Panlingua-KMI MT System for Similar Language Translation Task at WMT 2019

Atul Kr. Ojha[!], **Ritesh Kumar**[+], **Akanksha Bansal**[!], **Priya Rani**[+]
[!]Panlingua Language Processing LLP, New Delhi, [+]Dr. Bhimrao Ambedkar University, Agra
(shashwatup9k,akanksha.bansal15,pranijnu)@gmail.com, ritesh78_llh@jnu.ac.in

Abstract

The present paper enumerates the development of Panlingua-KMI Machine Translation (MT) systems for Hindi ↔ Nepali language pair, designed as part of the Similar Language Translation Task at the WMT 2019 Shared Task. The Panlingua-KMI team conducted a series of experiments to explore both the phrase-based statistical (PBSMT) and neural methods (NMT). Among the 11 MT systems prepared under this task, 6 PBSMT systems were prepared for Nepali-Hindi, 1 PBSMT for Hindi-Nepali and 2 NMT systems were developed for Nepali↔ Hindi. The results show that PBSMT could be an effective method for developing MT systems for closely-related languages. Our Hindi-Nepali PBSMT system was ranked 2nd among the 13 systems submitted for the pair and our Nepali-Hindi PBSMT system was ranked 4th among the 12 systems submitted for the task.

1 Introduction

Automated translation between languages from the same family is a challenging task. While similarity among language pairs may seem to be an advantageous situation in terms of the possibility of developing better performing machine translation systems even with low quantity of resources (like low volume of parallel data), the challenge is to figure out how exactly the advantage can be leveraged and what could be the best method to do it.

The area of Statistical Machine Translation (SMT) has witnessed a continuous rise for the last two decades. The availability of open source toolkits, like Moses (Koehn et al., 2007), have also provided it an impetus. However, neural models have garnered much attention in recent times as they provide robust solutions to machine translation tasks. Their popularity is heightened further with the availability of Neural Machine Translation (NMT) open source toolkits such as OpenNMT (Klein et al., 2017), which provides an almost out-of-the-box solution for developing the first NMT systems as well as experimenting with different kinds of architectures and hyper-parameters (which is crucial for developing a good NMT system). Keeping in view the recent results obtained in MT developments, we experimented with both PBSMT as well as NMT models and evaluated how different models perform in comparison to each other. In general, NMT systems are extremely data-hungry and require huge amounts of parallel data to give a good system. The team was motivated to know if NMT could perform better than PBSMT systems even with low volume of data and without making use of monolingual data when the language pairs were closely-related.

Thus, the broad objectives behind conducting these experiments were,

a) to compare the performance of SMT and NMT in case of closely-related, relatively low-resourced language pairs, and

b) to find how SMT can be made to perform better for closely-related language pairs.

2 System Overview

This section provides an overview of the systems developed for the WMT 2019 Shared Task. In these experiments, the Panlingua-KMI team explored both phrase-based statistical (Koehn et al., 2003) method and neural method for Nepali-Hindi and Hindi-Nepali language pairs. For this purpose, 11 MT systems were developed including 6 Phrase-based Statistical Machine Translation (PBSMT) for Nepali-Hindi, 1 PBSMT for Hindi-Nepali, 2 NMT for Nepali-Hindi and 2 NMT for

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 213–218
Florence, Italy, August 1-2, 2019. ©2019 Association for Computational Linguistics

Hindi-Nepali. The system details are provided in the following subsections.

2.1 Phrase-based SMT Systems

These systems were built on the Moses open source toolkit using the KenLM language model (Heafield, 2011) and GIZA++ aligner. 'Grow-diag-final-and heuristic' parameters were used to extract phrases from the corresponding parallel corpora. In addition to this, KenLM was used to build 5-gram language models. The pre-processing was done to handle noise in data (for example, hyperlink, non-UTF characters etc), the details of which are provided below in section 3.1.

2.2 Neural Machine Translation System

OpeNMT (pytorch port of this toolkit) was used to build 2 NMT systems. The first system was built with 2 layers using LSTM model while the second system was built with 6 layers using the Transformer model. 500 hidden units were used.

2.3 Assessment

Assessment of these systems was done on the standard automatic evaluation metrics: BLEU (Papineni et al., 2002) and Translation Error Rate (TER) (Snover et al., 2006). TER was evaluated only for systems whose BLEU score was above 5. In addition to these, the errors of the developed systems were also analysed.

3 Experiments

This section briefly describes the experiment settings for developing the systems.

3.1 Corpus Size

The parallel dataset for these experiments was provided by the *WMT Similar Translation Shared Task* [1]organisers and the Nepali monolingual dataset was taken from *WMT 2019 Shared Task: Parallel Corpus Filtering for Low-Resource Conditions*[2] (Barrault et al., 2019). The monolingual dataset for Hindi was procured from *Workshop on Asian Translation Shared Task 2018* (Nakazawa et al., 2018). The parallel data was sub-divided into training, tuning and monolingual sets, as detailed in Table 1.

Nepali-Hindi and Hindi-Nepali MT systems were tested on 2,000 and 1567 test sentences respectively.

Language Pair	Training	Tuning	Monolingual
Nepali↔ Hindi	65505	3000	-
Nepali	-	-	92296
Hindi	-	-	104967

Table 1: Statistics of Parallel and Monolingual Sentences of the Nepali and Hindi Languages

3.2 Pre-processing

The following pre-processing steps were performed as part of the experiments:

a) Both corpora were tokenized and cleaned (sentences of length over 40 / 80 words were removed).

b) True-casing of Latin characters in the corpora was performed. Even though neither of the language pairs use Latin-based scripts, this was needed as the corpora for training as well as testing contained some Latin characters as well.

All these processes were performed using Moses scripts. However, the tokenization was done by the RGNLP team tokenizer (Ojha et al., 2018). This tokenizer was used since Moses does not provide tokenizer for Indic languages. Also the RGNLP tokenizer ensured that the canonical Unicode representation of the characters are retained.

3.3 Development of MT Systems

The pre-processed dataset was used to develop three MT models per language pair – two different phrase-based statistical machine translation system using different language models and one neural MT system using the encoder-decoder framework. Both of these are discussed in the following subsections.

3.3.1 Training and Development of PBMST Systems

As mentioned above, we used the Moses open source toolkit for the development of the PBSMT system. The translation model (TM) and language models (LM) were trained independently and combined in a log-linear scheme where both the models were assigned a different weight using the Minimum Error Rate (MERT) Training tuning algorithm (Och and Ney, 2003). In addition, 3,000

[1] http://www.statmt.org/wmt19/similar.html

[2] http://www.statmt.org/wmt19/parallel-corpus-filtering.html

parallel sentences were used for Nepali-Hindi and Hindi-Nepali language pairs to tune the systems.

The details of the experiments are as follows:

I) Nepali-Hindi PBSMT - 6 different experiments (3 each for dataset with sentences of length up to 40 words and 80 words) were conducted for Nepali-Hindi PBSMT system. The difference among the experiments were only with respect to pre-processing alterations. It was used to gauge the effect of different pre-processing steps on the performance of MT system for closely-related languages. The following pre-processing alternations were used -

   a experiments without lowercasing

   b experiments without removing utterances with non-UTF characters

   c experiments with complete pre-processing including lowercasing and getting rid of utterances with non-UTF characters.

II) Hindi-Nepali PBSMT - Based on our experience with Nepali-Hindi system, we developed only one system for Hindi-Nepali pair, using the dataset with complete pre-processing including lowercasing and getting rid of utterances with non-UTF characters.

3.3.2 Training and Developments of NMT Systems

The OpenNMT toolkit was used to develop the NMT systems. The training was done on two layers of LSTM network with 500 hidden units at both, the encoder and decoder models for 1,00,000 epochs. The variability of the parameters was limited with the use of default hyper-parameters configuration. Any unknown words in the translation were replaced with the word in the source language bearing the highest attention weight. All the NMT experiments were carried out only with a dataset that contained sentences with length of up to 40 words.

The hyper-parameters and details of the architecture used for the experiments are as below.

a **LSTM Model -** This system was built using 2-layer LSTM model (Hochreiter and Schmidhuber, 1997). Our settings followed the Open-NMT training guidelines that indicate that the default training setup is reasonable for training of any language pairs. The model is trained on 1,00,000 epochs, using Adam with a default learning rate of 0.002 and mini-batches of 40 with 500 hidden units. Vocabulary size of 32308 and 32895 for Nepali-Hindi and Hindi-Nepali language pairs respectively was extracted. A static NMT-setup was maintained with the use of same hyper-parameters setting across two language pairs.

b **Transformer Model -** Another NMT system was developed using the Transformer model (implemented in pytorch port of OpenNMT) with 6 layers. The Nepali-Hindi system was trained for 20,000 epochs and Hindi-Nepali for 10,000 epochs. All other hyper-parameters were kept at default values in the OpenNMT implementation.

3.4 Post-processing

In the end, the translations of the test data using PBSMT systems were post-processed using methods including de-tokenization, de-truecasing for English tokens to improve the accuracy rate of the translated outputs.

4 Evaluation and Error Analysis

This section discusses the results of automatic evaluation, human evaluation, and comparative analysis of the PBSMT and NMT systems.

4.1 Automatic Evaluation Results

Both the PBSMT and NMT systems were evaluated using the reference set provided by the shared task organizers. The standard MT evaluation metrics, BLEU (Papineni et al., 2002) score and TER (Snover et al., 2006), were used for the automatic evaluation. These results were prepared on the Primary and Contrastive system submission which are depicted in the graph provided below as *_P and *_C, where P stands for Primary and C stands for Contrastive, respectively. The results of only the highest scoring system across both language pairs are presented in this paper. It gives a quantitative picture of particular differences across different teams, especially with reference to evaluation scores (Figure 1 and 2).

The Panlingua-KMI PBSMT system produced fourth and second best results for Nepali-Hindi and Hindi-Nepali language pair respectively, across 6 teams and 12-13 systems. Also for PBSMT systems, the Hindi-Nepali language pair

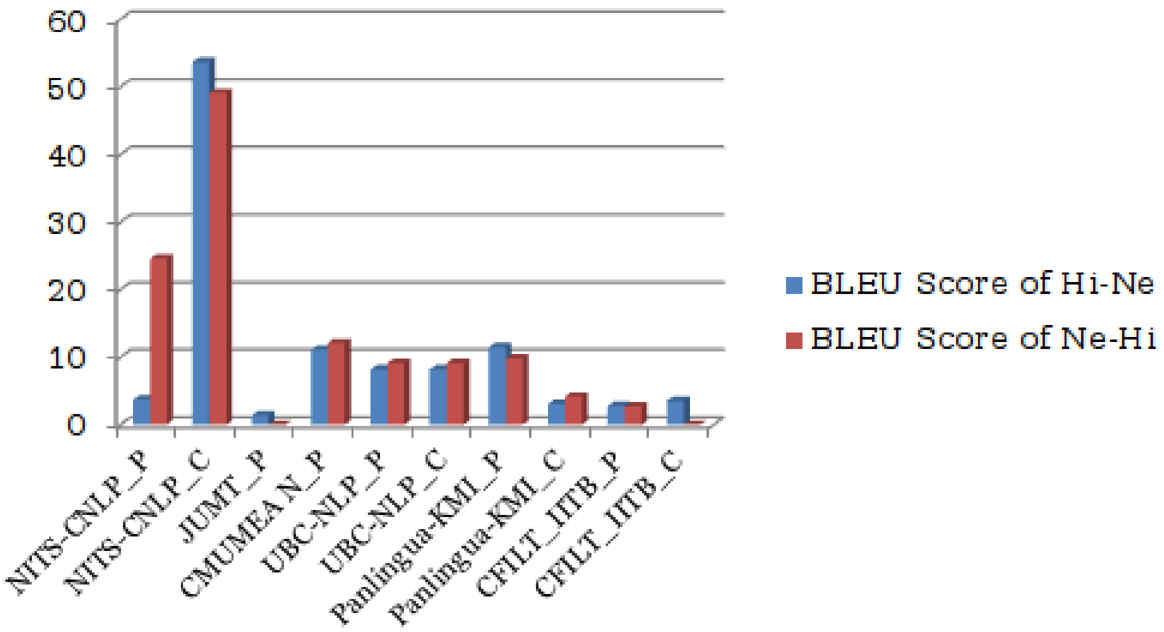


Figure 1: Accuracy of Nepali-Hindi and Hindi-Nepali MT System at BLEU Metric

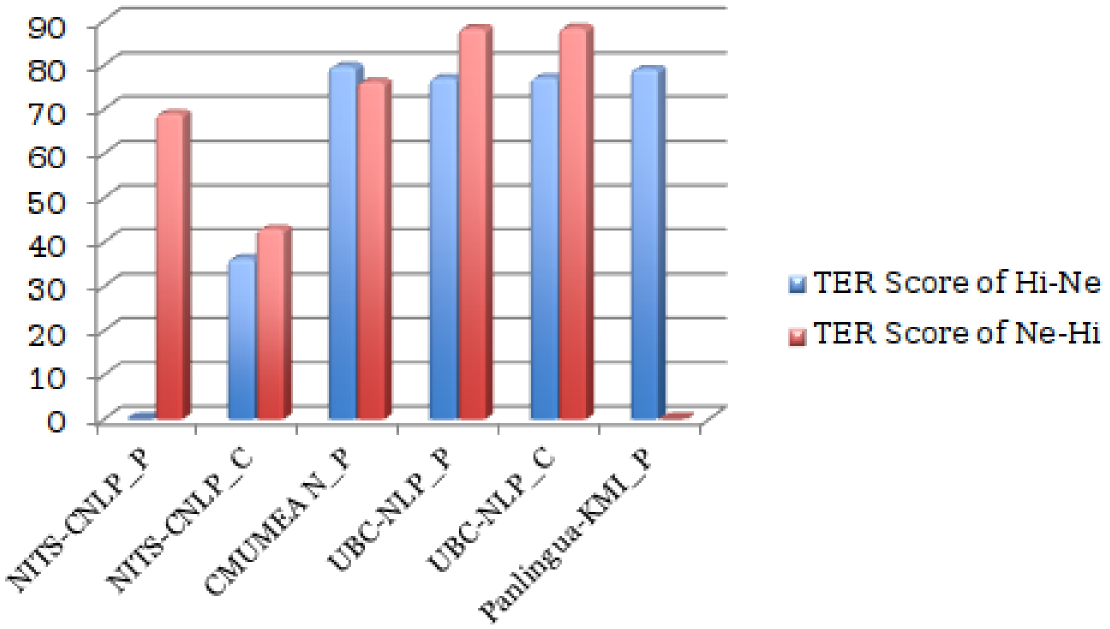


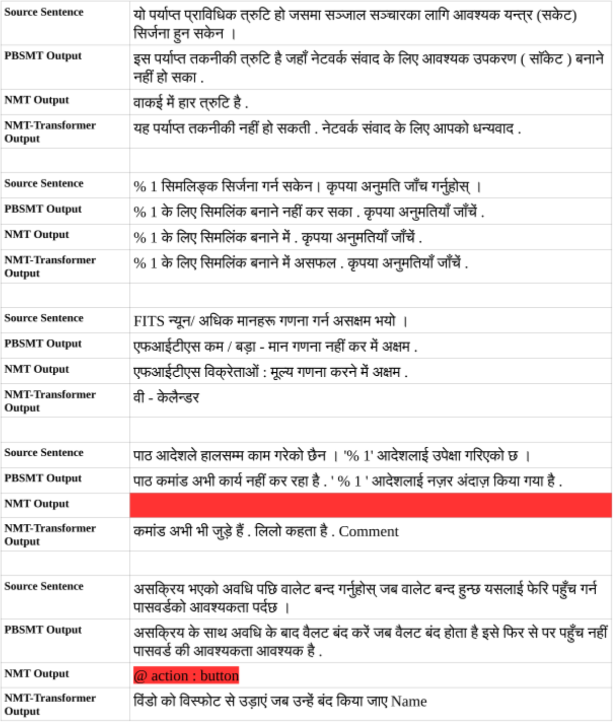
Figure 2: Accuracy of Nepali-Hindi and Hindi-Nepali MT System at TER Metric

showed better results in terms of both the metrics (11.5 in BLEU, 79.1 in TER) while the Nepali-Hindi language pair scored 9.8 in BLEU, 91.3 in TER.

4.2 Comparative Analysis of the PBSMT and NMT Systems

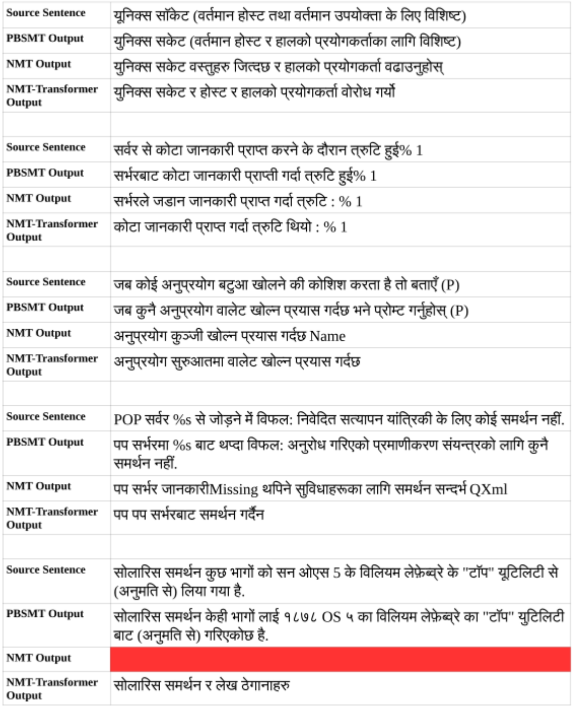
Across both the language pairs, PBSMT performed better than NMT as its accuracy rate was higher in BLEU and lower in TER metrics as shown in Figures 1 and 2. On further manual inspection of the outputs produced by Nepali-Hindi and Hindi-Nepali PBSMT, NMT-LSTM and

NMT-Transformer systems, we found that the outputs produced by the PBSMT seemed better than those produced by the NMT systems (shown in Figures 3 and 4).



Source Sentence	यो पर्याप्त प्राविधिक त्रुटि हो जसमा सञ्जाल सञ्चारका लागि आवश्यक यन्त्र (सकेट) सिर्जना हुन सकेन ।
PBSMT Output	इस पर्याप्त तकनीकी त्रुटि है जहाँ नेटवर्क संवाद के लिए आवश्यक उपकरण (सॉकेट) बनाने नहीं हो सका .
NMT Output	वाकई में हार त्रुटि है .
NMT-Transformer Output	यह पर्याप्त तकनीकी नहीं हो सकती . नेटवर्क संवाद के लिए आपको धन्यवाद .
Source Sentence	% 1 सिमलिङ्क सिर्जना गर्न सकेन। कृपया अनुमति जाँच गर्नुहोस् ।
PBSMT Output	% 1 के लिए सिमलिंक बनाने नहीं कर सका . कृपया अनुमतियाँ जाँचें .
NMT Output	% 1 के लिए सिमलिंक बनाने में . कृपया अनुमतियाँ जाँचें .
NMT-Transformer Output	% 1 के लिए सिमलिंक बनाने में असफल . कृपया अनुमतियाँ जाँचें .
Source Sentence	FITS न्यून/ अधिक मानहरू गणना गर्न असक्षम भयो ।
PBSMT Output	एफआईटीएस कम / बड़ा - मान गणना नहीं कर में अक्षम .
NMT Output	एफआईटीएस विक्रेताओं : मूल्य गणना करने में अक्षम .
NMT-Transformer Output	वी - केलैन्डर
Source Sentence	पाठ आदेशले हालसम्म काम गरेको छैन । '% 1' आदेशलाई उपेक्षा गरिएको छ ।
PBSMT Output	पाठ कमांड अभी कार्य नहीं कर रहा है . ' % 1 ' आदेशलाई नज़र अंदाज़ किया गया है .
NMT Output	🟥
NMT-Transformer Output	कमांड अभी भी जुड़े हैं . लिलो कहता है . Comment
Source Sentence	असक्रिय भएको अवधि पछि वालेट बन्द गर्नुहोस् जब वालेट बन्द हुन्छ यसलाई फेरि पहुँच गर्न पासवर्डको आवश्यकता पर्दछ ।
PBSMT Output	असक्रिय के साथ अवधि के बाद वैलट बंद करें जब वैलट बंद होता है इसे फिर से पर पहुँच नहीं पासवर्ड की आवश्यकता आवश्यक है .
NMT Output	@ action : button
NMT-Transformer Output	विंडो को विस्फोट से उड़ाएं जब उन्हें बंद किया जाए Name

Figure 3: Comparative Analysis of Nepali-Hindi PBSMT, NMT and NMT-Transformer MT's Output



Source Sentence	यूनिक्स सॉकेट (वर्तमान होस्ट तथा वर्तमान उपयोक्ता के लिए विशिष्ट)
PBSMT Output	युनिक्स सकेट (वर्तमान होस्ट र हालको प्रयोगकर्ताका लागि विशिष्ट)
NMT Output	युनिक्स सकेट वस्तुहरु जित्दछ र हालको प्रयोगकर्ता वढाउनुहोस्
NMT-Transformer Output	युनिक्स सकेट र होस्ट र हालको प्रयोगकर्ता वोरोध गर्यो
Source Sentence	सर्वर से कोटा जानकारी प्राप्त करने के दौरान त्रुटि हुई% 1
PBSMT Output	सर्भरबाट कोटा जानकारी प्राप्ती गर्दा त्रुटि हुई% 1
NMT Output	सर्भरले जडान जानकारी प्राप्त गर्दा त्रुटि : % 1
NMT-Transformer Output	कोटा जानकारी प्राप्त गर्दा त्रुटि थियो : % 1
Source Sentence	जब कोई अनुप्रयोग बटुआ खोलने की कोशिश करता है तो बताएँ (P)
PBSMT Output	जब कुनै अनुप्रयोग वालेट खोल्न प्रयास गर्दछ भने प्रोम्ट गर्नुहोस् (P)
NMT Output	अनुप्रयोग कुञ्जी खोल्न प्रयास गर्दछ Name
NMT-Transformer Output	अनुप्रयोग सुरुआतमा वालेट खोल्न प्रयास गर्दछ
Source Sentence	POP सर्वर %s से जोड़ने में विफल: निवेदित सत्यापन यांत्रिकी के लिए कोई समर्थन नहीं.
PBSMT Output	पप सर्भरमा %s बाट थप्दा विफल: अनुरोध गरिएको प्रमाणीकरण संयन्त्रको लागि कुनै समर्थन नहीं.
NMT Output	पप सर्भर जानकारीMissing थपिने सुविधाहरूका लागि समर्थन सन्दर्भ QXml
NMT-Transformer Output	पप पप सर्भरबाट समर्थन गर्दैन
Source Sentence	सोलारिस समर्थन कुछ भागों को सन ओएस 5 के विलियम लेफ़ेब्व्रे के "टॉप" यूटिलिटी से (अनुमति से) लिया गया है.
PBSMT Output	सोलारिस समर्थन केही भागों लाई १८७८ OS ५ का विलियम लेफ़ेब्व्रे का "टॉप" युटिलिटी बाट (अनुमति से) गरिएकोछ है.
NMT Output	🟥
NMT-Transformer Output	सोलारिस समर्थन र लेख ठेगानाहरु

Figure 4: Comparative Analysis of Hindi-Nepali PBSMT, NMT and NMT-Transformer MT's Output

NMT's result was affected primarily due to over-generation, NER issues, OOV (Out-of-Vocabulary), and word-order, hence, unable to provide output of 27 source sentences for Nepali-Hindi and 12 source sentences for Hindi-Nepali. The PBSMT's results were also influenced by the above-mentioned factors, but despite that, output of each source sentence was produced.

5 Conclusion

The entire series of experiments revealed several aspects of developing NMT system for closely-related languages. It may seem that NMT performs better than SMT on fluency level (3 and 4) but the relation between source and target language is erroneous, thereby, resulting in poor BLEU score and higher TER. Also, alterations at pre-processing stage do not render any improvement in SMT systems, thus, strengthening the importance of lower casing and excluding non-UTF characters from the data sets. It was also observed that datasets with maximum length of sentences upto 40 words performed better than those with upto 80 words.

The larger picture, based on these experiments, reveal that similarities between two languages did not yield any advantage, as expected at the initial stage. Thus it could be concluded that similar features shared between two languages do not have any significant effect on the performance of the MT systems, at least, as long as the standard methodologies are employed for developing the systems. In order to make use of the similarity in between the language pairs, some more sophisticated methods need to be explored and is a matter of further research.

Acknowledgments

We are grateful to the organizers of WMT Similar Translation Shared Task 2019 for providing us the Nepali-Hindi Parallel Corpus and evaluation scores. We would also like to acknowledge the WMT 2019 Corpus Filtering Shared Task and WAT 2018 for releasing Nepali and Hindi monolingual corpus respectively.

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, W. P. Pa, Isao Goto, Hideya Mino, K. Sudoh, and Sadao Kurohashi. 2018. Overview of the 5th workshop on asian translation. In *Proceedings of the 5th Workshop on Asian Translation (WAT2018)*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Atul Kr Ojha, Koel Dutta Chowdhury, Chao-Hong Liu, and Karan Saxena. 2018. The rgnlp machine translation systems for wat 2018. In *Proceedings of the 5th Workshop on Asian Translation (WAT2018)*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

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*, volume 200.