Detecting Untranslated Content for Neural Machine Translation

Isao Goto **Hideki Tanaka**
NHK Science & Technology Research Laboratories,
1-10-11 Kinuta, Setagaya-ku,
Tokyo 157-8510, Japan
goto.i-es@nhk.or.jp tanaka.h-ja@nhk.or.jp

Abstract

Despite its promise, neural machine translation (NMT) has a serious problem in that source content may be mistakenly left untranslated. The ability to detect untranslated content is important for the practical use of NMT. We evaluate two types of probability with which to detect untranslated content: the cumulative attention (ATN) probability and back translation (BT) probability from the target sentence to the source sentence. Experiments on detecting untranslated content in Japanese–English patent translations show that ATN and BT are each more effective than random choice, BT is more effective than ATN, and the combination of the two provides further improvements. We also confirmed the effectiveness of using ATN and BT to rerank the n-best NMT outputs.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015) outputs fluent translations. However, some of the source content—not only word-level expressions but also clause-level expressions—is sometimes missing from the output translation, especially when NMT translates long sentences. An example is shown in Figure 1. The occurrence of untranslated content is a serious problem limiting practical use of NMT.

Conventional statistical machine translation (SMT) (Koehn et al., 2003; Chiang, 2007) explicitly distinguishes the untranslated source words from the translated source words in decoding and keeps translating until no untranslated source words remain. However, NMT does not explicitly distinguish untranslated words from translated words. This means NMT cannot use coverage vectors as are used in SMT to prevent translations from being dropped.

There are methods that use dynamic states, which are regarded as a soft coverage vector, at each source word position (Tu et al., 2016b; Mi et al., 2016). These methods will alleviate the problem; however, they do not decide whether to terminate decoding on the basis of the detection of untranslated content. Therefore, the translation dropping problem remains.

We evaluated two types of probability for detecting untranslated content. One type is the cumulative attention (ATN) probability for each source position. The other type is the back translation (BT) probability of each source word from the MT output. The latter type does not necessarily require word-level correspondences between languages, which are not easy to infer precisely in NMT. We also compared direct use of the probabilities and the use of the ratio of the probabilities, which compares the negative logarithm of a probability to the minimum value of the negative logarithm of the probability in the n-best outputs. In addition, we evaluated the effect of using detection scores to rerank the n-best outputs of NMT.

We conducted experiments for the detection of untranslated source content words in 100 sentences with MT outputs translated using NMT on Japanese–English patent translation task data sets. The results are as follows. The detection accuracies achieved using the ratio of probabilities were higher than those achieved directly using the probabilities. ATN and BT are each more effective than random choice at detecting untranslated content. BT was better than ATN. The detection accuracy further improved when ATN and BT were used together. Reranking using the scores of the two types of probabilities improved the BLEU scores. BLEU scores improved further when the detection

Proceedings of the First Workshop on Neural Machine Translation, pages 47–55,
Vancouver, Canada, August 4, 2017.

Input	その後、 第 1 段 から 順に 第 M 段 まで 、 ADC # 1 と ADC # 2 の パイプ ライン ゲイン エラー 補正 を 交互に 繰り返す （ ステップ S 6 と S 7 、 ステップ S 8 と S 9 、 ステップ S 10 と S 11 ） 。
Reference	After that , the correction of a pipeline gain error of ADC # 1 and ADC # 2 is sequentially repeated alternately from the first stage to the Mth stage (steps S6 and S7 , steps S8 and S9 , steps S10 and S11) .
Output	After that , the pipeline gain error correction of the ADC # 1 and the ADC # 2 is alternately repeated (steps S6 and S7 , steps S8 and S11) .

Figure 1: Example of untranslated content in Japanese–English translation by NMT. The shaded parts in the input were mistakenly not translated. The shaded parts in the reference are the corresponding translations of the untranslated parts.

scores of the two types of probabilities were used together. We counted the number of untranslated content words in 100 sentences and found that the untranslated content in the reranked outputs was less than that in the baseline NMT outputs.

2 Neural Machine Translation

We briefly describe the baseline attention-based NMT based on previous work (Bahdanau et al., 2015) that we used. The NMT consists of an encoder that encodes a source sentence and a decoder that generates a target sentence.

Given an input sentence, we convert each word into a one-hot vector and obtain a one-hot vector sequence $\mathbf{x} = x_1, \ldots, x_{Tx}$. The encoder produces a vector $h_j = [\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top]^\top$ for each source word position j using long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and the word embedding matrix E_x for the source language. $\overrightarrow{h}_j = f(\overrightarrow{h}_{j-1}, E_x x_j)$ is the vector output by the forward LSTM, where f is the LSTM function, and $\overleftarrow{h}_j = f(\overleftarrow{h}_{j+1}, E_x x_j)$ is the vector output by the backward LSTM.

The decoder calculates the probability of a translation $\mathbf{y} = y_1, \ldots, y_{Ty}$ given $\mathbf{x}$, where y_i is also a one-hot vector at a target word position i. The decoder searches $\hat{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}}\, p(\mathbf{y}|\mathbf{x})$ to output $\hat{\mathbf{y}}$. The probability is decomposed into the product of the probabilities of each word:

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|y_1, \ldots, y_{i-1}, \mathbf{x}). \quad (1)$$

Each conditional probability on the right-hand side is modeled as

$$p(y_i|y_1, \ldots, y_{i-1}, \mathbf{x}) = \mathrm{softmax}(y_i^\top W_\mathrm{t} t_i), \quad (2)$$

$$t_i = \mathrm{maxout}(U_\mathrm{s} s_i + U_\mathrm{y} E_\mathrm{y} y_{i-1} + U_\mathrm{c} c_i), \quad (3)$$

where s_i is a hidden state of the LSTM, c_i is a context vector, $W_\cdot$ and $U_\cdot$ represent weight matrices, and E_y is the word embedding matrix for the target language. The state s_i is calculated as $s_i = f(s_{i-1}, [c_i^\top ; E_\mathrm{y} y_{i-1}^\top]^\top)$, where f is the LSTM function. The context vector c_i is calculated as a weighted sum of h_j: $c_i = \sum_j \alpha_{i,j} h_j$, where

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_j \exp(e_{i,j})}, \quad (4)$$

$$e_{i,j} = v^\top \tanh(W_\mathrm{s} s_{i-1} + W_\mathrm{y} E_\mathrm{y} y_{i-1}). \quad (5)$$

v is a weight vector.

$\alpha_{i,j}$ represents the attention probability, which can be regarded as a probabilistic correspondence between y_i and x_j to some extent.

3 Detection of Untranslated Content

We describe the two types of probabilities and their use in detecting untranslated content.[1]

3.1 Cumulative Attention Probability

Heavily attended source words would have been translated, while sparsely attended source words would not have been translated (Tu et al., 2016b). Therefore, the ATN probabilities for each source word position should provide clues to the detection of untranslated content. Using Equation (4), we define an ATN probability score (ATN-P) a_j, which represents a score of missing the content of x_j from $\mathbf{y}$, as

$$a_j = -\log\Big(\sum_i \alpha_{i,j}\Big). \quad (6)$$

The value[2] in parentheses in Equation (6) is the ATN probability at the source position j in $\mathbf{x}$. i represents a target word position in $\mathbf{y}$.

[1] The use of their combination is explained in Section 5.2.

[2] Adding a small positive value ϵ to the value is a practical solution of avoiding calculating $\log(0)$. In our experiments, there was no such case and we did not add ϵ.

However, some source words do not inherently correspond to any target word[3], and one source word may correspond to two or more target words. Therefore, a_j does not always correctly represent the degree of missing the content of x_j.

We solve this problem as follows. We define an ATN ratio score (ATN-R), which is based on a probability ratio. Here, the n-best outputs are represented as $\mathbf{y}^1, \ldots, \mathbf{y}^n$. Furthermore, we make the following assumption.

Assumption: Existence of translations

The translation of an arbitrary input word $x_j, (1 \leq j \leq T_x)$ exists somewhere in the n-best outputs $\mathbf{y}^d, (1 \leq d \leq n)$, except when x_j does not inherently correspond to any target words.

Accordingly, we regard $\min_d a_j^d$ as a score without missing a translation, where a_j^d represents a_j for y^d. The ATN-R r_j^d, which represents a score of dropping the content of x_j from $\mathbf{y}^d$, is defined as

$$r_j^d = a_j^d - \min_{d'}(a_j^{d'}) \tag{7}$$

This value represents the logarithm of the probability ratio.

3.2 Back Translation Probability

We define BT as the forced decoding from an MT output to its input sentence. When the content of a source word is missing in the MT output, the BT probability of the source word is expected to be small. We use this expectation as a clue for detecting untranslated content. A detection method based on the BT probability has the feature that the method does not require the specification of word-level correspondences between languages, which is not easy to infer precisely. Here, we present a BT probability score (BT-P) b_j^d based on the BT probability of x_j from $\mathbf{y}^d$ as

$$b_j^d = -\log(p(x_j|x_1, \ldots, x_{j-1}, \mathbf{y}^d)). \tag{8}$$

The probability in Equation (8) is calculated using the NMT method described in Section 2.

We again employ the assumption of the "existence of translations" in the previous section and accordingly $\min_d(b_j^d)$ is the score of an output that contains the content of x_j. With this, we calculate a score based on a probability ratio. We define the BT ratio score (BT-R) q_j^d, which is a score of missing the content of x_j from $\mathbf{y}^d$, as

$$q_j^d = b_j^d - \min_{d'}(b_j^{d'}). \tag{9}$$

4 Application to Translation Scores

The scores described in the previous section will contribute to the selection of a better output (i.e., one that has less untranslated content) from the n-best outputs. We evaluated the effect of reranking using these scores.

As a sentence score for reranking, we use the weighted sum of the output score and the detection score with a weight β:

$$\log(p(\mathbf{y}^d|\mathbf{x})) - \beta \sum_j r_j^d. \tag{10}$$

We subtract r_j^d, which is a score of missing the content of x_j, from the likelihood of the translation. When q_j^d is used, we replace r_j^d with q_j^d. Because reranking compares the n-best outputs of the same input, the reranking results of ATN-R and those of ATN-P are the same.[4] In the same manner, the reranking results of BT-R and those of BT-P are the same. In what follows, we use ATN-R and BT-R.

When r_j^d and q_j^d are used together, we use the score

$$\log(p(\mathbf{y}^d|\mathbf{x})) - \gamma \sum_j r_j^d - \lambda \sum_j q_j^d, \tag{11}$$

where γ and λ are weight parameters.

5 Experiments

As translation data sets including long sentences, we chose Japanese–English patent translations. We conducted experiments to confirm the effects of the scores on the detection of untranslated content and the effects on translation.

5.1 Common Setup

We used the NTCIR-9 and NTCIR-10 Japanese-to-English translation task data sets (Goto et al., 2011; Goto et al., 2013). The number of parallel sentence pairs in the training data was 3.2M. We

[3]For example, articles in English do not usually correspond to any words in Japanese.

[4]However, the results differ when we rank translations among input sentences. The following is an example of such a situation. The translations of many input sentences are ranked and the bottom translations are replaced with the outputs of SMT to reduce missing translation.

used sentences that were 100 words or fewer in length in the training data for Japanese to English (JE) translation. We used sentences that were 50 words or fewer in length in the training data for BT to reduce computational costs. We did not use any monolingual corpus. We used development data consisting of 1000 sentence pairs, which were the first half of the official development data. The numbers of test sentences were 2000 for NTCIR-9 and 2300 for NTCIR-10. We used the Stepp tagger[5] as the English tokenizer and Juman 7.01[6] as the Japanese tokenizer.

We used Kyoto-NMT (Cromieres, 2016) as the NMT implementation and modified it to fit Equation (5). The following settings were used. The most-frequent 30K words were used for both source and target words, and the remaining words were replaced with a special token (UNK). The numbers of LSTM units of the forward and backward encoders were each 1000, the number of LSTM units of the decoder was 1000, the word embedding sizes for the source and target words were each 620, and the size of the vector just before the output layer was 500. The number of hidden layer units and the sizes of the embedding/weight/vocabulary were the same as in (Bahdanau et al., 2015). The mini-batch size for training was 64 for JE and 128 for BT. We used Adam (Kingma and Ba, 2014) to train the NMT models. We trained the NMT models for a total of six epochs. The development data were used to select the best model during the training. The decoding involved a beam search with a beam width of 20. We limited the output length to double the length of the input. We used all of the outputs[7] from the beam search as the n-best outputs.[8]

β, γ, and λ in Section 4 were selected from $\{0.1, 0.2, 0.5, 1, 2\}$ using the development data such that the BLEU score was the highest.

5.2 Detecting Untranslated Content

We translated the NTCIR-10 test data from Japanese into English using the baseline NMT system and manually specified untranslated source parts. We then compared the effects of the scores in Section 3 on the detection of untranslated content.

Setup

We prepared the evaluation data as follows. Employing NMT, we translated NTCIR-10 test data whose lengths and reference lengths were each 100 words or fewer.[9] We used the best outputs from the beam search for each test sentence. To pick up translations including untranslated content, we sorted the translations on the basis of $(\text{translation length})/\min(\text{input length, reference length})$ in ascending order. We then selected 100 sentences from the top and identified 632 untranslated content words in the 100 selected sentences, which consisted of 4457 words. The 632 identified words were used as the gold data. In this process, we removed the sentences from the selected sentences when we could not identify untranslated parts.

Here, we regarded words including Chinese characters, Arabic numerals, katakana characters, or alphabet letters as content words in Japanese. This is because hiragana characters are basically used for functional roles in Japanese sentences. Even if the part-of-speech is a verb, words comprising only hiragana characters (e.g., *suru*) mainly play formal roles and do not contain substantive meaning in most cases for patents and business documents.

When r_j^d and q_j^d were used together, we calculated the detection score

$$\gamma r_j^d + \lambda q_j^d, \tag{12}$$

where γ and λ were those selected in Section 5.1.

Results and Discussion

We ranked words[10] in the 100 selected source sentences on the basis of the scores described in Section 3 and compared them with the gold data (632 words). The results are shown in Figure 2. The average precision of random choice was $0.14 = 632/4457$. The results were as follows.

- ATN-P and BT-P were more effective than random choice.
- ATN-R was better than ATN-P, and BT-R was better than BT-P for the detection.
- Back translation (BT-R) was more effective than cumulative attention (ATN-R).

[5] http://www.nactem.ac.uk/enju/index.html

[6] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN

[7] Word sequences that were terminated with the end of sentence (EOS) tokens.

[8] n was different for each input. n tended to be large when the input lengths were long.

[9] Sentences longer than 100 words were not included in the training data.

[10] More properly, we ranked word positions.

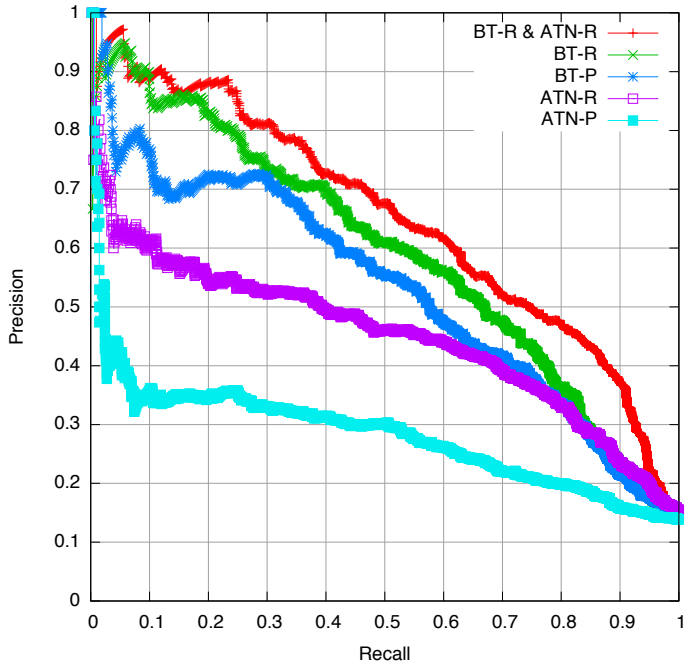

Figure 2: Detection results

Input	**ISO** 感度 値 が 小さい とき に は 増幅 度 が 小さく 、 **ISO** 感度 値 が 大きい とき に は 増幅 度 が 大きい 。
Reference	The amplification is small when the **ISO** sensitivity value is low , while the amplification is large when the **ISO** sensitivity value is high .
Output	When the **ISO** sensitivity value is small , the gain is small .

Figure 3: Unsuccessful example based on BT. Untranslated parts are shaded.

Untranslated content	BT-R	ATN-R
A content word appears only once in an input sentence	Good	Fair
A content word appears twice or more in an input sentence	Bad	Fair

Table 1: Sensitivity of detection of untranslated content.

- The combination of scores (BT-R & ATN-R) was better than the score of each component (BT-R or ATN-R).

Figure 3 shows an unsuccessful example of BT-R. The same content word (ISO) appears twice in the input. It was thus hard to detect the untranslated underlined ISO in the input on the basis of BT-R because the corresponding word (ISO) existed in the output.

On the one hand, the detection sensitivity of BT-P is thought to be high for a content word that appears only once in the input sentence. On the other hand, the detection sensitivity of BT-P is thought to be low for a content word that appears twice or more in the input sentence. Because BT-R is based on BT-P, it has the same characteristics as BT-P. In contrast, ATN-P is sensitive even when a content word appears twice or more in the input sentence because the cumulative probabilities increase depending on the frequency of the word in the MT output. Because ATN-R is based on ATN-P, it has the same characteristics as ATN-P.

Therefore, BT-R and ATN-R are complementary to some extent (Table 1), and this seems to be why the combination works best.

5.3 Reranking the n-best Outputs

We reranked the n-best NMT outputs following Section 4 and assessed the effect on the translation.

Setup

For comparison, we used the baseline NMT system with soft coverage models (Mi et al., 2016; Tu

	NTCIR-10	NTCIR-9
Phrase-based SMT	30.58	30.21
Hierarchical phrase-based SMT	31.99	31.48
NMT Baseline	38.68	37.83
Rerank with ATN-R	39.82	38.88
Rerank with BT-R	40.14	39.16
Rerank with ATN-R & BT-R	40.36	39.46
NMT Baseline with COVERAGE-neural	38.89	37.90
NMT Baseline with COVERAGE-linguistic	39.13	38.03

Table 2: Translation results (BLEU)

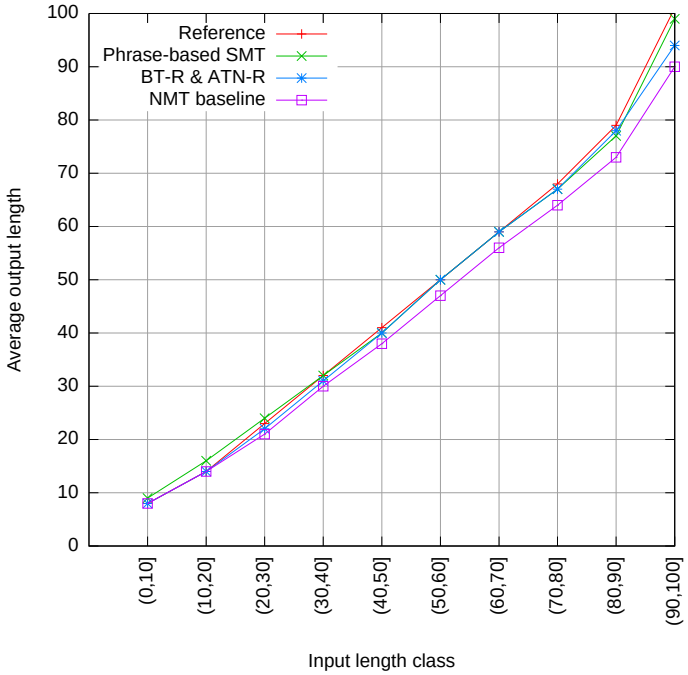

Figure 4: Average output lengths

et al., 2016b), which were used in first-pass decoding.[11] Whereas these studies used gated recurrent units (GRUs) (Chung et al., 2014) for the NMT and coverage models, we used LSTM.[12] The soft coverage model of (Mi et al., 2016) is called a neural soft coverage model (COVERAGE-neural). Tu et al. (2016b) proposed linguistic and neural soft coverage models. We used the linguistic version of (Tu et al., 2016b). We call this model the linguistic soft coverage model (COVERAGE-linguistic).

As references, we used conventional SMT using Moses (Koehn et al., 2007) with a distortion-limit of 20 for phrase-based SMT and a max-chart-span of 1000 for hierarchical phrase-based SMT.

Results and Discussion

Table 2 gives the results measured by case-insensitive BLEU-4 (Papineni et al., 2002). Overall, the results indicate the effectiveness of using ATN probabilities and BT probabilities for translation scores.

We now compare the soft coverage models. Because the difference between the results of the NMT baseline and the results of COVERAGE-neural are small, the effect of COVERAGE-neural was small for this dataset. The difference between the results of the NMT baseline and the results of COVERAGE-linguistic was also small (less than 0.5 BLEU points), whereas the improve-

[11] These methods are not competing but are cooperative because they can be used to produce better n-best outputs.

[12] Our experiments indicated that the BLEU scores of the baseline NMT system using LSTM were higher than those of the baseline NMT system using GRUs. The training time of the neural soft coverage model using the Chainer (Tokui et al., 2015) LSTM for one epoch was shorter than that of the neural soft coverage model using the Chainer GRU.

	Missing translation	Repeated translation
NMT baseline	0.061 (137/2251)	0.004 (9/2251)
Rerank with BT-R & ATN-R	0.020 (45/2251)	0.004 (9/2251)

Table 3: Rate of mistakenly untranslated content words (missing translation) and mistakenly repeated translations. The values in parentheses denote the number of source content words.

ment of COVERAGE-linguistic was greater than that of COVERAGE-neural. In contrast, the results of Rerank with ATN-R obtained improvements of more than 1 BLEU point compared with the NMT baseline. Both the soft coverage models and Rerank with ATN-R are based on attention probabilities. The soft coverage models therefore have room for improvement on this dataset, which means that there is a difficulty in training soft coverage models using end-to-end learning to take advantage of the attention probabilities as well as Rerank with ATN-R. The difficulties would depend on the data sets.[13]

We now compare ATN-R and BT-R. ATN-R and BT-R were effective in reranking. BT-R was slightly better than ATN-R. The combined use of ATN-R and BT-R was more effective than using only one component. These results are consistent with the detection results described in Section 5.2. The difference between reranking with BT-R and reranking with ATN-R & BT-R was statistically significant at $\alpha = 0.01$, which was computed using a tool[14] of the bootstrap resampling test (Koehn, 2004).

[13]We consider possible reasons that the improvements in the BLEU scores achived with the coverage models were not as great as improvements in (Tu et al., 2016b; Mi et al., 2016) as follows. We compare Figure 4 in this paper and Figure 6 in (Tu et al., 2016b) showing the lengths of translations. Contrary to our baseline results, the output lengths of their baseline were much shorter than those of the phrase-based SMT when source sentences were longer than 50 words. This means that there is less missing content for our baseline than for their baseline. We therefore believe the following reasons explain the smaller improvements achieved with the coverage models.

- There is less room for improvement for our baseline with the coverage models than for their baseline.
- Because there is less missing content for our baseline, there are fewer chances that the coverage model effectively improves the translations in our training, which are necessary to appropriately estimate the coverage model parameters. Therefore, the estimation of the coverage model parameters in our training would be more difficult than that in their training.

The second item is thought to be the reason that the improvements for COVERAGE-linguistic, which has fewer parameters, were larger than those for COVERAGE-neural, which has more parameters.

[14]https://github.com/odashi/mteval

We compared the average output lengths using NTCIR-10 test data for the test sentences no longer than 100 words. The average output lengths are shown in Figure 4. The figure shows that the average output lengths of the NMT baseline tend to be shorter than the average reference lengths for long sentences. The average lengths of Rerank with BT-R & ATN-R were longer than those of the NMT baseline, and they were closer to the average reference lengths than those of the NMT baseline.

To check whether the amount of untranslated content was reduced by Rerank with ATN-R & BT-R, we counted untranslated content words in 100 randomly selected test sentences from the NTCIR-10 test data and their translations produced by the NMT baseline and by Rerank with ATN-R & BT-R. We removed sentences from the selected test sentences when the test sentence or its reference sentence was longer than 100 words. Words were regarded as content words when the words met the conditions of content words explained in Section 5.2. The results are presented in Table 3. The results confirm that the amount of untranslated content was reduced by Rerank with ATN-R & BT-R without increasing the amount of mistakenly repeated translations.

6 Related Work

We introduced soft coverage models (Tu et al., 2016b; Mi et al., 2016) in Section 1. In addition to these published studies, there are several parallel related studies on arXiv (Wu et al., 2016; Li and Jurafsky, 2016; Tu et al., 2016a).[15] Wu et al. (2016) use ATN probabilities for reranking. Li and Jurafsky (2016) use BT probabilities for reranking. Tu et al. (2016a) use probabilities of inputs given the decoder states for reranking. Their probabilities are similar to the BT probabilities that we evaluated. However, unlike BT, to calculate

[15]Reviewers for EACL 2017 short paper mentioned Li and Jurafsky (2016) and Wu et al. (2016) in their comments. The neural MT tutorial given at NLP 2017 (Annual meeting of the Association for Natural Language Processing in Japan) introduced Tu et al. (2016a).

their probability, the actual y_i selected in the beam search is not used. These studies did not evaluate the effect on detecting untranslated content and did not assess the effect of combining ATN and BT. In contrast, we evaluated the effect on detecting untranslated content for ATN and BT. In addition, we investigated the effect of combining ATN and BT.

7 Conclusion

We evaluated the effect of two types of probability on detecting untranslated content, which is a serious problem limiting the practical use of NMT. The two types of probabilities are ATN probabilities and BT probabilities. We confirmed their effectiveness in detecting untranslated content. We also confirmed that they were effective in reranking the n-best outputs from NMT. Improvements in NMT will give a better chance of satisfying the assumption of the existence of translations. This is expected to lead to improvements in the detection of untranslated content.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR 2015*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of NIPS 2014 Workshop on Deep Learning, December 2014*.

Fabien Cromieres. 2016. Kyoto-NMT: a neural machine translation implementation in chainer. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 307–311, Osaka, Japan, December.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proceedings of NTCIR-9*, pages 559–578.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013. Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of NTCIR-10*, pages 260–286.

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL HLT 2013*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL Demo and Poster Sessions*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *CoRR*, abs/1601.00372.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas, November.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL 2002*, pages 311–318.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2016a. Neural machine translation with reconstruction. *CoRR*, abs/1611.01874.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016b. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan

Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.