuracy and 15.4% in F-Score.

- We perform the first study to improve test effectiveness for machine translations via combining multiple similarity metrics. Experiment results show that improvements can be achieved when proper metrics are combined. The experiment data and tool are publicly available [2].

## 2 MOTIVATION

Measuring the semantic difference/similarity of natural languages is still an open problem due to their subjective [60, 92], flexible [92] and context-aware [44, 81] features. We, therefore, confine our measurement to the semantics of quantifiers and logical relations, which can be represented formally by regular expressions. A **quantifier** is a word/phrase that usually goes before a noun to express the quantity of the object. It is traditionally defined using set-theoretic terms in linguistic theories [5, 41, 46, 62, 75]. Commonly used quantifiers include proportional quantifiers (e.g., "some", "a few", "many" and "more than half"), logic quantifiers (e.g., "none"), and quantifiable quantifiers (e.g., "more than half" and "more than 3 times"). We refer to those sentences that contain quantifiers as *quantifier-related sentences*. For **logical relations**, we follow existing works [3, 52, 53, 59, 64, 70] and focus on those semantics expressed in first-order logic such as conjunction, disjunction, negation, inclusive and exclusive relations, such as "X contains Y". We refer to those sentences that contain such logical semantics as *logic-related sentences*.

Since we confine our scope to quantifier- and logic-related sentences, a legitimate question is: *Are quantifier- and logic-related sentences common?* To answer this question, we collected five large-scale corpora (i.e., Europarl [39], CommonCrawl and News available from the Workshop on Machine Translation (WMT) [3], News Commentary Parallel Corpus [79], Financial News from Reuters [26]) which are commonly-used for machine translation and other natural language processing tasks, and analyzed the proportion of sentences that involve quantifiers or logic relations. The corpora that we analyzed cover a broad range from policy, finance to daily news, which indicates that the findings made based on them can be generalized. We followed the methodology of earlier work [1, 62] to identify quantifier-related sentences and logic-related sentences. The list of patterns that we used for the analysis is publicly available [4].

Table 1 shows the statistics of the investigation. Initially, over 4 million sentences were collected. After filtering out those containing less than 10 words (e.g., the sentence "I understand"), the statistics were conducted over 3.7 million sentences. According to Table 1, there are 639,179 (16.96%) sentences that are quantifier- or logic-related. It reflects the popular use of such sentences in daily life. For a further breakdown, 12.52% sentences are quantifier-related, and 5.22% are logic-related respectively.

While not all of these quantifiers and logical relations can be precisely captured by regexes (e.g., "a few" and "many" cannot be quantified by an exact number), we further measured the proportion of quantifiers and logical relations that can be precisely captured by regexes. As revealed by Table 1, 89.6% (=15.19/16.96) of the quantifier- and logic-related sentences in the selected dataset can be captured precisely. A closer examination reveals that 90.8% of the quantifier- and 83.9% of the logic-related sentences can be precisely captured. Such high ratios reflect the fact that quantifier- or logic-related sentences that can be captured precisely are also pervasive, and the focus on such sentences will not largely narrow the scope of application of our approach to detect mistranslation bugs. Besides, we also discuss the situation where the semantics can be captured approximately in §5.3.

[2] https://github.com/SemMT-2021/SemMT
[3] http://www.statmt.org/wmt13/
[4] https://github.com/SemMT-2021/SemMT

Table 1. **Prevalence of Quantifier and Logic-related Sentences in Real-World Corpora.** The table shows the total number of sentences before (#Total) and after (# Filtered) filtering, the number and ratio of sentences with quantifiers (# Qty, % Qty), with logical relations (# Lgc, % Lgc), and with either one (# Related, % Related). Those quantifier or logic-related sentences that can be quantified are labeled with subscript *PRS*.

| Corpora | # Total | # Filtered | Quantity-related | | | | Logic-related | | | | Related | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | # Qty | # $Qty_{PRS}$ | % Qty | % $Qty_{PRS}$ | # Lgc | # $Lgc_{PRS}$ | % Lgc | % $Lgc_{PRS}$ | # Related | # $Related_{PRS}$ | % Related | % $Related_{PRS}$ |
| ReutersNews | 285,964 | 243,664 | 53,453 | 51,627 | 21.94% | 21.19% | 12,061 | 10,394 | 4.95% | 4.27% | 62,119 | 59,107 | 25.49% | 24.26% |
| WMT-News | 14,522 | 12,219 | 2,331 | 2,189 | 19.08% | 17.91% | 507 | 443 | 4.15% | 3.63% | 2,711 | 2,526 | 22.19% | 20.67% |
| CommonCrawl | 1,845,286 | 1,640,632 | 199,945 | 175,989 | 12.19% | 10.73% | 93,572 | 77,653 | 5.70% | 4.73% | 279,598 | 243,909 | 17.04% | 14.87% |
| Commentary | 174,441 | 153,258 | 21,892 | 19,111 | 14.28% | 12.47% | 6,876 | 5,867 | 4.49% | 3.83% | 27,633 | 24,210 | 18.03% | 15.80% |
| Europarl | 1,965,734 | 1,718,060 | 194,110 | 179,657 | 11.30% | 10.46% | 83,730 | 70,744 | 4.87% | 4.12% | 267,118 | 242,436 | 15.55% | 14.11% |
| **Total** | 4,285,947 | 3,767,833 | **471,731** | **428,573** | **12.52%** | **11.37%** | **196,746** | **165,101** | **5.22%** | **4.38%** | **639,179** | **572,188** | **16.96%** | **15.19%** |

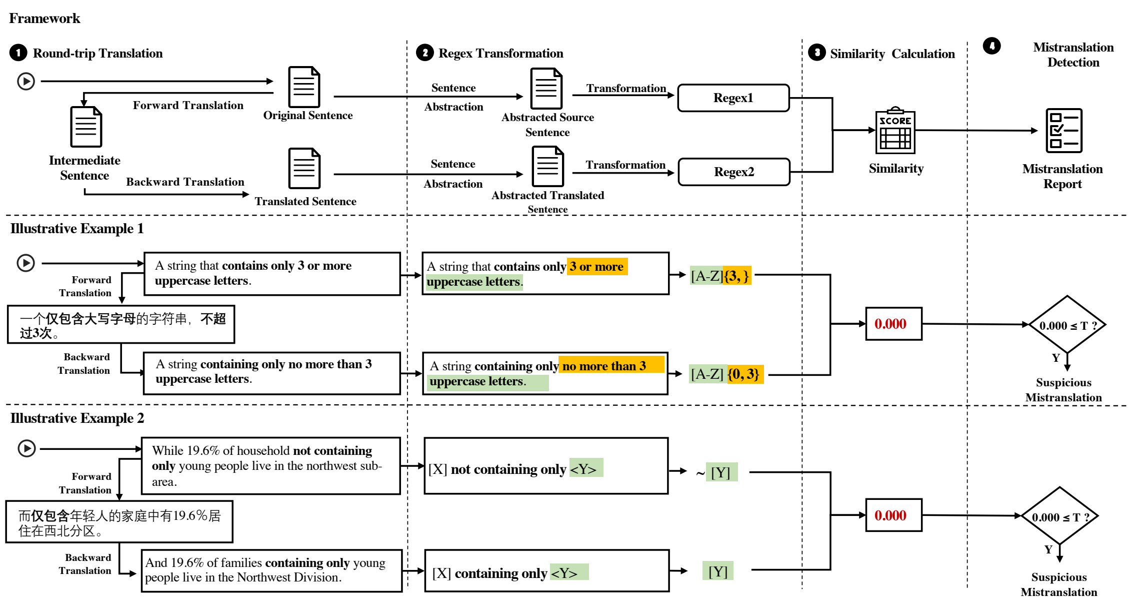



Fig. 3. **Framework and Two Illustrative Examples.** The words highlighted in yellow are quantifiers that modify the object words highlighted in green.

## 3 APPROACH

In this section, we first give an overview of SemMT, followed by the explanation of its methodology.

### 3.1 Overview

Fig. 3 illustrates the framework of SemMT with two illustrative examples. The upper diagram in Fig. 3 shows the general workflow of SemMT. Given the original sentences in one language, SemMT detects mistranslation in four steps: (1) conducting the round-trip translation to collect the intermediate and translated sentences, (2) abstracting and transforming the source and translated sentences into regular expressions using existing tools, (3) calculating the semantic similarity between the regular expressions based on a set of regex-related metrics and (4) detecting mistranslation according to customized thresholds. Finally, the detected mistranslations are reported.

The middle and lower diagrams demonstrate two mistranslations that can be detected by SemMT [5]. The first original sentences is taken from the NL-RX-Synth dataset [47], and the second one is taken from an online housing needs document [48] (same as the third example in Fig. 1). Given the original sentences, the round-trip translation is firstly conducted by the forward translation from the source language (take English as example) to the intermediate language (take Chinese as example), then backward translation to the source language. Secondly, the semantics regarding the quantifiers and logical relations are identified from the source and translated sentences and transformed into the corresponding regular expressions. The purpose of sentence abstraction is to capture the quantifiers and logical relations as precise as possible, and the details of sentence abstraction and regular expression transformation are explained in §3.3.1 and §3.3. After the second step, the semantics of the source and translated sentences can be captured by regular expressions. Take Illustrative Example 1 as an example, the meaning of two English sentences can be described by the corresponding two regular expressions `[A-Z]{3,}` and `[A-Z]{0,3}`, where `[A-Z]` represents an uppercase letter, and quantifiers `{3,}` and `{0,3}` serve as quantifiers "3 or more" and "no more than 3", respectively. Next, semantic similarity is calculated over the two regular expressions by regex-based semantic similarity metrics. In particular, SemMT utilized three semantic similarity metrics to differentiate the semantic meanings. If the similarity is higher than the predefined threshold, the semantic meaning of the original sentence is well-preserved after translation; otherwise, SemMT reports it as a candidate round-trip mistranslation. As shown in the illustrative examples, take DFA-based similarity (details can be found in § 3.4.2) as example, the semantic similarities of two examples are relatively low (0.000) for both translations, SemMT, therefore, reports the source, intermediate and and translated sentences as a potential round-trip mistranslation.

In the following of this section, we will explain the details of the four steps using two illustrative examples, showing how SemMT is able to capture semantic difference and detect mistranslations in round-trip translations. Challenges arise in the measurement of similarity between regexes and the determination of thresholds. We will discuss them in §3.3, §3.4 and §3.5.

### 3.2 Round-Trip Translation

Round trip translation (RTT) is also known as back-and-forth translation. It translates a given text or sentence into an intermediate language (the forward translation), and then translates the result back into the source language (the back translation) [74]. It reflects the general quality of a translation system over longer texts [74] and is a cost-effective choice [54, 60, 92] to derive reference translation automatically.

The benefit of adopting RTT in our methodology is that the semantics of the source and back-translated sentences can be uniformly measured and compared in the same language. The selection of source and intermediate languages are not restricted by our methodology. Yet the selection of language pairs will affect the effectiveness of methodology in two aspects. First, in some languages (e.g., Chinese and Japanese), nouns are the same in both singular and plural forms, while they are in different forms in other languages like English and Russian. The RTT between these two kinds of languages may lose/switch the singularity or plurality information, making the semantics changed. Second, the availability of automated transformation approaches used in the following steps (i.e., the second step in Fig. 3) also needs to be considered. If the automatic approaches of deriving regex from natural language sentences in the source language is unavailable, the workflow is unlikely to proceed automatically. Hence the selection of language

[5]The translation results were collected on December 17, 2020 on Google Translator.

pairs under test should take these two aspects into account. Besides, since SemMT adopts a black-box testing manner, so the translators under test can be either open- or close-sourced.

Nevertheless, one may concern that RTT involves testing two systems instead of one [60, 92] and it is unclear which trip is buggy when a sentence is mistranslated. It subsequently motivates us to explore the possibility to locate the buggy trip automatically , and we discuss one possible solution in Section 5.4.

### 3.3 Regular Expression and Deterministic Finite Automaton Transformation

SemMT is empowered by the recent advances made on the synthesis of regular expressions from natural languages using rule-based [42, 66] and learning-based [47, 58, 89] approaches. In this subsection, we present the key ideas of how these techniques can be applied in the methodology of SemMT. In the following, we adapt two illustrative examples in Fig. 3 as running examples to show how the resulting transformed regexes and deterministic finite automata are derived together with the introduction of the related formalism.

- **Running Example 1 (Original Sentences)**
  - S1: A string that contains only **3 or more** uppercase letters.
  - S2: A string containing only **no more than 3** uppercase letters.
- **Running Example 2 (Original Sentences)**
  - S3: While 19.6% of household **not containing only** young people live in the northwest sub-area.
  - S4: And 19.6% of families **containing only** young people live in the Northwest Division.

#### *3.3.1* ***Sentence Abstraction.***

Sentence abstraction is a preprocessing step conducted on the source and translated sentences before converting them into regular expressions. It helps to focus on semantics that relate to quantifiers and logical relations. In general, the sentence abstraction works in two steps: (1) identify the non-terminal words/tokens that describe quantifiers and logical relations in the sentence, and (2) abstract the terminals or non-terminals that are grouped by the identified non-terminals as abstracted objects, denoted by symbols such as `X` and `Y`. Specifically, the non-terminals and terminals we used are specified and used in existing works [47, 58], and are associated verbalization for both regular expressions and language descriptions. For the first running example, the terminals (e.g., "uppercase letters") and non-terminals (e.g., "contains only", "3 or more", "no more than 3") are specified in [47], and therefore no abstraction is needed. While for the second running example, the terminal nouns/phrases grouped by non-terminals (i.e., "not containing only" and "containing only") are not specified, so they are abstracted and denoted as abstract symbols `X` and `Y`. Hence, after abstraction, the sentences in two running examples are as follows:

- **Running Example 1 (Abstracted Sentences)**
  - S1': A string that contains only **3 or more** uppercase letters.
  - S2': A string containing only **no more than 3** uppercase letters.
- **Running Example 2 (Abstracted Sentences)**
  - S3': `X` **not containing only** `Y`.
  - S4': `X` **containing only** `Y`.

Note that after abstraction, the meaning of sentences in the first example are almost preserved while in the second one, the most semantics are abstracted, remaining only the logical related meaning.

#### *3.3.2* ***Regular Expression (regex).***

Regular expressions are widely used in practice. Let $\Sigma$ be a finite alphabet of symbols, a *word* is a finite sequence of symbols chosen from this alphabet, and the set of all words over $\Sigma$ is denoted by

$\Sigma^*$. For example, `01101` is a *word* from the binary alphabet $\Sigma = \{0, 1\}$. The empty word and the empty set are denoted by $\varepsilon$ and $\emptyset$, respectively. Regexes over $\Sigma$ are defined inductively as follows: $\varepsilon$, $\emptyset$, $a \in \Sigma$, and $[C]$ where $C \subseteq \Sigma$ are regular expressions; for regular expressions $r_1$ and $r_2$, the disjunction $r_1|r_2$, the concatenation $r_1r_2$, the intersection $r_1\&r_2$, the negation $\neg r_1$, and the quantifier $r_1\{m, n\}$ where $m \in \mathbb{N}$, $n \in \mathbb{N} \cup \{\infty\}$, and $m \leq n$ are also regular expressions. Besides, $r?$, $r^*$, $r^+$ and $r\{i\}$ where $i \in \mathbb{N}$ are abbreviations of $r\{0, 1\}$, $r\{0, \infty\}$, $r\{1, \infty\}$ and $r\{i, i\}$, respectively. $r_1\{m, \infty\}$ is often simplified as $r_1\{m, \}$. For the running examples, the sentences are tokenized and fed into the state-of-the-art model [58], resulting in the following regexes:

- **Running Example 1 (Transformed Regexes)**
  - R1: `[A-Z]{3, }`.
  - R2: `[A-Z]{0,3}`.
- **Running Example 2 (Transformed Regexes)**
  - R3: `~[Y]`
  - R4: `[Y]`

where $[A - Z]$ denotes an uppercase letter, the quantifier $\{0, 3\}$ represents the object before it (i.e., [A-Z] in the example) could appear zero to three times. While the quantifier $\{3, \}$ represents the object could appear three or more times. One may notice that after transformation, the abstract symbol `X` in the second running example is not appear in the regex R4. It is because for the non-terminals which describe part-whole relations [52, 53, 70] such as inclusion (e.g., "contain") or exclusion (e.g., "exclude"), the preceding terminal is regarded as the collection while the latter one represents the part/component inside. For example, the sentence "`X` containing only `Y`" (S4') is transformed to `[Y]` where `X` is the collection which includes only the abstract symbol `Y`. Similarly, the terminal "a string" in the first example is also regarded as the collection which is absent in R1 and R2.

#### 3.3.3 *Language of regular expression.*

The term *language* denotes a set of strings chosen from the set of words $\Sigma^*$. Let $L(r)$ be the language of regular expression $r$. $L(r)$ can be defined inductively as follows: $L(\emptyset) = \emptyset$; $L(\varepsilon) = \{\varepsilon\}$; $L(a) = \{a\}$; $L([C]) = C$; $L(r_1|r_2) = L(r_1) \cup L(r_2)$; $L(r_1r_2) = \{vw \mid v \in L(r_1), w \in L(r_2)\}$; $L(r_1\&r_2) = L(r_1) \cap L(r_2)$; $L(\neg r_1) = \{v \mid v \notin L(r_1)\}$; $L(r\{m, n\}) = \bigcup_{m \leqslant i \leqslant n} L(r)^i$.

For the running examples, the languages of the regular expressions in the running examples are:

- **Running Example 1 (Language of Regexes)**
  - L(R1) = $L([A - Z]\{3, \}) = \{AAA, AAB, \ldots, ZZZ, AAAA, \ldots\}$
  - L(R2) = $L([A - Z]\{0, 3\}) = \{\varepsilon, A, B, C, \ldots, ZZZ\}$
- **Running Example 2 (Language of Regexes)**
  - L(R3) = $L(\sim [Y]) = L(\Sigma^*) \setminus \{Y\}$ [6]
  - L(R4) = $L([Y]) = \{Y\}$

In addition, two regular expressions are **equivalent** if they describe the same language. Also, we denote $|L(r)|$ as the *language size* of regex $r$, which is the number of words described by $r$. In the running examples, $|L(R2)|$ and $|L(R4)|$ are $1 + 26 + 26^2 + 26^3 = 18279$ and 1, respectively, while $|L(R1)|$ and $|L(R3)|$ are infinite.

#### 3.3.4 *Deterministic Finite Automaton (DFA).*

A Deterministic Finite Automaton (DFA) is a finite-state machine that accepts or rejects a given string of symbols, by running through a state sequence uniquely determined by the string [37]. It has equivalent expressive power as the corresponding regular expression, which means any regular

[6] We set the alphabet ($\Sigma$) of symbols is set to be letters and numbers.

expression can be converted into a DFA that recognizes the language described, and vice versa. A DFA [37] can be formally defined as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where $Q$ is a finite set of *states*, $\Sigma$ is an *alphabet*, $\delta : Q \times \Sigma \longrightarrow Q$ is the *transition function*, $q_0 \in Q$ is the *start state*, and $F \subseteq Q$ is the *set of accept states*. In Fig. 4, we visualized the DFAs of two regular expressions ($R_1$ and $R_2$) for better illustration.

In addition, according to §3.3.3, each regular language can be expressed by a regular expression. There exists a unique minimal automaton that accepts the given regular language in a minimum number of states. This minimal automaton is known as a minimal DFA. A minimal DFA is guaranteed to have a regular expression which is semantically equivalent to the minimal DFA [89]. For example, the illustration of the DFAs of the regular expressions $R1$ and $R2$ in §3.3.2 are shown in Fig. 4.

#### 3.3.5 ***Approximated Semantic Transformation***.

In the two running examples above, the meaning of quantifiers such as "3 or more" and "no more than 3" can be precisely captured with the quantifiers in regex `{3,}` and `{0,3}`, respectively. However, as mentioned in §2, some quantifiers and logical relations such as "a few" and "many" can only be approximately captured by regexes or DFAs. Hence, we give an example to illustrate how SemMT can be adapted to handle this situation.

Take the following sentence as an example:

- S5: The U.S. contains **a few** states which choose to have the judges in the state courts serve for life terms. [7].

The vague quantifier "a few" is used with plural countable nouns. It emphasizes a small number of objects. To transform this sentence into regex, we first process it via sentence abstraction. The resulting abstracted sentence is :

- S5': [X] contains **a few** [Y].

where the abstract symbol `[Y]` represents the "states which choose to have the judges on the state's courts serve for life terms", and `X`, abstracted from the words "The U.S.", represents the collection which includes only the abstract symbol `Y`.

Then, to approximate the semantics of S5 by regex, either over- and under-approximation can be applied. Specifically, the over-approximation enlarges the number of countable objects compared with the original number, while under-approximation underestimates the amount. For the example S5, two possible approximated sentences are as follows [8]:

- $S5_O$: [X] contains **at least three** [Y].
- $S5_U$: [X] contains **at most three** [Y].

which can be transformed into the following regexes:

- $R5_O$: `Y{3, }`
- $R5_U$: `Y{1,3}`

where the quantifiers `{3,}` and `{1,3}` in $R5_O$ and $R5_U$ prescribe the symbol `Y` could appears more than three times and once to three times, respectively.

By doing so, the semantics of sentences like S5 can be approximately captured by regexes, then the semantic similarities can be calculated on top of them, hence the mistranslation can be calculated accordingly. However, the approximated semantics may affect the effectiveness of mistranslation detection to some extent, so we also discussed such influence in §5.3. Note that the conversion of sentences to regexes is a major research problem in natural language processing. We do not make contribution to such conversion in SemMT but leverage existing conversion techniques.

[7]This sentence is adapted from https://en.wikipedia.org/wiki/U.S._state

[8]Since "a few" is a vague quantifier, there is no specific number that everyone would agree what it actually means. In this example, we quantify "a few" as "at least 3" following a common usage of "a few".

Since the effectiveness of mistranslation detection can be affected by the quality of the approximated regexes, we evaluate SemMT based on the sentences with quantifiers or logical relations that can be precisely quantified due to the availability of automated tools for transforming such relations. The finding based on the five large-scale natural language corpora in §2 indicates that a large majority (89.6%) of quantifiers and logical relations can be precisely quantified.

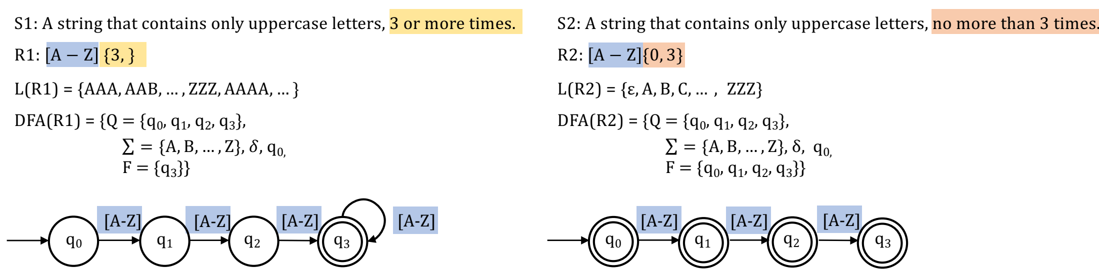



Fig. 4. A Running Example of the Transformation from Natural Language to Regular Expressions, Language of Regular Expressions and Deterministic Finite Automata.

### 3.4 Semantic Similarity Calculation

The semantic differences of sentences involving quantifiers and logical relations cannot be adequately captured by SBERT [67], which is the state-of-the-art metric proposed to measure semantic similarities. As shown in the second example of Fig. 1, SBERT gives nearly 100% semantic similarity between "at least" and "at most". Such weakness is commonly found when SBERT is applied to the NL-RX-Synth [47] dataset used by the natural language processing community for sentences that contain quantifiers and logical relations. We will discuss it in more detail in our experiment. This motivates us to develop new metrics based on regexes and DFAs to better measure the semantic similarities of sentences involving quantifiers and logical relations.

*3.4.1* ***Regex-based Similarity (SemMT-R)***. Considering the transformation rules of regular expression, the similarity between two regexes can be calculated by their Levenshtein distance. Specifically:

$$S_{REG} = 1 - \frac{D_L(r1, r2)}{Max(len(r1), len(r2))} \tag{1}$$

where $D_L(r1, r2)$ is a function that computes the Levenshtein distance between regular expressions $r1$ and $r2$. Note that in Levenshtein distance calculation, we followed the convention to count terminals (e.g., $[a - z]$ which denotes an arbitrary lower-case letter, and $[0 - 9]$ which represents an arbitrary number from 0 to 9) specified in [47] as distance one. Hence, for the running examples, the regex-based similarities are:

- **Running Example 1 (SemMT-R Similarity)**

$$S_{REG}(R1,\ R2) = 1 - 3/6 = 0.500$$

- **Running Example 2 (SemMT-R Similarity)**

$$S_{REG}(R3,\ R4) = 1 - 1/4 = 0.750$$

Though efficient, this regex-based similarity may fail to capture the semantic difference in some situations [12, 21]. For example, R3 and R4 have semantically opposite meaning, while the Levenshtein distance between them is only 1. Hence, we further explore the measurement of semantic similarity using DFAs, which can capture the languages described by the regexes.

#### 3.4.2 *DFA-based Similarity (SemMT-D)*.

An alternative to measure similarity between two regexes is to evaluate the Jaccard similarity between their regular languages. This approach can be effective in finding similarity between two finite sets of data, drawn from various application domains [4, 21, 29]. Specifically, given two regexes, one can construct two corresponding semantic equivalent minimal DFAs, then calculate the Jaccard similarity using the following equation [21, 37]:

$$S_{DFA}(r1, r2) = \frac{|L(r1) \cap L(r2)|}{|L(r1) \cup L(r2)|} \tag{2}$$

whether $L(r1)$ and $L(r2)$ are languages of regexes $r1$ and $r2$, respectively.

However, an issue with such similarity is that regular languages can be infinite. To address this issue, we adapt ideas from existing works [8, 9, 19, 69] to improve the efficiency. We first define the function $S'_{DFA}(r1, r2, \lambda)$ as follows:

$$S'_{DFA}(r1, r2, \lambda) = \frac{|L(r1)^{\leq\lambda} \cap L(r2)^{\leq\lambda}|}{|L(r1)^{\leq\lambda} \cup L(r2)^{\leq\lambda}|} \tag{3}$$

where $\lambda \in \mathbb{N}$. Formally, for a language $L(r)$, let $|L(r)^{\leq\lambda}|$ denote the number of words in $L(r)$ of length at most $\lambda$. Then, we reduce the calculating function $S_{DFA}(r1, r2)$ to calculate the limit of function $S'_{DFA}(r1, r2, \lambda)$, as shown below:

$$S_{DFA}(r1, r2) = \lim_{\lambda \to +\infty} S'_{DFA}(r1, r2, \lambda) \tag{4}$$

Finally, the calculation proceeds iteratively until the solution converges to a preset threshold. Note that the threshold is customized to balance the efficiency and effectiveness. In our evaluation, we set the threshold as 0.001.

So for the running examples, the DFA similarities between two groups of regexes are as follows:

- **Running Example 1 (SemMT-D Similarity)**

$$S_{DFA}(R1, R2) = \lim_{\lambda \to +\infty} S'_{DFA}(R1, R2, \lambda) \approx S'_{DFA}(R1, R2, 6) = \frac{|L(R1)^{\leq 6} \cap L(R2)^{\leq 6}|}{|L(R1)^{\leq 6} \cup L(R2)^{\leq 6}|}$$
$$= \frac{26^3}{1 + 26 + 26^2 + \cdots + 26^6} = \frac{17576}{321272407} \approx 0.000$$

- **Running Example 2 (SemMT-D Similarity)**

$$S_{DFA}(R3, R4) = \lim_{\lambda \to +\infty} S'_{DFA}(R3, R4, \lambda) \approx S'_{DFA}(R3, R4, 1) = \frac{|L(R3)^{\leq 1} \cap L(R4)^{\leq 1}|}{|L(R3)^{\leq 1} \cup L(R4)^{\leq 1}|} = 0.000$$

where 6 and 1 are the resulting $\lambda$s given that the limit of the function $S'_{DFA}(r1, r2, \lambda)$ can approximate the DFA similarities over the infinite regular languages.

#### 3.4.3 *Hybrid Similarity (SemMT-H)*.

The above described SemMT-R and SemMT-D methods have their own advantages in the measurement of semantic similarities. Regex-based similarity measures the semantics similarity between two regexes in the textual level, while SemMT-D similarity measures it from the perspective of language set. Hence, we propose a hybrid metric to enjoy both advantages by combining SemMT-R and SemMT-D with customized weights.

The hybrid similarity is calculated by the following equation:

$$S_{HYB}(r1, r2) = K \cdot S_{REG}(r1, r2) + (1 - K) \cdot S_{DFA}(r1, r2) \tag{5}$$

where $K$ is a customized parameter which adjusts the balance of REG- and DFA-based similarity metrics. For example, if $K$ is set to be 0.5, the hybrid similarity of the running examples are calculated as follow:

- **Running Example 1 (SemMT-H Similarity)**

$$S_{HYB}(R1, R2)_{K=0.5} = K \cdot S_{REG}(R1, R2) + (1 - K) \cdot S_{DFA}(R1, R2) = 0.250$$

- **Running Example 2 (SemMT-H Similarity)**

$$S_{HYB}(R3, R4)_{K=0.5} = K \cdot S_{REG}(R3, R4) + (1 - K) \cdot S_{DFA}(R3, R4) = 0.375$$

Note that the selection of different values for parameter K can influence the effectiveness of SemMT-H similarity. The larger the parameter K is, the SemMT-H similarity $S_{HYB}$ depends more on SemMT-R similarity ($S_{REG}$), or vice versa. To further demonstrate the influence of K on the effectiveness of HYB-based similarity, we conducted an experiment to discuss this issue in §5.5.

### 3.5 Mistranslation Detection

Using one of the above regex-based metrics, the semantic similarity between the source and back-translated sentences can be calculated. We then decide whether a given translated sentence is far enough from the source to indicate the presence of a mistranslation. To do so, following an existing study TransRepair [77], we first conduct the threshold selection through fine-level granularity enumeration guided by the target evaluation metrics (i.e., F-Score), and then detect mistranslations based on the similarity threshold which achieves the optimum performance. Finally, we identify those sentences whose semantics are less similar to the original sentence than the threshold as suspicious mistranslated sentences. The intuition of guiding by F-Score is that it can better balance precision and recall. Besides, the threshold can be customized since the user may prioritize minimizing false positives or maximizing recall depending on their goals. In §4, we show the trade-offs for different threshold values.

If the similarity between the source and back-translated sentences is less than the threshold, SemMT will report it as a suspicious mistranslation. Considering the fact that a forward mistranslation can hardly lead to a reasonable backward translation [74], some may believe it is enough to only report the forward trip translation (i.e., the pair of source and intermediate sentences). However, according to our analysis and investigation, there are 26% mistranslations detected by SemMT are introduced in the backward trip (§5.4). Therefore, when the similarity falls beyond the threshold, SemMT will report the source, intermediate and back-translated sentences together.

## 4 EXPERIMENTS

This section reports the effectiveness of SemMT by studying the following three research questions (RQs):

- **RQ1: How effective is SemMT in finding buggy translations?** We evaluated the effectiveness of SemMT in terms of accuracy, F-Score, precision and recall compared with other semantic and non-semantic similarity metrics. We also quantified their capabilities in distinguishing buggy and correct translations.
- **RQ2: Can SemMT outperform the state-of-the-art works?** We evaluated the performance using the number of issues detected, precision, recall and F-score for each work.

- **RQ3: Can SemMT's effectiveness be improved by combining different similarity metrics?** Different metrics tend to evaluate similarity from diverse aspects, so we explored combinations of similarities to see whether they are mutually-improved on the effectiveness.
- **RQ4: What is the applicability of SemMT?** The applicability of a testing framework is also critical to measure its practical usefulness. So we further investigated the applicability of our framework, i.e., whether it can achieve similar effectiveness on different translators, language pairs and datasets.

### 4.1 Experiment Setup

We implemented SemMT in Python, and conducted experiments on a machine powered by one Intel i7-8700K@3.7GHz CPU that supports 6 cores, Nvidia GeForce Titan V 12GB VRAM, 64GB memory and Dual Tesla M60 8GB VRAM. The parameter $K$ for SemMT-H is set to be 0.5 on the purpose of striking the balance between them. All experiment results have been released for validation.[9]

**Dataset.** Since we mainly focus on the semantics of quantifier and logical relations, the dataset selected should be consistent of sentences conveying such semantics, which is non-trivial for the evaluation. In this study, we adopted two benchmarks, NL-RX-Synth [47] and KB13 [47, 58] as the source of test inputs for SemMT. The two benchmarks are frequently-used [14, 47, 58] in generating regular expressions for natural language description tasks. In these datasets, the words/tokens can be represented as semantic-equivalent alphabet of symbols, quantities, logic relations or quantitative modifiers. For example, the word "numbers" can be represented by the set of digits (`[0-9]`), and the semantic relations "A or B" can be represented as the disjunction between them (`A | B`). In total, NL-RX-Synth consists of 10,000 pairs of English sentences and the corresponding regular expressions. The average character number of sentences in NL-RX-Synth dataset is 66.36, and 40.61 in KB13. For the first three RQs, the test inputs were randomly selected from NL-RX-Synth, while for RQ4, the test inputs were randomly selected from KB13.

**Labeling.** The output of SemMT is a list of suspicious issues, each of which consists of the original sentence, intermediate translation and target translation. Two of the authors inspected all the results separately and discussed all the inconsistent labels until convergence. Besides labeling whether the RTT is correct or not, we also labeled whether the forward and backward translations are correct. In addition, to compare with existing the state-of-the-art approaches, we also labeled the issues reported by each of them. The output issues of existing approaches are a list of suspicious issues including source, intermediate and back-translated sentences.

**Model Training.** We adopted the state-of-the-art regex synthesis work [58] to transform natural language to regex. which used reinforcement learning to train a sequence-to-sequence model. In addition, to make it better fit into our work, we proceeded the grammar checking and data augmentation to the original NL-RX-Synth dataset in order to improve accuracy and enlarge vocabulary size. Specifically, to augment data, we selected one fourth of the sentences and then replace with synonyms. Note that a manual checking of the replaced synonyms is necessary because we need to ensure the semantic meaning is maintained after the replacement. We also confirmed with a linguist if the semantics are preserved after the synonym substitution. After augmentation, 13,588 sentences were obtained with an average vocabulary size of 123. We then split the training, validation and testing set by 80%, 5%, 15% respectively. The model was trained for 30 epochs in total, achieving 90.93% accuracy on the test set. To answer RQ4, we also perform training

[9] https://github.com/SemMT-2021/SemMT

on the KB13 dataset using a process similar to that on the NL-RX-Synth dataset. The resulting model can achieve the accuracy of 77.67% on the test set. [10]

**Similarity Metrics.** Various similarities have been used by recent works to detect translation errors [32, 35, 77] and estimate translation quality [15, 45, 50, 51, 51, 54, 57, 63, 81]. We selected three syntactic-based similarities that are recently used by related works [35, 77] and one state-of-the-art semantic-based similarity [67] as baselines to compare the accuracy, precision and recall of bug detection with the three similarities supported by SemMT.

- **Levenshtein-based similarity (LEVEN).** It is a way of quantifying how dissimilar two strings are by calculating Levenshtein distance (a.k.a., Edit distance), i.e., counting the minimum number of operations required to transform one string into the other [68] and normalizing it in the same way as [31, 77, 87]
- **Dependency-based similarity (DEP).** Dependency relations describe the direct relationships [13] of strings. We evaluated the distance between two sets of dependency relations by summing up the absolute difference in the number of each type of dependency relations as described in a recent work [35].
- **BLEU-based similarity (BLEU).** The BLEU (BiLingual Evaluation Under study) [57] aims to automatically evaluate machine translation quality by checking the correspondence between the output of machines and that of humans [20, 57]. For one target sentence, the score of BLEU is calculated by comparing it to a set of good quality reference translations. The details of BLEU can be found in [57].
- **SBERT similarity (SBERT).** We also used the state-of-the-art sentence-level semantic approach, SBERT [67], a refined version of BERT [25], as a baseline for semantic similarity metric in our evaluation, because it performs the best in RTT quality estimation as [54] suggests.

**Comparisons.** In the experiment, we compared with SIT [35], TransRepair [77] and PatInv [32]. These works utilize different methods to generate mutants, then detect inconsistencies according to different metrics. In particular, SIT [35] generated syntactically equivalent mutants using BERT by replacing noun and adjective words in sentences. Similarly, TransRepair [77] generated mutants by replacing nouns, adjective words and numbers with their synonyms and further construct the word-mutant pair. For comparison, we implemented these works to generate mutants either using their release source code [35] or by carefully following the explanations in their paper [32, 77]. We adjusted the parameters for these works following the original strategies published in their papers based on our dataset. TransRepair [77] used 0.9 as the minimum cosine similarity of word embeddings to generate word pairs. In our experiment, we lowered the threshold to 0.8 in order to generate sufficient number of pairs. For SemMT, since it is not originally designed for mutant generation, we adapted similar mutant generation processes in the baselines by replacing nouns, numbers and relational adverbs with their synonyms. We also manually validate the generated mutants to preserve the semantics after synonym substitution. The manual checking process was under the guidance of a linguist to reduce the threat to the reliability of experimental results that may be introduce. and removed semantically-changed mutants by manual check. For each seed, SemMT generated up to two mutants.

Note that since SemMT adopts the RTT paradigm, the reported issues may be caused by either the forward and backward trips. Therefore, we labeled both translation trips, and for fairness, the comparison only focused on the correctness of the forward translation while removing those whose backward translations are incorrect.

**Evaluation metrics.** To evaluate the effectiveness, we adopted accuracy, precision, recall and F1-Score (abbrev. F-Score). Given the number of true positives (TPs, a TP refers to a mistranslated sentence that is reported to be a

[10] Since the accuracy on the KB13 dataset is lower than that on the NL-RX-Synth dataset, which may affect the comparison with the effectiveness between two datasets, two authors manually checked and refined the resulting regexes until convergence for better comparison.

mistranslation), false positives (FPs, a FP refers to a correctly translated sentence that is reported to be a mistranslation), true negatives (TNs, a TN refers to a correctly translated sentence that is not reported to be a mistranslation) and false negatives (FNs, a FN refers to a mistranslated sentence that is not reported to be a mistranslation), the metrics are defined as follow:

- **Accuracy**: the proportion of correctly reported (whether the translation is correct or not) sentences.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

- **Precision**: the proportion of real mistranslations over the reported mistranslations.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

- **Recall**: the ratio of reported mistranslations over all the real mistranslations.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

- **F-Score**: twice of the multiplication of precision and recall divided by the sum of them.

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

### 4.2 RQ1: Effectiveness of SemMT

To evaluate the effectiveness of our SemMT, we randomly sampled 500 sentences from the NL-RX-Synth dataset, applied the round-trip translation and collected the translation results.[11] We then transformed both the original sentences and the round-trip translation results into regular expressions by the trained transformation model, which is described earlier in the experiment setup. The three SemMT similarities (SemMT-R, SemMT-D, SemMT-H) were then calculated based on regular expressions. Other metrics were calculated on the source and target sentences. Worth to mention that in order to evaluate all four evaluation metrics, we labeled all 500 translations following the previously mentioned labeling process.

Table 2. **Effectiveness of Different Similarity Metrics For Bug Detection.** Each entry reports the average/highest score among threshold values ranging from 0 to 1 at step 0.01.

| | Accuracy | Precision | Recall | F-Score |
|---|---|---|---|---|
| LEVEN | 0.52 / 0.56 | 0.53 / 0.75 | 0.42 / 1.00 | 0.40 / 0.70 |
| DEP | 0.47 / 0.55 | 0.41 / 0.54 | 0.34 / 1.00 | 0.31 / 0.70 |
| SBERT | 0.47 / 0.56 | 0.34 / 0.55 | 0.22 / 1.00 | 0.23 / 0.70 |
| BLEU | 0.48 / 0.53 | 0.47 / 0.72 | 0.40 / 1.00 | 0.37 / 0.69 |
| SemMT-R | 0.54 / **0.60** | **0.67** / **1.00** | 0.42 / 1.00 | 0.41 / 0.70 |
| SemMT-D | **0.56** / 0.59 | 0.56 / 0.59 | **0.71** / 1.00 | **0.63** / 0.69 |
| SemMT-H | 0.55 / 0.58 | 0.61 / 1.00 | 0.57 / 1.00 | 0.52 / 0.70 |

[11] The translation results were collected on July 27, 2020 on Google translator.

Table 2 shows the comparison on the average and highest accuracy, recall, precision and F-Score of different similarity metrics against thresholds (from 0.1 to 1.0 with step 0.01). Generally, the three similarity metrics in SemMT perform better than the other similarity metrics regarding all four measurements. Moreover, SemMT-D achieves the highest average recall (0.71) and F-Score (0.63), while the highest value of other similarity metrics (i.e., LEVEN, DEP, SBERT, BLEU) are 29% and 23% lower. Particularly, SemMT-D performs the best among our three similarities and baseline similarities with respect to most of the evaluation metrics.

Since the effectiveness of these metrics are affected by customized threshold, we also illustrated how the performance of these evaluation metrics vary against different thresholds. Specifically, we normalized all similarity values, and set the threshold from 0.0 to 1.0, with step 0.01. In Fig 5, we presented the trends on accuracy, precision, recall and F-Score. Overall, all the three of our similarity metrics (drawn in red) outperform the others in terms of accuracy, precision, recall and F-Score for most of the threshold values. Apart from our three similarities, LEVEN outperforms other similarities at most times. Although SBERT measures semantic similarity, its performance is not as good as expected.

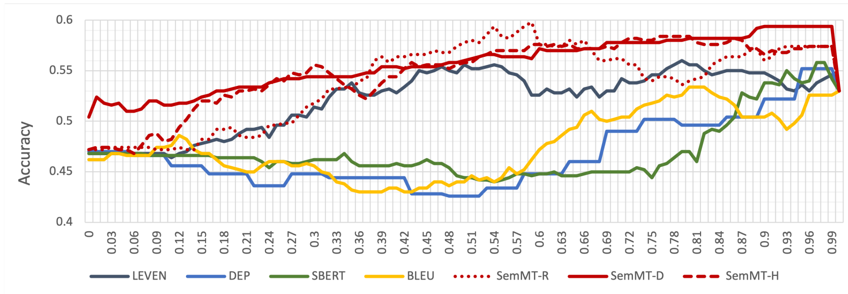
**Summary of Findings Related to RQ1:** The three similarity metrics used in SemMT are effective for mistranslation detection. They outperform the other metrics against almost all threshold values. Specifically, in terms of precision, recall and F-Score, our metrics achieve an increase of 13%, 30% and 23% compared with the highest value achieved by other metrics.

### 4.3 RQ2: Comparison with Existing Works

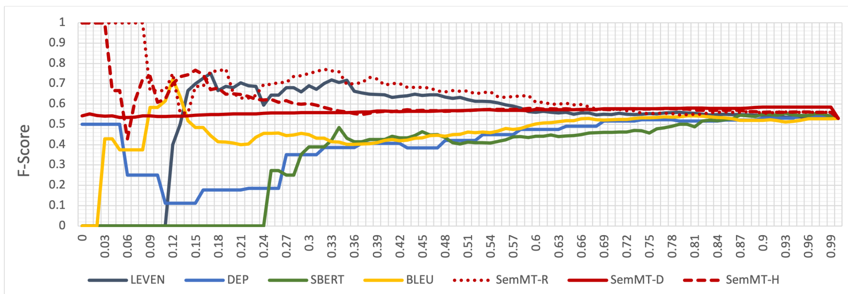
In this section, we compared SemMT with SIT [35], TransRepair [77] and PatInv [32] on mutant generation and the effectiveness of bug detection in terms of accuracy, precision, recall and F-Score under the En-Zh language setting on the Google translator. We randomly selected 200 sentences from the dataset and used them as seeds.

We first generated and filtered mutants in the way as described in the original paper of baselines [32, 35, 77]. The numbers of generated mutants are listed in Table 3. These mutants are generated using their own generation approaches, and after filtering, there are 223 to 452 mutants left for each work. Note that the number of filtered mutants does not necessarily indicate a better capability on bug detection, it is depended on different strategy of mutant generation, the effectiveness is continued to be evaluated. Specifically, we evaluated the effectiveness of each method in terms of the four evaluation metrics. The results are presented in Table 3. The threshold that achieves the optimum performance with respect to F-Score is chosen for each work, as listed in Table 3. Note that the threshold of SIT is the distance between the dependency parse trees, while PatInv is not tuned by the threshold. We choose two most well-performed (with the highest F-Score at 0.82 and 0.82 with thresholds 0.963 and 0.906, respectively) metric values out of the four in TransRepair, which is Levenshtein- (denoted as "ED" in [77]) and BLEU-based method, written as TransRepair(L) and TransRepair(B). For TransRepair(L) and (B), a buggy translation issue is reported when the metric value is smaller than or equal to the selected thresholds, For SIT, a buggy translation issue will be reported when the metric value is larger than or equal to the selected threshold.

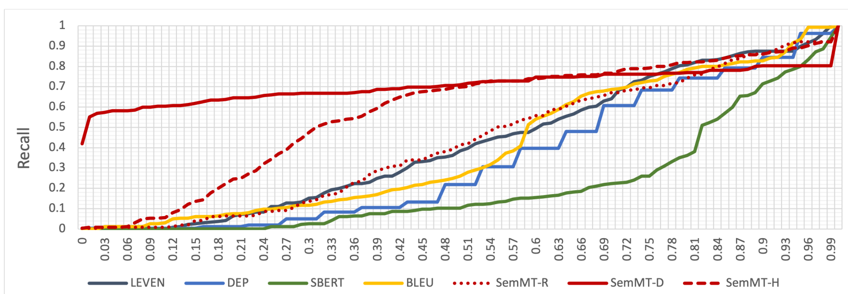
According to Table 3, SemMT achieves the highest accuracy and F-Score compared with existing works with similar number of issues detected. In particular, the highest accuracy achieved by SemMT (74.1% by SemMT-H) is 34.2% higher than the highest accuracy achieved by TransRepair(B) (55.2%), and the highest F-Score (56.3%) achieved by SemMT (SemMT-R) is 15.4% larger than the highest value achieved by SIT (48.8%). In addition, although SIT achieves the best



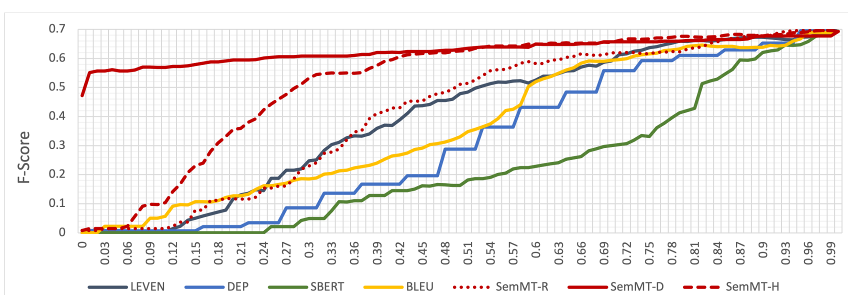

(a) Accuracy




(b) Precision




(c) Recall




(d) F-Score

Fig. 5. Effectiveness of Mistranslation Detection Over Different Similarity Thresholds

Table 3. **Precision and the Number of Mistranslation Using Different Threshold Values.** The symbol '-' denotes the values are inapplicable.

| | # Mutants Filter (Gen) | Thrsh | # Issues | Acc | Recall | Prec | F-Score |
|---|---|---|---|---|---|---|---|
| SIT | 452 (635) | 5 | 124 | 0.497 | **0.932** | 0.331 | 0.488 |
| TransRepair(L) | 223 (869) | 0.96 | 108 | 0.520 | 0.506 | 0.398 | 0.446 |
| TransRepair(B) | 223 (869) | 0.88 | 93 | 0.552 | 0.459 | 0.419 | 0.438 |
| PatInv | 266 (358) | — | 16 | — | — | **0.565** | — |
| SemMT-R | 328 (384) | 0.62 | 105 | 0.730 | 0.710 | 0.466 | **0.563** |
| SemMT-D | 328 (384) | 0.42 | **166** | 0.585 | 0.855 | 0.355 | 0.502 |
| SemMT-H | 328 (384) | 0.32 | 86 | **0.741** | 0.594 | 0.476 | 0.529 |

recall (93.2%), its precision is relatively low (33.1%). The highest precision is achieved by PatInv (56.5%), yet the number of issues identified is small (16). In constrast, SemMT (SemMT-D) achieves a comparatively high recall (85.5%) with 150 (166 - 16 = 150) more issues reported.

Moreover, we plotted the correlation between precision and the number of mistranslation detected in Fig 6 as threshold varies. A translation is regarded as a candidate mistranslation if its similarity is smaller than (for the thresholds of distance, it should be larger than) or equal to the given threshold. The X and Y axes represent the number of issues detected and precision, respectively. The threshold setup for each method is proceeded as follow: For SemMT and TransRepair, we normalized the similarity values to the range of [0,1], and set the threshold from 0.0 to 1.0, with a step of 0.1. The threshold of distance for SIT ranges from 1 to 17. No threshold is required for PatInv. Therefore, there are 11 dots for SemMT-R, SemMT-D, SemMT-H, TransRepair(B) and TransRepair(L), 17 dots for SIT and 1 dot for PathInv. In the figure, we did not denote every threshold for each dots, while the threshold values can be implied - the closer the dot is to the y-axis, the larger the similarity threshold (the smaller the distance threshold).

According to Fig 6, we can see that there is a trade-off between the precision and the number of mistranslation issues reported, i.e., with the similarity threshold increases, the more translations are regarded as mistranslation (i.e., the larger the number of issues reported), while the more false positives may be involved, resulting in a lower precision. To better illustrate such balance, we set precision and the number of mistranslation reported as axes, and the more the dot is closed to the top-right corner, the more the result strikes the balance. Overall, three variants of SemMT (as shown in red) outperform other baseline works for most of the cases with the change of thresholds, followed by SIT and TransRepair(B). On the contrary, PathInv and TransRepair(B) perform less satisfying, with less number of reported mistranslations and lower accuracy.

**Summary of Findings Related to RQ2:** SemMT outperforms the state-of-the-art approaches. In particular, compared with the optimum performance achieved by other works, SemMT achieves an improvement of 34.2% and 15.4% on accuracy and F-Score with similar number of issues detected. Further experiment shows that in general, SemMT can report more mistranslations with higher precisions over the change of thresholds.

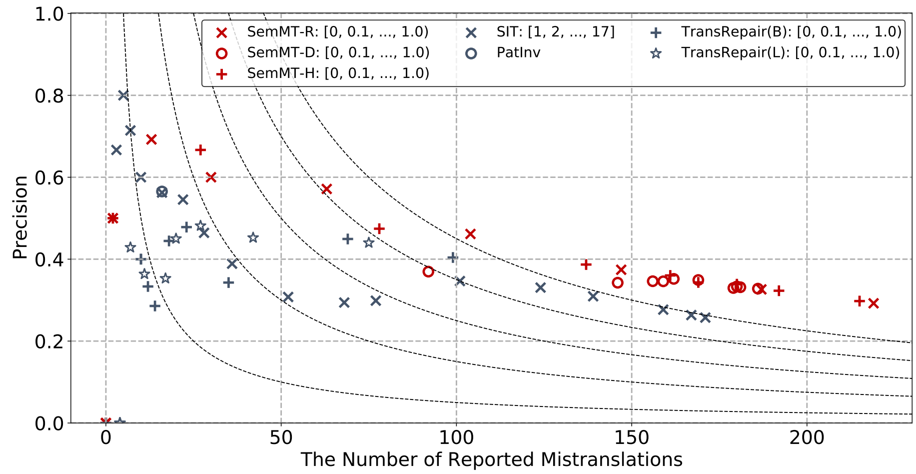



Fig. 6. **Precision and the Number of Mistranslation Reported by SemMT Compared With Other Works.**

### 4.4 RQ3: Can Metrics of SemMT Find Mistranslations That Cannot Be Detected By Other Metrics?

To answer RQ3, we analyzed whether the mistranslations reported by different metrics overlap. We also explored whether the combination of metrics can improve the performance with respect to accuracy, F-Score and the number of issues detected. As previously mentioned, a metric's performance varies with threshold values. Therefore, we chose the threshold value that maximizes its performance based on the largest product of true positives and false negatives for each metric in order for fair comparison.

Fig. 7 compares the number of bugs uniquely and commonly detected by three semantic-based metrics (i.e., REG, DFA, HYB) with other metrics. As shown in the figure, DFA detects the most mistranslations (213) compared with other metrics, with 40 (i.e., 213 - 173) more than the second most. Besides, though the total number of mistranslations reported by REG and HYB are not the most, the number of unique mistranslations (i.e., mistranslations that can only be detected by one metric) are high, with 19 and 18, respectively. Such results also reveal that existing metrics mildly complement each other since they can detect different mistranslations, indicating that the combination of them may lead to improvement. Motivated by this, we studied if the performance of SemMT can be boosted by combining it with other existing techniques. Specifically, we adopted a simple strategy which assumes a translation as buggy if either of the two combined metrics report it as a mistranslation.

The experiment result is illustrated in Fig. 8. The heatmaps show the increase of the number of issues, ratio of accuracy and F-Score according achieved by different combinations. The value in each grid (e.g., v[i][j] in i-th row, j-th column) represents the improvement of the i-th similarity metric combing with the j-th metric. The greatest improvement is achieved when LEVEN is combined with DFA, detecting 102 more issues and achieving 16% improvement in F-Score. In particular, a combined use of any of our three metrics with an existing one can detect 38 to 102 more issues. Our metrics can also detect 9 to 79 more mistranslations when combined with another existing metric. Even for DFA which has already reported 213 issues, a combination with SBERT can help to detect 24 more bugs. The F-Score of all similarity metrics are mostly improved when a metric is combined with another one. Finally, we combined all 7 metrics and found that 246 bugs can be found, with 1% increase in F-score.

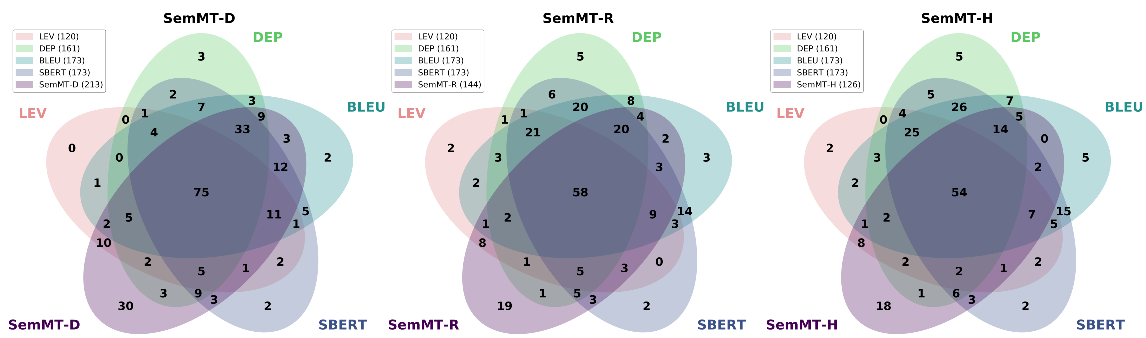



Fig. 7. **Mistranslations Reported By Different Similarity Metrics.**

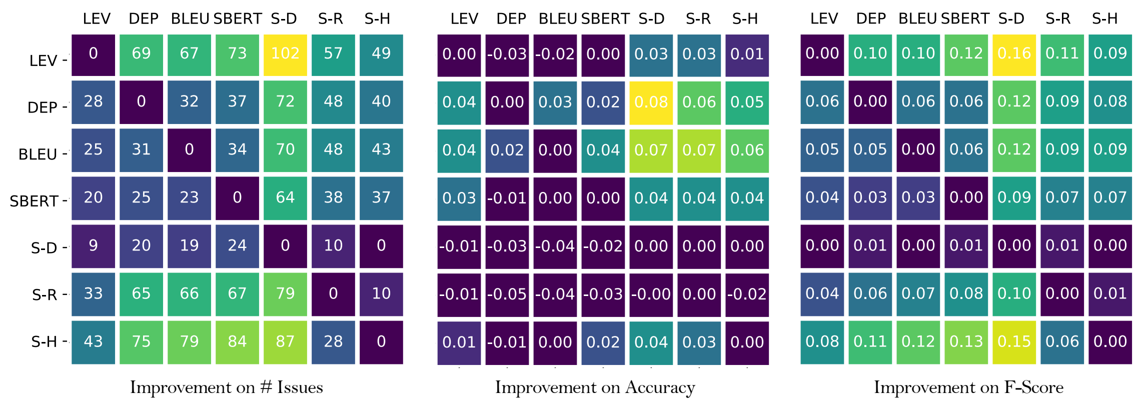



Fig. 8. **Improvement on Number/Accuracy/F-Score of Mistranslations Detection Reported By Different Similarity Metrics.** (S-D, S-R and S-H are abbreviations for SemMT-D, SemMT-R and SemMT-H, respectively.)

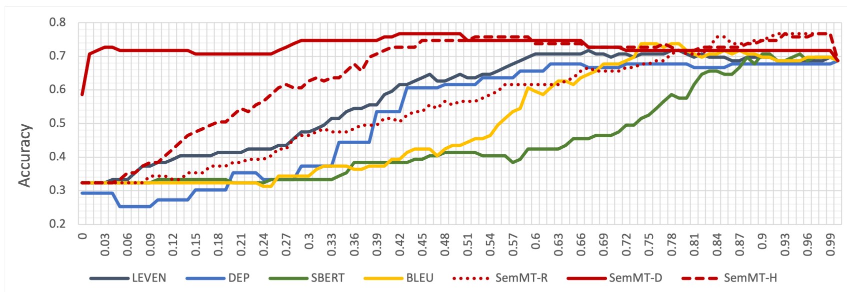
**Summary of Findings Related to RQ3:** The combination of different metrics can improve the effectiveness mutually to a large degree in terms of number of mistranlations reported, accuracy and F-Score. In particular, DFA boosts the performance of other similarity metrics the most, with 102 more mistranslations detected and 16% higher F-Scores achieved. Besides, DFA can be in turn boosted by SBERT with 24 more mistranslations.

### 4.5 RQ4: Applicability of SemMT

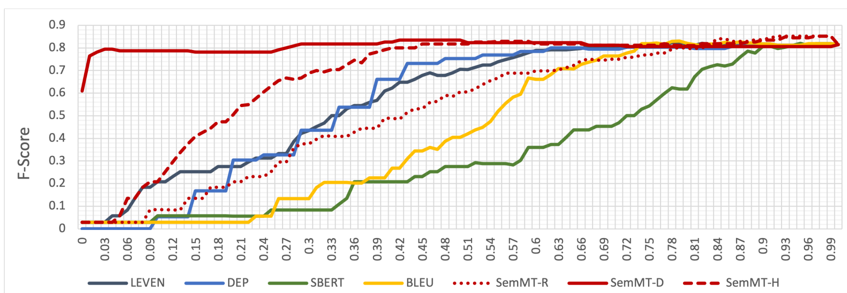
In the above RQs, we evaluate the effectiveness of SemMT by testing Google translator under the EN-ZH (i.e., English-Chinese) language setting. In the following, we present three experiments to further investigate whether similar effectiveness can be observed on other popular translators, language pairs and test sets, respectively.

*4.5.1* ***Impact of Translator Under Test.*** Our first experiment is to repeat the experiment in RQ1 by replacing the Google translator with Microsoft Bing Translator. Specifically, we randomly selected 100 sentences as test inputs from the NL-RX-Synth dataset, applied the round-trip translation on the Bing translator and collected the translation results.[12] We repeated the labeling procedure as described in § 4.1 and calculated similarities across thresholds 0.0 to 1.0 at the step of 0.01, making sure the experiment setup is the same as that of RQ1 apart from the translator under test.

The experiment result is shown in Fig. 9. We can see that SemMT-D and SemMT-H outperform other metrics on both accuracy and F-Score for most threshold values. Specifically, SemMT-R achieves the highest accuracy (76%) and F-Score (84%) when the threshold is above 0.9. Among the existing metrics, LEVEN and DEP achieve the highest accuracy and F-Score than others in general, , while SBERT is the least effective one for most threshold values. The result also shows that changing the translator under test cast little impact on the effectiveness of the three SemMT metrics. Specifically, different variants of SemMT still outperform existing baselines significantly while SemMT-D performs the best over most threshold values.




(a) Accuracy




(b) F-Score

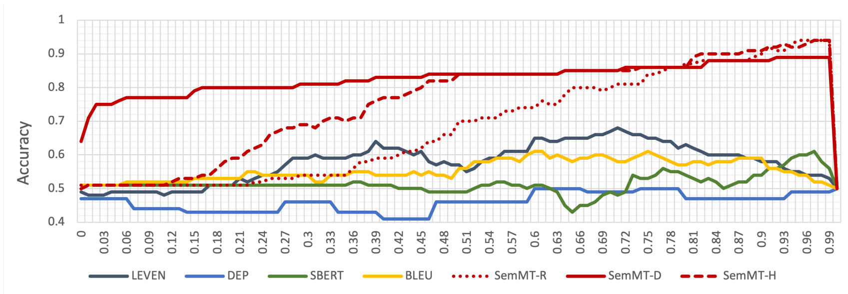
Fig. 9. Effectiveness of Mistranslation Detection On the Microsoft Bing Translator.

#### 4.5.2 *Impact of Language Pair.*

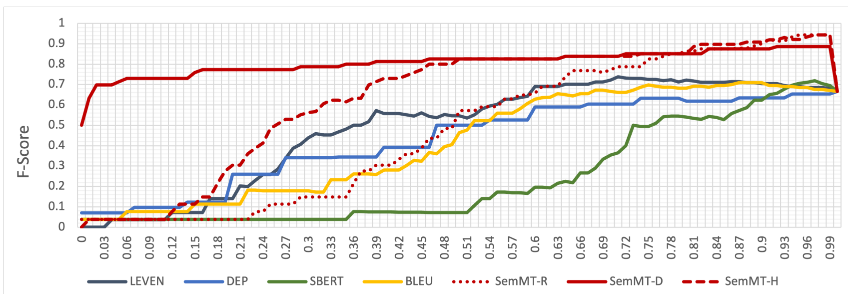
We also examined whether the use of language pair would affect the effectiveness of SemMT. Specifically, besides the translation between English and Chinese, we conducted another round-trip translation between English and Japanese using the Google translator on 100 randomly selected sentences from the NL-RX-Synth dataset.[13] The experiment results are illustrated in Fig. 10. The result shows that our three metrics outperform others for most of the thresholds. And among our three similarity metrics, SemMT-D outperforms SemMT-R and SemMT-H at most times. Among other existing metrics, LEVEN outperforms other existing metrics in terms of accuracy and F-Score, while DEP and SBERT reach the lowest accuracy and F-Score, respectively. The result also echos that in Fig. 5,

[12] The translation results were collected on March 11, 2021 on Microsoft Bing translator.
[13] The translation results were collected on March 11, 2021 on Google translator.

indicating that the change of language pair has little impact on the effectiveness of our three SemMT metrics, i.e., they outperform existing baselines significantly while SemMT-D achieves the best overall effectiveness. Furthermore, as compared with the result of RQ1 (as shown in Fig. 5), SemMT shows similar effectiveness over the thresholds when changing the language pair, which indicates the effectiveness of SemMT may hold under the change of language pairs.




(a) Accuracy



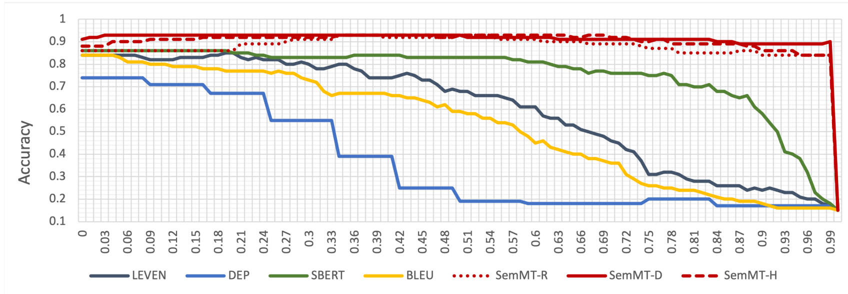

(b) F-Score

Fig. 10. Effectiveness of Mistranslation Detection on the English-Japanese Language Pair.

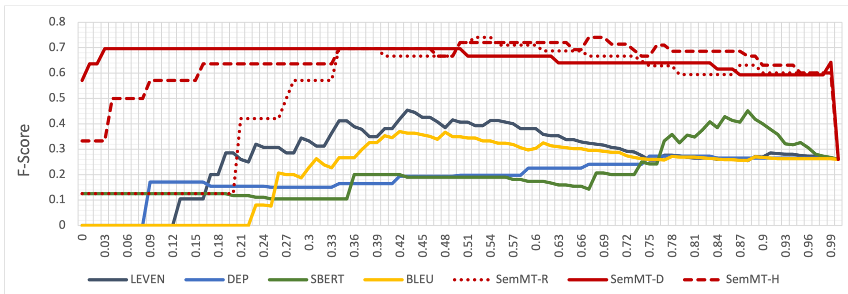
#### 4.5.3 *Impact of Test Dataset.*

Finally, we explored whether similar effectiveness can be achieved given test inputs extracted from other datasets. We thus randomly selected 100 sentences from the KB13 dataset to proceed the evaluation. The round-trip translation was conducted on the Google translator between English and Chinese.[14] The result is illustrated in Fig. 11. As shown in the red lines, we can see that three metrics of SemMT outperform other metrics as the threshold changes. Specifically, SemMT-D achieves the highest effectiveness on average, while SBERT and BLEU are less effective among these similarity metrics. In addition, compared with the results displayed in Fig. 5, our three metrics follow similar trends on both datasets, indicating the potential applicability of applying SemMT on various datasets. While other existing metrics (such as LEVEN, SBERT and BLEU) show apparently less effectiveness in terms of F-Score than that on NL-RX-Synth dataset.

[14]The translation results were collected on March 11, 2021 on Google translator.




(a) Accuracy




(b) F-Score

Fig. 11. Effectiveness of Mistranslation Detection Over KB13 Dataset.

**Summary of Findings Related to RQ4:** The applicability of SemMT has been evaluated by changing the translator (i.e., Bing translator), language pair (English-Japanese) and test set (KB13 [47, 58]). The results obtained are similar with those in RQ1, which indicate that SemMT is applicable to performing testing under different settings, and three similarity metrics of SemMT outperform the existing ones for most thresholds.

### 4.6 Summary and Recommendation

We made three observations on SemMT's performance from our experiments. First, the experiment results reveal that our proposed similarity metrics (i.e., SemMT-R, SemMT-D and SemMT-H) outperform the baselines on a wide range of threshold values in terms of accuracy and F-Score, etc. Second, compared with other state-of-the-art works, SemMT offers a better balance on precision and number of mistranslation detected, and achieves an improvement of 34.2% and 15.4% on accuracy and F-Score, respectively, with a similar number of issues detected. Third, we investigate the potential improvement that can be achieved by combining different metrics, and find that SemMT-D can boost the performance of other similarity metrics with 102 more issues detected and 16% higher F-Scores achieved.

The three SemMT similarities have their own merits. For best precision (> 0.8), SemMT-R with small threshold (< 0.1) is a good choice. For best recall (> 0.8), SemMT-H with high threshold (> 0.8) and a K value of 0.5 is recommended. For best F-Score, SemMT-D outperforms the other two metrics in a wide threshold range. One may switch between the three similarity metrics and adjust the threshold value according to the need of an application.

# 5 DISCUSSION

## 5.1 Characteristics of Mistranslations Detected by SemMT

To follow up on RQ3, since each similarity metric tends to capture certain aspect of mistranslations, we then analyzed the types of mistranslations detected by each metric for further investigation. We manually labeled the type of mistranslations according to the existing work [35], which has concluded five types of mistranslations (i.e., Under-Translation, Over-Translation, Word/phrase Mistranslation, Incorrect Modification and Unclear Logic).

The statistics are listed in Table 4. We can see that for the first three types, the number of mistranslations detected by each metric type are relatively similar, while for "Unclear logic" and "Modification", the number varies a lot. As such, we further subdivided these two mistranslation types to better characterize these two categories of mistranslations detected by SemMT. These new subcategories are motivated by the mistranslations detected in sentences that are mainly logic- or quantifier-related.

We subdivide "Unclear Logic" (i.e., all the tokens/phrases are correctly translated but the sentence logic is incorrect [35]) into three subcategories.

1 **Incorrect Order.** If all tokens and phrases are translated correctly, yet the order of tokens/phrases are organized in different orders after translation, it is an incorrect order mistranslation. For example, as shown in Table 5 (Example 1), in the original sentence, the string *"dog"* is arranged before the string *"truck"* or a *letter* , while after translation, the *letter* may be arranged before string *"truck"*.

2 **Incorrect Affiliation.** If all tokens and phrases are translated correctly, yet the affiliation relation is incorrect, it is an incorrect affiliation mistranslation. As shown in Table 5 (Example 2), the string contains letters and lowercase letters in the original sentence, while after translation, the affiliation is missed.

3 **Incorrect Semantics.** If tokens/phrases are correctly translated, yet the semantic logic of the original sentence is not preserved after translation, it is an incorrect semantics mistranslation. An example is presented in Table 5 (Example 3), the original semantic logic describes that the lines contains 5 or more numeric characters, while after translation, the semantic meaning is changed, describing the number of lines instead of the lines themselves.

For "Incorrect Modification" (i.e., if the modifier modify the wrong element), we subdivide it into two subcategories according to the type of modifiers.

1 **Incorrect Qualitative Modification**. If a qualitative modifier modifies the wrong element in a sentence, it is an incorrect qualitative modification mistranslation. An example is illustrated in Table 5 (Example 4). The attribute *"string"* modifies *"dog"* in the original sentence, while after translation, the modifier of *"dog"* becomes *"numeric character"*.

2 **Incorrect Quantitative Modification.** Similarly, if a qualitative modifier modifies the wrong elements , it is an incorrect quantitative modification mistranslation. An example is illustrated in Table 5 (Example 5). The quantitative modifier *"at least once"* modifies different elements after translation.

## 5.2 False Positives of SemMT

**False Positives Caused by Singularity and Plurality.** When we analyzed the mistranslations reported by SemMT in dataset used by RQ4.2, there are 148 mistranslations among 500 are plural-related, i.e., nouns are the same in both singular and plural forms in Chinese, while they are in different forms in English. As a result, the singularity/plurality is mistakenly missed or imposed during translation. Such plural-related mistranslations are commonly-seen across

Table 4. Number of Quantifier- and Logic-Related Mistranslations

| | LEVEN | DEP | BLEU | SBERT | SemMT-D | SemMT-R | SemMT-H |
|---|---|---|---|---|---|---|---|
| Under-Translation (35) | 27 | **30** | 32 | 28 | 28 | 24 | 20 |
| Over-Translation (6) | 5 | 5 | 5 | 1 | 5 | 4 | 4 |
| Mistranslation (1) | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Unclear logic (60) | 36 | 34 | 37 | 30 | **59** | 47 | 43 |
| Affiliation (32) | 13 | 13 | 13 | 10 | **31** | 21 | 21 |
| Order (19) | 14 | 12 | 15 | 11 | **19** | 17 | 13 |
| Semantic (9) | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| Modification (183) | 65 | 107 | 114 | 121 | **136** | 79 | 65 |
| Quality (8) | 4 | 6 | 6 | **7** | **7** | 6 | 5 |
| Quantity (175) | 61 | 101 | 108 | 114 | **129** | 73 | 60 |

Table 5. Examples of Five Subcategories of Mistranslations Reported by SemMT.

| | |
|---|---|
| **Example 1 (Incorrect Order)** | lines with the string "dog" **before** the string "truck" or **a letter**.<br>在字符串"truck"之前的字符串"dog"或字母<br>the string "dog" or **letter** **before** the string "truck". |
| **Example 2 (Incorrect Affiliation)** | strings **with a letter followed by** a lower-case letter, zero or more times.<br>字符串，其后跟一个小写字母的字母，零次或多次<br>string , **followed by a letter with** a lowercase letter, zero or more times. |
| **Example 3 (Incorrect Semantic)** | **lines with a number** , 5 or more times.<br>行数大于等于5次<br>**the number of rows** is greater than or equal to 5. |
| **Example 4 (Qualitative Modification)** | lines with the **string "dog"** before a vowel or a numeric character.<br>在元音或数字字符之前的字符串**"dog"**<br>string before vowel or **numeric character "dog"**. |
| **Example 5 (Quantitative Modification)** | lines starting with **a lower case letter** **at least once** or **a capital letter**.<br>以小写字母开头的行或至少一个大写字母的行<br>lines beginning with **lowercase letters** or lines with **at least one capital letter** |

languages. For example, languages such as Slovenian, Russian, and Welsh have several plural forms, while languages such as Chinese and Japanese do not have counterparts to the forms of singular and plural in languages like English. Though minor and easy-to-neglect, in aid of regex and DFA, our SemMT is able to capture such subtle differences to some degree, nevertheless false positives may be caused due to this reason.

**False Positives Caused by Inaccurate Transformation From Natural Language to Regular Expression.** The inaccurate transformation from natural language sentences to the corresponding regexes may also lead to false positives of SemMT. The main reason is caused by the existence of out-of-vocabulary words in the round-trip translated sentences, leading to the inaccurate transformation from natural language to regex. To alleviate such concerns, we enlarged the vocabulary of the training dataset in order to achieve accuracy as high as we can. Specifically, we collected the translation results from the Google translator, and obtained a list of parsed tokens. Then we augmented the training data by synonym substitution (i.e., replacing the tokens in the original training dataset by the tokens that are not in the original dataset but are derived in the returned dataset.).

### 5.3 Influence of Approximated Semantics Measurement on Mistranslation Detection

In §3.3.5, we explained how can the semantics be captured over- or under-approximately. However, if we apply the approximated semantics to detect mistranslations over source and translated sentences, the derived results are likely to be unreliable due to the approximated semantics. Take the example (i.e., S5) and its back-translated sentence (i.e., T5) in §3.3.5 as examples [15]:

- S5: The U.S. contains **a few** states which choose to have the judges on the state's courts serve for life terms.
- T5: The United States contains **several** states, which choose to let judges in state courts serve for life.

After sentence abstraction, we will obtain the following abstracted sentences:

- S5': [X] contains **a few** [Y].
- T5': [X] contains **several** [Y].

Since "a few" and "several" are vague quantifiers for which the semantics are hard to be precisely quantified, we approximate their semantics using regexes as follows:

- R_S5'$_O$: `[Y]{3,}`
- R_T5'$_O$: `[Y]{5,}`

After calculation, the SemMT-D semantic similarity is up to 0.957, meaning that there is little difference after translation. However, people tend to believe "a few" is less than "several" [36], while such semantic difference between these two vague quantifiers is hard to be captured after approximation. As a result, it may be harder to capture semantic difference due to the wider range of quantification.

Therefore, we discuss one possible solution, i.e., to quantify vague quantifiers more precisely by considering the common practice and the context of the sentence. Plenty of studies have been performed on quantifying vague quantifiers [10, 38, 43, 73, 85] from logical, linguistic and psychological aspects. For example, "a few" and "several" are believed to be less than the quantifier "a half". And being aware of the fact that the number of states in the U.S. is at most 50, then the quantification-related semantics of the above sentences (S5' and T5') can be more precisely quantified by the following regexes:

- RS5'$_P$: `[Y]{3,25}`
- RT5'$_P$: `[Y]{5,25}`

where the quantifiers "a few" and "several" are quantified to be `{3,25}` and `{5,25}`, respectively. By doing so, the SemMT-D similarity between them is 0.84, where the subtle semantic difference can be detected.

### 5.4 Buggy Trip Localization

Despite the advantage of round-trip translation, it has been criticized for not testing one translation system but two [54, 74]. Hence, in this section, we discuss a potential solution to localize buggy trip (the trip that produces wrong translation) using the idea of cross reference. The intuition is as follows: if the translation returned by one translator is

[15] The translation results were collected on December 24, 2020 on Google Translator, using Chinese as the intermediate language.

Table 6. **Statistics of Buggy Trip Localization.** The first three major columns denote the average similarity scores or distances over correctly/mistranslated sentences across different translators. The last major column shows the number and accuracy of correctly identified buggy trip using different similarity metrics. The values in bold represent the maximum number of correctly located buggy trip.

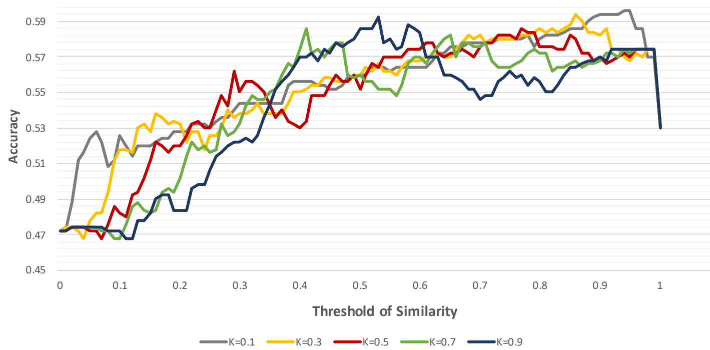
| | AveSim_Correct | | AveSim_BuggyFW | | AveSim_BuggyBW | | # FW | # BW | Accuracy |
|---|---|---|---|---|---|---|---|---|---|
| | Sim_FW | Sim_BW | Sim_FW | Sim_BW | Sim_FW | Sim_BW | | | |
| LEVEN | 0.50 | 0.64 | 0.39 | 0.66 | 0.42 | 0.54 | 57 | 19 | 0.65 |
| DEP | 3.15 | 6.34 | 3.07 | 5.70 | 3.28 | 8.87 | 36 | 8 | 0.38 |
| BLEU | 0.42 | 0.67 | 0.36 | 0.74 | 0.44 | 0.62 | 55 | **23** | 0.67 |
| SBERT | 0.94 | 0.90 | 0.91 | 0.92 | 0.95 | 0.89 | **74** | 15 | 0.76 |

different from the results of other translators, it is likely to be incorrect. The less similar with other translation results, the more likely the original sentence is error-prone.

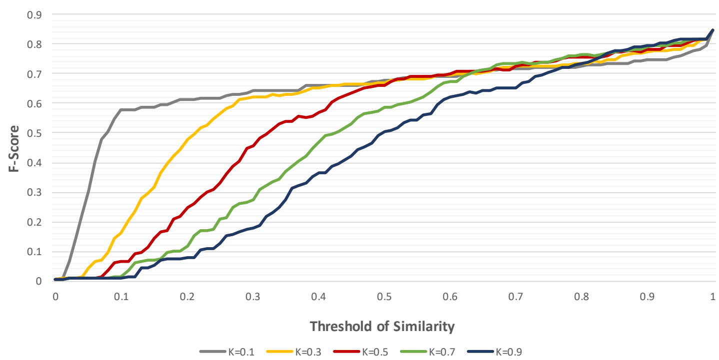
For better understanding, we analyzed 500 pairs of round-trip translations used in §4.2 and manually identified the buggy trip for all the 265 mistranslations. If both trips are mistranslated, we regarded it as a forward mistranslation because it is where the mistranslation was first introduced. Apart from the 148 plural-related mistranslations, there are 87 forward mistranslations and 30 backward mistranslations in total. For cross reference, we used the Microsoft Bing and Youdao translator. Then we conducted preliminary statistics, calculating the average similarity scores using four similarity metrics (i.e., LEVEN, DEP, BLEU and SBERT) across two translation trips (i.e., forward trip translating from English to Chinese, and backward trip from Chinese to English) for correctly and incorrectly translated sentences.

The result is shown in Table 6. The values in the first three major columns (i.e., `AveSim_Correct`, `AveSim_BuggyFW`) and `AveSim_BuggyBW` denote the average similarity scores of the forward translations (columns `Sim_FW`) in Chinese and the backward sentences (columns `Sim_BW`) except for DEP which calculates the distance. The higher the similarity scores, the more similar the sentences that are translated by different translators. We can see that similarity scores are not identical in different translation trips, and on average, the similarity scores of the correct translated sentences (column `AveSim_Correct`) are higher than that of mistranslated ones (column `AveSim_BuggyFW` and `AveSim_BuggyBW`). For example, for LEVEN, the average similarity scores in the forward and backward trips are 0.50 and 0.64, respectively. And the average similarity on correctly translated sentences is 0.50, while if the forward translation on Google is mistranslated, the similarity between sentences translated by Google was lower (with 0.39) than 0.50. Similar patterns can be observed for other similarity metrics. To sum up, we made three observations: (1) The similarity score for correct or incorrect translations vary from languages. (2) Within the same language, the similarity scores also vary from different similarity metrics. (3) The similarity scores of the correct translated sentences are higher than that of mistranslated ones on average.

On top of these observations, we tried the following strategy: given an original sentence which has been mistranslated in either trip, if the difference between the average similarity scores (i.e., as shown in Table 6) for the correct translations and the forward similarity scores is larger than or equal to backward trip, then this mistranslation is considered as a forward-trip mistranslation, otherwise a backward-trip mistranslation. The result is listed in the last three columns in Table 6. We can see that the capability of differentiating mistranslation trip differs for different similarity metrics. Specifically, SBERT achieved the highest accuracy (76%) by correctly identifying 74 forward mistranslations over 87 and 15 backward mistranslations over 30, while the syntactic-based DEP has the lowest accuracy (38%). In addition, the




(a) Accuracy




(b) F-Score

Fig. 12. Effectiveness of Mistranslation Detection Over Different Parameter K on Hybrid-based Similarity.

result indicates different similarity metrics tend to have diverse capabilities in identifying different translation trips. For example, SBERT founds the most forward buggy trip (74) while BLEU performs better in identifying the buggy backward trip.

### 5.5 Effect of the Parameter K on SemMT-H Similarity

On top of the remarkable results achieved by SemMT, we explore a further question: *whether the setting of K will affect the effectiveness of SemMT-H?* To answer this question, we set K from 0.1 to 0.9 with a step of 0.2 (i.e., K is set to be 0.1, 0.3, 0.5, 0.7 and 0.9), and examined the accuracy and F-Score on SemMT-H with different values of K against all the threshold values. We can see from Fig. 12 that the general trends of accuracy are similar under various fluctuations, while the trends of F-Score vary largely. Specifically, for accuracy, choosing a smaller K value in the hybrid metric would achieve better performance when the similarity threshold is less than 0.4 or larger than 0.8. For F-Score, a small K value in the hybrid metric outperforms that using a large K value when threshold is less than 0.8, and the reverse situation is observed when using a threshold beyond (0.8 to 1.0). Furthermore, considering accuracy and F-Score together, SemMT-H similarity offers a better performance over a wide threshold range when K assumes a value between 0.3 and 0.5. And if

SemMT-H similarity is to be used under a small threshold value where precision takes priority, a small K such as 0.1 is preferable.

## 6 RELATED WORK

### 6.1 Machine Translation Testing

Machine translation testing aims at finding sentences that trigger translation errors [32]. Pesu et al. [60] first applied metamorphic testing on machine translation systems. They proposed a Monte Carlo method to avoid round-trip translation by selecting eight target languages given the fixed source language, English. Under the factorial design and analysis, they evaluated the performance of translation over different combination of the source and target languages. After that, more metamorphic testings were developed. Sun et al. [92] designed straight-forward metamorphic relations focusing on short sentences in *subject-verb-object* structure (e.g., "Mike loves to eat KFC" and "Mouse loves to eat KFC"). They generated test inputs by replacing human names before "likes" or "hates", and brands after them. Wang et al. [84] detected under- and over-translation in the absence of reference translation. By checking the frequency of occurrence and learning mappings between bilingual words and phrase, their work is able to detect these two kinds of mistranslations efficiently and scalably. Later, He et al. [35] and Sun et al. [77] developed metamorphic testing techniques for general translation errors based on the assumption that similar sentences should have similar translation results. To be more specific, SIT [35] generated similar testing inputs by substituting one word in a given sentence, such that the generated inputs are semantically-similar and syntactically equivalent as the given sentence. Then they further reported the suspected issues if structures of translated sentences are different. Similarly, TransRepair [77] conducted mutation testing via context-similar word replacement. The intuition is that the translations of both original sentence and mutants should be consistent except for the changed token. While PatInv [32], on the other hand, considered pathological invariance: sentences of with different meanings should not have the same translation. Following this intuition, they generated syntactically similar but semantically different sentences by either replacing one word with a non-synonymous word using masked language models or removing one word based on its constituency structure. They further detected the potential mistranslations with closer textual similarity. We can see that the existing techniques mainly focused on textual or syntactical level, while the preservation of semantics during translation has not gained enough attention. Therefore, SemMT aims at filling this gap by evaluating whether the semantic meaning is preserved during translation, complementary to the existing works.

### 6.2 Robustness of Neural Machine Translator

Evaluating on adversarial examples has become a standard procedure to measure the robustness of deep learning models [30]. Adversarial examples are inputs designed to slightly manipulate the real-world examples such that a well-trained machine learning model performs poorly against these adversarial examples [27]. In general, these works mainly fall into two categories: black-box and white-box methodologies. In particular, for black-box adversarial samples generation, they assume the implementation of translation system is agnostic. Belinkov et al. [7] showed that character-level machine translation systems are overly sensitive to random character manipulations, such as keyboard typos. They used black-box heuristics to generate character-level adversarial examples, without using the model parameters or gradients to generate adversarial examples. Zhao et al. [88] searched for black-box adversarial examples in the space of encoded sentences and generate adversarial examples by perturbing the latent representation until the model is tricked. On the other hand, the white-box methodologies are model-aware. Ebrahimi et al. [27] investigated adversarial examples of both untargeted and targeted attack for character-level neural machine translation in a white-box manner. They transferred this problem into an optimization problem, then generated the adversarial examples utilizing gradients of translation models to inflict more damaging manipulations for a larger decrease in the BLEU score or other target metrics.

### 6.3 Quality Estimation of Machine Translation

Unlike machine translation testing, quality estimation considers beyond correctness - it aims at deriving similar estimation results made by humans. Traditionally, it estimates the required amount of post-editing efforts for converting the given translation result to the reference translation [72]. In the recent decade, the trend is to find effective quality estimation metrics which can directly provide scores to the translation result without human-written reference [28]. To alleviate the manual effort in providing reference translation, round-trip translation (RTT) has been proposed [40, 71]. The general idea of RTT is to use the original sentence as reference, compared the translated sentence with it, and calculated estimation metrics such as BLEU score to show the correlation with human judgement. In early 2010s, Aiken el al. [2] manually reassessed the correlation on input sentences and translated outputs and reassessed this correlation in a RTT manner. The result implied that if a suitable semantic-level metric is provided, RTT-based method can be reliably used for machine translation evaluation. They also pointed out that RTT quality might reflect the general quality of machine translation system used over the length of a longer text or multiple language pairs. Afterwards, with the emergence of BERT [25] and SBERT [67], semantic similarity on both word- and sentence-level can be better captured [11]. Moon et al. [54] then revisited RTT for quality estimation and achieved the highest correlations with human judgments compared with the state-of-the-art works, indicating that RTT-based method can be used to evaluate machine translation systems when semantic similarity is considered. By observing the correlated results, they illustrated the robustness of the choice of backward translation system on RTT-based quality estimation. It motivates us to adopt RTT and develop three semantic similarity metrics.

## 7 THREATS TO VALIDITY

The semantic metrics used in SemMT rely on the transformation of regexes from natural language. The precision of regex synthesis may affect SemMT's performance. If the derived regexes are imprecise, our proposed semantic similarity metrics might not accurately measure the real semantic relationship between source and translated sentence. To alleviate the influence of imprecise regex transformation, we adopted the state-of-the-art model and trained the model on augmented dataset to minimize the inaccurate prediction caused by out-of-vocabulary problem.

Our analysis on false positives reported by SemMT shows that the performance of SemMT can also be influenced by invalid synthesized regexes such as unpaired or mis-paired brackets (e.g., `[a-z{0,3}` is an invalid regex due to lacking of a square bracket) or incorrect combination of operators (e.g., `+{0,3}`). The invalid regex cannot be transformed into DFA, making the similarity computation of DFAs infeasible. To solve this problem, we performed post processing of transformed regexes to mitigate this problem, pairing/repairing the unpaired/mis-paired brackets.

Moreover, the precision cannot be ensured if the original sentence presents unclear/ambiguous logic [47, 58, 89]. Even though the round-trip testing that we used is to compare two sentences in the same language, the ambiguity of translation between the two languages in the forward trip and backward trip can influence our test results. The consequences of ambiguity include mistranslation of sentences and incorrect regex transformation (i.e., the regex is mistakenly transformed, leading to the change in semantic meaning after transformation). Besides, the reliance on published transformation tools from natural language to regex limits our evaluation to English datasets.

Finally, the English proficiency of authors may cast impacts on the evaluation. To alleviate such impacts, we consulted a linguist to ensure the semantics are preserved during the synonym substitution (during dataset augmentation and mutation operator construction described in § 4.1). In addition, for quantifying vague quantifiers when conducting semantics approximation (§3.3.5 and §5.3), we also discussed with the expert linguist for confirmation.

## 8 CONCLUSION

In this paper, we proposed SemMT, a semantic-based machine translation testing framework. It tests the semantic similarity during translation, taking the first step to semantic-aware testing approach for translation systems. Specifically, we focused on the semantics of quantifiers and logical relations, which take up a non-trivial ratio in the daily life, and the mistranslation of them may cause severe consequences. Via transforming such sentences into regular expressions, SemMT can capture the semantics of sentences during translation, and detect the suspicious mistranslation by semantic similarity measurements. The evaluation showed that SemMT can achieve higher effectiveness compared with state-of-the-art works, achieving an increase of 34.2% on accuracy. Furthermore, considering the unique mistranslation detection, SemMT can cover the most of the bugs found by existing techniques and can locate 67 additional bugs that are ignored by existing techniques. Furthermore, our exploration also indicated the potential improvement may occur when proper combinations of various similarity metrics are adopted.

## REFERENCES


[1] Khurshid Ahmad and Heather Fulford. 1992. Semantic relations and their use in elaborating terminology. *Computing Sciences Report CS-92-07. Guildford: University of Surrey* (1992).

[2] Milam Aiken and Mina Park. 2010. The Efficacy of Round-trip Translation for MT Evaluation. *Translation Journal* 14, 1 (2010).

[3] Jens Allwood, Gunnar-Gunnar Andersson, Lars-Gunnar Andersson, and Osten Dahl. 1977. *Logic in Linguistics*. Cambridge University Press.

[4] Arvind Arasu, Venkatesh Ganti, and Raghav Kaushik. 2006. Efficient Exact Set-Similarity Joins. In *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*. ACM, 918–929.

[5] Elke Bach, Eloise Jelinek, Angelika Kratzer, and Barbara BH Partee. 2013. *Quantification in natural languages*. Vol. 54. Springer Science & Business Media.

[6] Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence*. Springer, 241–301.

[7] Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=BJ8vJebC-

[8] Geert Jan Bex, Wouter Gelade, Frank Neven, and Stijn Vansummeren. 2010. Learning Deterministic Regular Expressions for the Inference of Schemas from XML Data. *ACM Trans. Web* 4, 4 (2010), 14:1–14:32.

[9] Geert Jan Bex, Frank Neven, Thomas Schwentick, and Stijn Vansummeren. 2010. Inference of concise regular expressions and DTDs. *ACM Transactions on Database Systems (TODS)* 35, 2 (2010), 1–47.

[10] NORMAN M. BRADBURN and CARRIE MILES. 1979. Vague Quantifiers. *Public Opinion Quarterly* 43, 1 (1979), 92–101.

[11] Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval-2017*. Association for Computational Linguistics, 1–14.

[12] Carl Chapman and Kathryn T. Stolee. 2016. Exploring regular expression usage and context in Python. In *Proceedings of the 25th International Symposium on Software Testing and Analysis, ISSTA*. ACM, 282–293.

[13] Danqi Chen and Christopher D. Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 740–750.

[14] Qiaochu Chen, Xinyu Wang, Xi Ye, Greg Durrett, and Isil Dillig. 2020. Multi-modal synthesis of regular expressions. In *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*. ACM, 487–502.

[15] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 1657–1668.

[16] Tsong Yueh Chen, S. C. Cheung, and Siu-Ming Yiu. 1998. Metamorphic Testing: A New Approach for Generating Next Test Cases. Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong.

[17] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, TH Tse, and Zhi Quan Zhou. 2018. Metamorphic testing: A review of challenges and opportunities. *Comput. Surveys* 51, 1 (2018), 1–27.

[18] Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020. Exploring Logically Dependent Multi-task Learning with Causal Inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2213–2225.

[19] Noam Chomsky and George A Miller. 1958. Finite state languages. *Information and control* 1, 2 (1958), 91–112.

[20] Deborah Coughlin. 2003. Correlating automated and human assessments of machine translation quality. In *Proceedings of MT summit IX*. Citeseer, 63–70.

[21] Cewei Cui, Zhe Dang, Thomas R. Fischer, and Oscar H. Ibarra. 2013. Similarity in languages and programs. *Theor. Comput. Sci.* 498 (2013), 58–75.

[22] Bernadeta Dadonaite and Hannah Ritchie. 2018. Diarrheal diseases. *Our World in Data* (2018). https://ourworldindata.org/diarrheal-diseases.

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

[24] James C. Davis, Daniel Moyer, Ayaan M. Kazerouni, and Dongyoon Lee. 2019. Testing Regex Generalizability And Its Implications: A Large-Scale Many-Language Measurement Study. In *34th IEEE/ACM International Conference on Automated Software Engineering, ASE*. IEEE, 427–439.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, 4171–4186.

[26] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using Structured Events to Predict Stock Price Movement: An Empirical Investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1415–1425.

[27] Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On Adversarial Examples for Character-Level Neural Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 653–663. https://www.aclweb.org/anthology/C18-1055

[28] Erick R. Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 Shared Tasks on Quality Estimation. In *Proceedings of the Fourth Conference on Machine Translation, (WMT)*. Association for Computational Linguistics, 1–10.

[29] Hector Garcia-Molina. 2008. *Database systems: the complete book*. Pearson Education India.

[30] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[31] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search Engine Guided Neural Machine Translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*. AAAI Press, 5133–5140.

[32] Shashij Gupta, Pinjia He, Clara Meister, and Zhendong Su. 2020. Machine Translation Testing via Pathological Invariance. In *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering ESEC/FSE*. 863–875.

[33] Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2020. Logical Inferences with Comparatives and Generalized Quantifiers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 263–270.

[34] Kim M. Hazelwood, Sarah Bird, David M. Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, James Law, Kevin Lee, Jason Lu, Pieter Noordhuis, Misha Smelyanskiy, Liang Xiong, and Xiaodong Wang. 2018. Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective. In *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE Computer Society, 620–629.

[35] Pinjia He, Clara Meister, and Zhendong Su. 2020. Structure-Invariant Testing for Machine Translation. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE '20)*. 961–973.

[36] Jaakko Hintikka and Gabriel Sandu. 1994. What is a Quantifier? *Synthese* (1994), 113–129.

[37] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. 2007. *Introduction to automata theory, languages, and computation, 3rd Edition*. Addison-Wesley.

[38] JJ Hox, N Borgers, and Dirk Sikkel. 2003. Response quality in survey research with children and adolescents: the effect of labeled response options and vague quantifiers. *International Journal of Public Opinion Research* 15, 1 (2003), 83–94.

[39] Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, Vol. 5. Citeseer, 79–86.

[40] Philipp Koehn and Christof Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings on the Workshop on Statistical Machine Translation, (StatMT)*. Association for Computational Linguistics, 102–121.

[41] Howard S Kurtzman and Maryellen C MacDonald. 1993. Resolution of quantifier scope ambiguities. *Cognition* 48, 3 (1993), 243–279.

[42] Nate Kushman and Regina Barzilay. 2013. Using Semantic Unification to Generate Regular Expressions from Natural Language. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. The Association for Computational Linguistics, 826–836.

[43] Shalom Lappin. 2000. An intensional parametric semantics for vague quantifiers. *Linguistics and philosophy* (2000), 599–620.

[44] Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*. Association for Computational Linguistics, 4791–4796.

[45] Chi-kiu Lo. 2019. YiSi - a Unified Semantic MT Quality Evaluation and Estimation Metric for Languages with Different Levels of Available Resources. In *Proceedings of the Fourth Conference on Machine Translation, (WMT)*. Association for Computational Linguistics, 507–513.

[46] Sebastian Löbner. 1987. Quantification as a major module of natural language semantics. *Studies in discourse representation theory and the theory of generalized quantifiers* 8 (1987), 53.

[47] Nicholas Locascio, Karthik Narasimhan, Eduardo DeLeon, Nate Kushman, and Regina Barzilay. 2016. Neural Generation of Regular Expressions from Natural Language with Minimal Domain Knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. The Association for Computational Linguistics, 1918–1923.

[48] Fordham Research Ltd. [n.d.]. Housing Needs Study. https://www.ipswich.gov.uk/sites/default/files/Housing_Needs_Study.pdf.

[49] Fiona Macdonald. 2015. The Greatest Mistranslations Ever. (2015). https://www.bbc.com/culture/article/20150202-the-greatest-mistranslations-ever

[50] Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting Evaluation in Context: Contextual Embeddings Improve Machine Translation Evaluation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2799–2808.

[51] Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (ACL)*. Association for Computational Linguistics, 4984–4997.

[52] Kris McDaniel. 2004. Modal realism with overlap. *Australasian Journal of Philosophy* 82, 1 (2004), 137–152.

[53] David Hugh Mellor. 2006. Wholes and parts: The limits of composition. *South African Journal of Philosophy* 25, 2 (2006), 138–145.

[54] Jihyung Moon, Hyunchang Cho, and Eunjeong L. Park. 2020. Revisiting Round-Trip Translation for Quality Estimation, In the 22nd Annual Annual Conference of the European Association for Machine Translation (EAMT). *CoRR*, 91–104.

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

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

[57] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 311–318.

[58] Jun-U. Park, Sang-Ki Ko, Marco Cognetta, and Yo-Sub Han. 2019. SoftRegex: Generating Regex from Natural Language Descriptions using Softened Regex Equivalence. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing EMNLP-IJCNLP*. Association for Computational Linguistics, 6424–6430.

[59] Barbara BH Partee, AG ter Meulen, and R Wall. 1990. *Mathematical Methods in Linguistics*. Vol. 30. Springer Science & Business Media.

[60] Daniel Pesu, Zhi Quan Zhou, Jingfeng Zhen, and Dave Towey. 2018. A Monte Carlo Method for Metamorphic Testing of Machine Translation Services. In *3rd IEEE/ACM International Workshop on Metamorphic Testing MET*. ACM, 38–45.

[61] Stanley Peters, Dag Westerstahl, and Dag Westerstahl. 2006. *Quantifiers in language and logic*. Oxford University Press.

[62] Sandro Pezzelle, Shane Steinert-Threlkeld, Raffaella Bernardi, and Jakub Szymanik. 2018. Some of Them Can be Guessed! Exploring the Effect of Linguistic Context in Predicting Quantifiers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*. Association for Computational Linguistics, 114–119.

[63] Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, (WMT@EMNLP)*. The Association for Computer Linguistics, 392–395.

[64] Christopher Potts. 2007. Logic for linguists. Course for LSA institute.

[65] United Nations World Food Programm. 2020. *Policy Brief: The Impact of COVID-19 on children.* Technical Report. United Nations World Food Programm. https://www.wfp.org/publications/policy-brief-impact-covid-19-children.

[66] Aarne Ranta. 1998. A multilingual natural-language interface to regular expressions. In *Finite State Methods in Natural Language Processing (FSMNLP '09)*. 79–90.

[67] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 3980–3990.

[68] Eric Sven Ristad and Peter N. Yianilos. 1997. Learning String Edit Distance. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 287–295.

[69] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.

[70] Richard Sharvy. 1980. A more general theory of definite descriptions. *The philosophical review* 89, 4 (1980), 607–624.

[71] Tomohiro Shigenobu. 2007. Evaluation and Usability of Back Translation for Intercultural Communication. In *Usability and Internationalization. Global and Local User Interfaces, Second International Conference on Usability and Internationalization, (UI-HCII) (Lecture Notes in Computer Science, Vol. 4560)*. Springer, 259–265.

[72] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, Vol. 200. Cambridge, MA.

[73] Stephanie Solt. 2011. Vagueness in quantity: Two case studies from a linguistic perspective. *Understanding vagueness. logical, philosophical and linguistic perspectives* (2011), 157–174.

[74] Harold L. Somers. 2005. Round-trip Translation: What Is It Good For?. In *Proceedings of the Australasian Language Technology Workshop, ALTA 2005, Sydney, Australia, December 10-11, 2005*. Australasian Language Technology Association, 127–133.

[75] Shashank Srivastava, Igor Labutov, and Tom M. Mitchell. 2018. Zero-shot Learning of Classifiers from Natural Language Quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Association for Computational Linguistics, 306–316.

[76] Wolfgang G. Stock. 2010. Concepts and semantic relations in information science. *J. Assoc. Inf. Sci. Technol.* 61, 10 (2010), 1951–1969.

[77] Zeyu Sun, Jie M. Zhang, Mark Harman, Mike Papadakis, and Lu Zhang. 2020. Automatic testing and improvement of machine translation. In *ICSE '20: 42nd International Conference on Software Engineering, Seoul, South Korea, 27 June - 19 July, 2020*, Gregg Rothermel and Doo-Hwan Bae (Eds.). ACM, 974–985.

[78] Jakub Szymanik. 2016. *Quantifiers and Cognition - Logical and Computational Perspectives*. Studies in linguistics and philosophy, Vol. 96. Springer.

[79] Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), 2214–2218.

[80] Margus Veanes, Peli de Halleux, and Nikolai Tillmann. 2010. Rex: Symbolic Regular Expression Explorer. In *Third International Conference on Software Testing, Verification and Validation, ICST*. IEEE Computer Society, 498–507.

[81] Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, (ACL)*. Association for Computational Linguistics, 1198–1212.

[82] Peipei Wang and Kathryn T. Stolee. 2018. How well are regular expressions tested in the wild?. In *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE*. ACM, 668–678.

[83] Wenyu Wang, Wujie Zheng, Dian Liu, Changrong Zhang, Qinsong Zeng, Yuetang Deng, Wei Yang, Pinjia He, and Tao Xie. 2019. Detecting Failures of Neural Machine Translation in the Absence of Reference Translations. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN (Industry Track) 2019, Portland, OR, USA, June 24-27, 2019*. IEEE, 1–4.

[84] Wenyu Wang, Wujie Zheng, Dian Liu, Changrong Zhang, Qinsong Zeng, Yuetang Deng, Wei Yang, Pinjia He, and Tao Xie. 2019. Detecting Failures of Neural Machine Translation in the Absence of Reference Translations. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN (Industry Track)*. IEEE, 1–4.

[85] Daniel B Wright, George D Gaskell, and Colm A O'Muircheartaigh. 1994. How much is 'quite a bit'? Mapping between numerical values and vague quantifiers. *Applied Cognitive Psychology* 8, 5 (1994), 479–496.

[86] Naomi Yamashita and Toru Ishida. 2006. Effects of machine translation on collaborative work. In *Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work (CSCW)*. ACM, 515–524.

[87] Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding Neural Machine Translation with Retrieved Translation Pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, 1325–1335.

[88] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating Natural Adversarial Examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=H1BLjgZCb

[89] Zexuan Zhong, Jiaqi Guo, Wei Yang, Jian Peng, Tao Xie, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2018. SemRegex: A Semantics-Based Approach for Generating Regular Expressions from Natural Language Specifications. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing EMNLP*. Association for Computational Linguistics, 1608–1618.

[90] Zexuan Zhong, Jiaqi Guo, Wei Yang, Tao Xie, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2018. Generating Regular Expressions from Natural Language Specifications: Are We There Yet?. In *The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI Workshops, Vol. WS-18)*. AAAI Press, 791–794.

[91] Zhiquan Zhou, Shaowen Xiang, and Tsong Yueh Chen. 2016. Metamorphic Testing for Software Quality Assessment: A Study of Search Engines. *IEEE Trans. Software Eng.* 42, 3 (2016), 264–284.

[92] Zhi Quan Zhou and Liqun Sun. 2018. Metamorphic Testing for Machine Translations: MT4MT. In *Proceedings of the 25th Australasian Software Engineering Conference (ASWEC)*. IEEE Computer Society, 96–100.