As Easy as 1, 2, 3: Behavioural Testing of NMT Systems for Numerical Translation

Jun Wang,[1] **Chang Xu,**[1] **Francisco Guzmán,**[2] **Ahmed El-Kishky,**[3*]
Benjamin I. P. Rubinstein,[1] **Trevor Cohn**[1]
[1]University of Melbourne, Australia
[2]Facebook AI, [3]Twitter Cortex
jun2@student.unimelb.edu.au
{xu.c3,benjamin.rubinstein,trevor.cohn}@unimelb.edu.au
fguzman@fb.com, aelkishky@twitter.com

Abstract

Mistranslated numbers have the potential to cause serious effects, such as financial loss or medical misinformation. In this work we develop comprehensive assessments of the robustness of neural machine translation systems to numerical text via behavioural testing. We explore a variety of numerical translation capabilities a system is expected to exhibit and design effective test examples to expose system underperformance. We find that numerical mistranslation is a general issue: major commercial systems and state-of-the-art research models fail on many of our test examples, for high- and low-resource languages. Our tests reveal novel errors that have not previously been reported in NMT systems, to the best of our knowledge. Lastly, we discuss strategies to mitigate numerical mistranslation.

1 Introduction

Just as neural machine translation (NMT) systems have achieved tremendous benchmark results, they have been proven brittle when faced with irregular inputs such as noisy text (Belinkov and Bisk, 2018; Michel and Neubig, 2018) or adversarial inputs (Cheng et al., 2020). Among such errors, mistranslation of *numerical text* constitutes a crucial but under-explored category that may have profound implications. For example, in the medical domain, mistranslating the number of confirmed cases of a contagious disease like COVID-19 may exacerbate public health misinformation. Numerical errors made in financial document translation, *e.g.*, an extra or omitted digit or decimal point, could lead to significant monetary loss. Surprisingly, we find that numerical mistranslation is a general issue faced by state-of-the-art NMT systems, including commercial and research systems, with evidence present across contexts: for both high and low resource languages, and for both close and distant languages.

Type	Input	Output
Separators	The distance is 557,601.101 meters.	**[De]**: Die Entfernung beträgt 557.601.101 Meter.
Numerals	The total weight is two hundred and two kg.	**[Zh]**: 总重量为220公斤。
Digits	The R0 of the disease is 3.28.	**[Ne]**: रोगको R0 २८.२२८ हो।
Units	There were 100.01 million cases worldwide.	**[Zh]**: 全世界有1.001 亿病例。(100.1 million)

Table 1: Numerical errors discovered by our method when behavioural testing two popular commercial translation systems using their public APIs.

De facto standard metrics such as BLEU (Papineni et al., 2002) may fail to flag a numerical translation error, which only contributes a very minor penalty, as it is typically a single-token mistranslation. To facilitate the discovery of numerical errors made by NMT systems, we propose a black-box test method[1] for assessing and debugging the numerical translation of NMT systems in a systematic manner. Our method extends the `CheckList` behavioural testing framework (Ribeiro et al., 2020) by designing automatic test cases to assess a suite of fundamental capabilities a system should exhibit in translating numbers.

Our tests on state-of-the-art NMT systems expose novel error types that have evaded close examination (Table 1). These error types greatly extend the *number* category (`NUM`) of the *catastrophic errors* (Specia et al., 2020) of NMT systems with richer error types. Finally, the abuse of these errors constitute vectors of attack: error-prone numerical tokens injected into monolingual data may

*This work was conducted while author was working at Facebook AI

[1]Our code is available at https://github.com/JunW15/NumberTest

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4711–4717
August 1–6, 2021.

Capability	Examples
Integers	There were 914 confirmed cases of COVID-19 reported yesterday.
Decimals	The reproduction number of COVID-19 is between 3.28 and 5.70.
Numerals	The total amount of transfer is fifty-two dollars and seven cents.
Separators	123,456.12 (En) → 123.456,12 (De)

Table 2: Tested capabilities of NMT systems in translating common types of numerical text.

corrupt *back-translation-based* training, as the resulting back-translated sentences are very likely to contain the desired errors.

2 Method

We follow Ribeiro et al. (2020)'s `CheckList` in designing our evaluation suite for NMT systems: we present several basic *capabilities* an NMT system should be expected to exhibit in translating common everyday numerical text; we then generate *test examples* specific to each capability to benchmark performance and find bugs in NMT systems.

2.1 Capabilities of Translating Numbers

We explore four capabilities (see Table 2), demonstrating expected translation ability of a system on common types of numerical text. Concretely, the *Integers* and *Decimals* represent basic capabilities; they can be manifested by testing on sequences of digits with variable lengths (*e.g.*, 100 vs. 10000) or decimals with the decimal mark placed at varying locations (1.001 vs. 10.01). We find that the tested NMT systems are more likely to malfunction when translating larger integers and decimals with longer fractional parts. The *Numerals* capability pertains to whether a system is able to translate numbers that are presented as words. The *Separators* capability checks if a model can deal with numbers containing decimal or thousands separators.[2] Systems that fail to manifest one or more of these capabilities may produce wrong numbers that can be inconspicuous to users and become a ready, exploitable source of misinformation.

[2]While *Decimals* and *Separators* may have overlapping instances (*e.g.*, the decimal mark), their specific formats in our testing are different (Table 2), which leads us to find non-overlapping error types: most *Decimals* errors involve translating numbers into wrong digits, whereas *Separators* errors pertain to mistranslation in localisation usage (*e.g.*, German and English use different decimal and thousand separators).

2.2 Test Examples

To efficiently test the identified capabilities across multiple systems on distinct language pairs, we generate desired test examples using templates. For example, to test the *Numerals* capability, we use a template sentence such as "CNBC reported there were at least `[NUM]` cases worldwide.", where "`[NUM]`" is a number with the format "`ddd.ddn`", consisting of multiple digits and a numeral (*e.g.*, 100.01 million).

We experiment with formats of various lengths and decimal-point positions. We fill a format with random digits and numerals, and explore 25 different formats across all capabilities. This allows us to generate a diversity of numbers at scale, akin to fuzzing a program with random inputs to uncover bugs. We also note that all the numbers created for a format can be seen as a set of "adversarial" examples, as they are small perturbations of each other. Details about the test examples for each capability and the testing process can be found in Supplementary material.

3 Evaluation

Before presenting experimental results and discussion of our test framework, we first detail our evaluation setup.

Language pairs. We test both high-resource (HR) and low-resource (LR) scenarios. For HR, we consider two language pairs: English-German and English-Chinese, and for LR, we focus on English-Tamil and English-Nepali. We test both translation directions for each pair.

SOTA systems. We conduct behavioural testing against two popular commercial translation systems (denoted by **A** and **B**). As research systems, we use pre-trained models that were shown to perform well in WMT competitions (denoted by **R**), specifically, fairseq's transformer for English-German (Ng et al., 2019), English-Tamil (Chen et al., 2020), and English-Chinese/Nepali (Fomicheva et al., 2020).

The evaluation metric. For each capability we curate a list of test examples (sentences containing numbers), which are taken from various sources, including existing corpora or manually crafted (details in Supplementary material). To these sentences we remove the number component, and replace it with a number based on the specific capa-

bility being tested. This test collection is then input to a translation system, and we report the *Pass Rate* (PR), the fraction of inputs where the system translates the numerical component perfectly.[3]

3.1 Testing Performance

Table 3 shows the results of testing the three SOTA systems across the HR/LR language pairs.

Among the four capabilities, *Numerals* turns out to be the most challenging across the systems tested, with the average $\overline{\text{PR}}$ of 70.8%. This is probably because, compared to other forms, numbers are less frequently written as words, resulting in insufficient examples available for training. At the other extreme, *Integers*, which tests on pure digits, is the easiest capability, as expected. Despite this, it is not a 'solved problem', given all systems report imperfect PR < 100 on at least one language.

Across the systems, the research system **R** ($\overline{\text{PR}}$: 77.8%) underperforms the two commercial ones ($\overline{\text{PR}}_A$: 80.6%, $\overline{\text{PR}}_B$: 90.4%). This is largely caused by the fact that the research system fails markedly on the En→Ne direction.

Per language, the results are similar in both translation directions, implying that numerical translation is a symmetric problem. Note that the results on LR are not always worse than that on HR (PRs on En-Ta are surprisingly the highest of all). This suggests that the size of training data is not the sole factor for high-quality numerical translation.

3.2 Error Analysis

We present analysis of novel types of mistranslations discovered from testing.

Decimal/thousands separators. We find that the decimal/thousands separators are prone to be mistranslated in localisation scenarios, when conventions differ between the languages (*e.g.*, "," and "." are the thousands and decimal separators in English while they are swapped in German). A common type of error is that a separator remains the same after translation (Table 4, row 1). This is probably due to the lack of sufficient training data to learn the translation of the separators in the target language.

Cardinal numerals. Cardinal numerals are commonly used in commercial and financial contexts. For example, the *financial characters* (*e.g.*, "壹" meaning one) are typical in Chinese financial documents. However, we find that the tested translation systems perform fairly poorly in translating cardinal numerals (Table 4, row 2). Common errors include mistranslation or under-translation of the unit words (*e.g.*, hundred) or the number words (*e.g.*, "陆拾"). Most often, the errors appear to be caused by the unique unit words used in different languages (*e.g.*, "万" in Chinese equals to 10 thousand), where a system needs to "compute" the correct amount for translation.

Digits. The pure digit translation (10→10) is expected to be easy, since a system may opt to copy the entire number as the translation. However, we find that the digit translation between English and low-resource languages can be far from satisfactory. An example is the translation between English and Nepali (Table 4, row 3). One reason for this result is that Nepali has its own numerals for digits (*e.g.*, १ denotes 1). As a result, a system would try to convert a digit into a Nepali digit (instead of keeping it unchanged) when translating numbers, which is difficult given limited training resources (Guzmán et al., 2019). Another common issue in digit translation is handling repeats of the same digit. A system is prone to omit or add one or more digits in the translation.

Units. This error often occurs when translating numbers accompanied by units of measurements (*e.g.*, 10 meters), especially when the target unit is unique to the language, *e.g.*, "角" in Chinese means "10 cents". In such cases (Table 4, last row), the system may need to learn the implicit conversion rules and then use them to "calculate" the correct numbers with the target unit of measurement. For example, when translating "10.01 million" into "1001 万" in Chinese, the system has to convert "10.01" into "1001" and then use the correct unit "万". An error may occur if the system fails either or both stages of this process (*i.e.*, mistranslating the numbers and/or units).

4 Potential Mitigation Strategies

Finally, we discuss several strategies that may mitigate the above errors discovered by our method[4].

[3] We count a Pass if the output matches the ground truth number, allowing for the use of digits (Arabic or local scripts) or numerals. For this purpose we use `num2words` (https://pypi.org/project/num2words/), `cn2an` (https://github.com/Ailln/Cn2An.jl), and locally developed scripts (for Ne and Ta).

[4] We leave validation of these strategies to future work.

Lang	Integers			Decimals			Numerals			Separators			Avg
	A	B	R	A	B	R	A	B	R	A	B	R	
En→Zh	100.0	100.0	94.0	100.0	100.0	92.2	77.5	72.5	67.5	100.0	100.0	91.4	91.2
Zh→En	94.0	78.0	90.0	100.0	100.0	93.8	82.0	78.0	56.7	100.0	100.0	83.3	88.0
En→De	100.0	100.0	100.0	93.8	78.1	68.8	87.5	67.5	95.0	83.3	80.0	80.0	86.2
De→En	100.0	100.0	100.0	98.4	79.7	95.3	87.0	84.0	65.7	97.1	68.5	65.7	86.8
En→Ta	100.0	98.0	100.0	100.0	100.0	100.0	100.0	97.5	100.0	100.0	100.0	100.0	99.6
Ta→En	98.0	100.0	100.0	98.4	100.0	98.4	96.0	100.0	90.0	100.0	88.6	100.0	97.4
En→Ne	18.0	-	70.0	17.2	-	72.0	7.5	-	65.0	16.7	-	60.0	**40.8**
Ne→En	98.0	-	88.0	96.9	-	1.6	46.7	-	5.3	80.0	-	0.0	**52.1**
Avg	88.5	96.0	92.8	88.1	92.9	77.8	**61.1**	83.2	**68.2**	84.6	89.5	**72.6**	-

Table 3: Test results (Pass Rate %) on the capabilities for numerical translation, with low averaged scores in bold. Nepali is not supported by System **B**.

Error Type	Sys/Lang	Test input sentence	System output	Ground-truth
Decimal/ Thousands separators	**A**, **B** En→De	The distance between Sydney to Washington is 9,756.001 miles.	Die Entfernung zwischen Sydney und Washington beträgt 9.756.001 Meilen.	9.756,001
	A, **B** En→De	The reproduction number of this disease is between 9.718 and 9.911.	Die Reproduktionszahl dieser Krankheit liegt zwischen 9.718 und 9.911.	9,718 and 9,911
	A, **B**, **R** De→En	Die Flugzeit von Punta Arenas beträgt etwa 85,619 Stunden.	The flight time from Punta Arenas is about 85,619 hours.	85.619
Cardinal numerals	**A, R** Zh→En	这支手表的总价是叁佰陆拾壹元。	The total price of this watch is [**A** three hundred and one yuan.] [**R** one dollar.]	361
	A, **B** En→Zh	The total amount of remittance is ninety thousand six hundred thirty-eight dollars and forty-seven cents.	汇款总额为 [**A** 九万六千三百八十八美元和][**B** 九千六百三十八元及]四十七美分。(**En**: **A** [96388 dollars, 47 cents] **B** [9638 yuan, 47 cents])	90638
Digits	**A** En→Ne	There have been 670 confirmed cases of COVID19 to date.	अहिले सम्म COVID19 को 7070० पुष्टि भएका घटनाहरू छन् । (**En**: 70700)	670
	B, R En→Zh	An average of 1000009 people is infected every day due to this disease.	平均每天有 [**B** 100009] [**R** 10009] 人因这种疾病而感染。	1000009
	R Ne→En	UNESCO ले अनुमान गरेको छ कि ५१८८८९ शिक्षार्थीहरू सम्भावित जोखिममा छन् ।षा)	UNESCO estimates that 51889 teachers are at potential risk	518889
Units	**A** En→Zh	CNBC reported there were at least 100.01 million cases worldwide.	CNBC 报道，全球至少有 [**A** 1 亿 1001 万 **En**: 110.01 million] 例。[**B** 1.001 亿 **En**: 100.1 million)]	1 亿1万
	B Zh→En	这两根电线的长度分别是十米和五分米。(**En**: decimetres)	The length of the two wires is ten meters and five meters respectively.	decimetres
	R En→Zh	Case report forms were submitted to CDC for 7.415 million cases.	现已就74.15万宗案件向中心提交案件报告表。	741.5万

Table 4: Examples of four major types of errors discovered by our tests on three SOTA NMT systems.

Separate treatment of numbers. Although NMT models have been shown capable of performing basic arithmetic or bracket matching (Suzgun et al., 2019), this paper demonstrates that handling the various forms of numerical text in reality is still challenging. It may be worth separating numerical translation out into an individual process, as in Statistical MT (Koehn, 2009), that identifies numbers in the input, applies specific translation rules to them, and incorporates the translation into the output (Tu et al., 2012).

Data augmentation. Training with more quality data leads to better translation quality (Barrault et al., 2020). In our testing, we observe a large proportion of errors (*e.g.*, financial characters, units) stemming from mistranslation of specific numerals that are unique or used less frequently (*e.g.*,

"角", decimetres) in a language. Such errors could potentially be reduced if more *numeral-specific* instances were added to training.

Tailoring BPE segmentation. The Byte Pair Encoding (BPE) has been used by most leading NMT systems. However, long sequences of digits or numbers with separators (*e.g.*, ",", ".") are often split into varying sized fragments by BPE. This would render learning more difficult, as the system has to account for the dependency between the partitions. To circumvent this, one may wish to segment numbers differently, *e.g.*, to encode all numbers as character sequences, or as meaningful groupings of components (*e.g.*, segment into groups of 3 digits when processing English.)

Sanity checks. It is helpful to post-check whether all numbers in a translation are correct by comparing them to the inputs. This could be automated in the same way as we measure the Pass Rate (§3), and once again drawing parallels to software testing, could be fully automated via continuous integration of NMT systems.

5 Conclusion

In this paper, we propose an evaluation method to systematically assess four fundamental capabilities of NMT systems in translation numbers by virtue of a variety of test cases. Our tests reveal novel types of errors that are general across multiple SOTA translation systems for both high and low resource languages. We hope that our study will help improve numerical translation quality and reduce misinformation caused by numerical mistranslation.

Acknowledgements

We thank all anonymous reviewers for their constructive comments. The authors acknowledge funding support by Facebook.

Impact Statement

This work aims to improve the performance of NMT systems. The impact of poor numerical translation may go beyond poor user experience, potentially leading to financial loss, medical misinformation, and even a vector for poisoning NMT systems. This paper's behavioural testing could be used by an attacker to uncover flaws in a commercial NMT system. However, as in attack research in the security community, responsible highlighting of such flaws serves the purpose of improving systems: knowledge of systemic flaws in numerical translations helps vendors improve their systems to mitigate these effects in the first place, while concerted attackers are likely to discover vulnerabilities independently.

References

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *ICLR*.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook AI's WMT20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav

Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. 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)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Lucia Specia, Zhenhao Li, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel, and Xian Li. 2020. Findings of the WMT 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation*, pages 76–91, Online. Association for Computational Linguistics.

Mirac Suzgun, Yonatan Belinkov, Stuart Shieber, and Sebastian Gehrmann. 2019. LSTM networks can perform dynamic counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 44–54, Florence. Association for Computational Linguistics.

Mei Tu, Yu Zhou, and Chengqing Zong. 2012. A universal approach to translating numerical and time expressions. In *International Workshop on Spoken Language Translation (IWSLT) 2012*.

A Appendix

A.1 Testing Process

Our behaviour testing proceeds in four steps.
1) Test template selection: we select English sentences from quality real corpora TICO-19 (Anastasopoulos et al., 2020) and WikiMatrix (Schwenk et al., 2019). TICO-19 contains documents about COVID-19 (*e.g.*, scientific articles, conversations, Wikipedia entries) between English and 36 languages. WikiMatrix consists of parallel sentences extracted from Wikipedia articles in 85 languages. We randomly select five sentences with each containing numbers for the evaluation of each capability. We make each sentence a template by replacing the contained number with the placeholder "`[NUM]`".
2) Template filling-in: we fill the templates with randomly generated numbers in the digital format (*e.g.*, 1230000). Then, we convert the digital number into desired formats for testing (*e.g.*, 1,230,000 for a separator or 1.234 million for a numeral).
3) Translation: To test a system, we use it to translate all the test examples across all capabilities, and collate the translation results.
4) Evaluation: Finally, we check the correctness of the number translation by comparing the number to that in the input. We account for various forms of the number (*e.g.*, for the number 5, we also consider 5.0 and five as correct translations) so as to reduce the false positives. We also manually examine the incorrect translations to ensure they are not false positives.

A.2 Create (Transfer) Test Cases to a New Domain

Our testing framework facilitates constructing test instances for new domains in the following steps:

1. Obtain a large corpus of text that contains numbers (*e.g.*,CommonCrawl);
2. Check if there is a number in the output translation;
3. If so, then test if the output number is the correct "translation" for the number in the source sentence;
4. Use instances that pass this test as templates for switching in our different numbers.

Capability	Template	Count	Remarks
Integers e.g., 5	1. As of March 28, 2020, a total of [NUM] laboratory-confirmed COVID-19 cases (Figure) were reported to CDC. 2. Case report forms were submitted to CDC for [NUM] cases. 3. UNESCO estimates [NUM] learners are potentially at risk (pre-primary to upper-secondary education). 4. There have been [NUM] confirmed cases of COVID19 to date. 5. CNBC reported there were at least [NUM] cases worldwide.	50	Ranging from 1 to 10 digits
Decimals e.g., 7.14	1. An average of [NUM] people is infected every day due to this disease 2. The distance between Sydney to Washington is [NUM] miles 3. The genome size of the coronavirus is approximately [NUM] 4. At this point, Rosberg was about [NUM] seconds behind his teammate. 5. The reproduction number of this disease is between [NUM] and [NUM].	40	Ranging from 1 to 4 decimal position
Numerals e.g., five hundred	1. The total amount of remittance is [NUM]. 2. Case report forms were submitted to CDC for [NUM] cases. 3. As of 8 April 2020, approximately [NUM] learners have been affected due to school closures in response to COVID-19. 4. As of December 2019, [NUM] cases of MERS-CoV infection had been confirmed by laboratory tests 5. They then planned an ambitious open-air concert in Tokyo, with a stage costing [NUM] dollars US.	40	{hundred, thousand, million, trillion}
Separators e.g., 12,230	1. An average of [NUM] people is infected every day due to this disease 2. The distance between Sydney to Washington is [NUM] miles 3. The genome size of the coronavirus is approximately [NUM] 4. They then planned an ambitious open-air concert in Tokyo, with a stage costing [NUM] dollars US. 5. As of December 2019, [NUM] cases of MERS-CoV infection had been confirmed by laboratory tests	35	Ranging from 4 to 10 digits

Table 5: Summary of test examples used in our behavioural testing of NMT systems in translating numbers between English and {German, Chinese, Nepali, Tamil}.

A.3 Test Examples Details

Table 5 shows the details of the test examples used in our behavioural testing, including the templates used and the types of digits and numerals generated to fill in the templates.