Towards Detecting Errors: Classifying Model-Generated Output in Chat Translation

Yunmeng Li[1], Ryo Fujii[1], Makoto Morishita[2,1], Jun Suzuki[1,3], Kentaro Inui[1,3]
[1]Tohoku University [2]NTT [3]RIKEN
li.yunmeng.r1@dc.tohoku.ac.jp

1 Introduction

With the deepening of globalization, an increasing number of people have to face a problem: how to communicate with others who speak different languages that we do not understand. Compared to hiring a translator, more people tend to use a machine translation system to get what they want to say or what they do not understand. Although machine translation has improved rapidly in news, speeches, and biomedical translation tasks [1, 2], it is still in its infancy in the area of chat translation. It has been pointed out in recent studies that even a document-level system is not entirely qualified for translating chat due to its unique characteristic of multi-speakers [3, 4, 5]. In this research, we focus on figuring out why machine translation models are of low quality when translating chat. In other words, we need to evaluate the model for chat translation to find errors and make improvements for the future.

Considering the fundamental purpose of translation systems, most users are not familiar with the target language and cannot determine whether the translation is accurate. Therefore, we need a solution to confirm whether the translation model fully conveys the interlocutor's meaning to avoid communication problems or misunderstandings. In chat, in addition to the correctness of words and grammar, we also have to pay attention to the gaps in understanding of each other. Hence, we need to determine whether the translation model can generate texts that are well connected to the context and make parties understand each other. To achieve this goal, we build classifiers for evaluating the performance of generated texts. The task is to predict whether a given utterance is generated by the machine translation model (labeled as 'model-generated') or taken from the corpus (labeled as 'original'). By applying these classifiers to the model-generated translations, we can conclude the behaviors of the translation model.

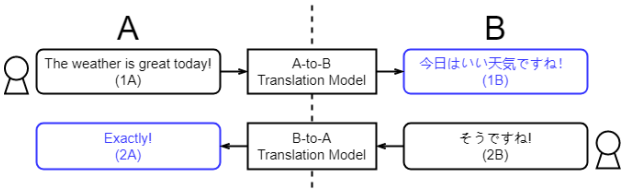

Figure 1 An example of multilingual chat using translation models.

In this research, we focus on chat defined as a two-sentence dialogue between two humans using different languages to simplify the task, assuming both parties do not understand each other's language. Figure 1 is an illustration of multilingual chat of two people with the help of translation models. We trained our models and applied the classifiers on generated translations to find that classifiers had their own merits according to the types of information they had access to. Although they still have some errors, to a large extent, we can use these classifiers to correctly determine most of the chat translations with good cohesion and correct expression.

2 Methods

To confirm the translation model's performance on chat translation, we build a translation model and a series of classification models in this research. After training the model, we generate translations and apply binary classification models to label the given utterance. With the results of classification, we can determine the translation model's performance on chat translation.

2.1 Translation Model

We condition that a chat is composed of two consecutive sentences. Therefore, we choose the 2-to-2 strategy [6] for training, where the inputs and outputs of the model are composed of two texts. With the generated outputs, we continue to train the classification models to create the classifiers.

2.2 Classification Models

We assume that there are two people joining in the chat, each of them speaks different languages and does not understand each other's language. With the assumption, we assign **A** and **B** to be the languages the speakers use. Additionally, we label the sentences of the chat in order as **1** and **2**. Combined the two signs, the first pair of texts is labeled as **1A** and **1B**; the second pair is labeled as **2B** and **2A**. The corresponding labels of the example chat are shown under the texts in Figure 1, and listed in Table 1.

Since applying translation models to both the A-to-B and B-to-A directions will make our task extremely complicated to determine the cause of errors, we simplified the problem by focusing only on the B-to-A side. We assume that all the 1A, 1B, and 2B are from the corpus, and only 2A can be either generated by the model or from the corpus. With the assumption, we can apply different classifiers to 2A and then synthesize the results to determine its quality. In the experiment, we prepare four different classifiers. These four classifiers are trained with 2A together with one or multiple of 1A, 1B, and 2B. To make the classifiers able to distinguish the machine-like translations, we label the data as 'model-generated' to indicate the 2A part is generated by a model; otherwise, we label the data as 'original' to indicate it is from the corpus.

2B-2A For the first classifier, we use 2B and 2A as the training data. This classifier can predict whether a translation model generates 2A, taking 2B as the reference information.

1A-2A For the second classifier, during training, we use 1A and 2A as the training data. This classifier can predict whether a translation model generates 2A, taking 1A as the reference information.

1A-2B-2A For the third classifier, we use 1A, 2B, and 2A as the training data. This classifier takes both 2B and 1A as the reference information.

1A-1B-2B-2A The fourth classifier takes all the 1A, 1B, and 2B as the reference information to predict whether a translation model generates 2A.

With the different parts of the chat as reference information, the four classifiers can determine whether 2A is generated by a model or not. By looking at the results, we can conclude how the reference information contributes to the classification and evaluate the generated translations.

1A	The weather is great today!
1B	今日はいい天気ですね！
2B	そうですね!
2A	Exactly!

Table 1 An example of chat with the labels 1A, 1B, 2B and 2A.

3 Experiment

3.1 Dataset

Our ideal corpus needs to contain a considerable amount of chat between two people who speak different languages, and better to have labels indicating whether the conversations are smooth. Unfortunately, no existing corpus fully meets our requirements. In this research, we resorted to choose OpenSubtitles2018[1] [7, 8, 9] since it has texts closer to chat conversations with multi-speakers' scenes compared with corpora of news, speeches or academic literature.

In the experiment, we selected English and Japanese to be the two languages in the chat. Specifically, we assumed A to be English and B to be Japanese, according to the labels described in section 2.2. Taking 1,000 movie stories from the English-Japanese corpus of OpenSubtitles2018, we split every two consecutive sentences as a pair, and finally obtained 644,000[2] lines of chat as the translation model's training data.

To make the classification models' training data, we re-translated the texts used for training the translation model to obtain negative examples. Though those texts are not necessarily unsuitable as a response to the preceding context in the chat, we assume the idea to label model-generated texts as errors is reasonable to some extent. Actually, due to the low quality of OpenSubtitles, our translation model achieved a BLEU score of no more than 27.2, which we believe is not very high.

As mentioned in section 2.2, what we need to classify is the type of 2A. We took 2A from the model-generated texts and 2A from the corpus, combined different types of original texts as reference to build the four classifiers. Each line of data is labeled as model-generated or original depending on its 2A part. Although the patterns are differ-

1) http://www.opensubtitles.org/

2) 805,000 lines in total, 644,000 for training, 80,500 for validation, and 80,500 for testing.

classifier	validation accuracy
2B-2A	0.8511
1A-2B	0.7678
1A-2B-2A	0.8505
1A-1B-2B-2A	0.8526

Table 2 The accuracy on validation dataset of each classifier after training.

ent, each classification model takes about 1,030,000 lines of data for the training. About 257,600 lines of validation data is used as well to evaluate the learning progress of each model during the training[3)].

For the evaluation, we manually selected 650 lines from the translation model's 80,500 lines of test data to be the test data of the classifiers. These pairs of chat were considered to be smoothly connected. Hence, we could make sure there are no non-chat noises when applying the classifiers to the test data.

3.2 Settings

In the experiment, we used fairseq [10] to build the translation model. Following the descriptions in section 2.1, we designed a 2-to-2 translation model in the direction of B-to-A, specifically, Japanese-to-English. To establish the training data for classification, we used the translation model to re-translate the training data, as mentioned in section 2.2. In this way, we obtained the model-generated 2A that correspond to the original 2A from the corpus.

Based on the languages, we selected the multilingual BERT model [11] as the basis of the classification models. Four classification models were fine-tuned from multilingual BERT through transformers [12], provided by huggingface[4)]. We built the four classifiers: the 2B-2A classifier, the 1A-2A classifier, the 1A-2B-2A classifier, and the 1A-1B-2B-2A classifier, based on the different types of reference data. To track the learning progress, we evaluated the models during the training of the classification models with the validation data. In the test time, the classifiers are to assign the label to the data as model-generated or original, indicating the type of the contained 2A.

3) Meaningless data (i.e., containing too many emojis or garbled texts) were deleted during this process.
4) https://huggingface.co/

	classified into		
classifier	model-generated	original	accuracy
2B-2A	596	54	0.917
1A-2A	585	65	0.900
1A-2B-2A	604	46	0.929
1A-1B-2B-2A	607	43	0.934

Table 3 The classification accuracy and results on the test data containing only **the model-generated 2A.**

	classified into		
classifier	model-generated	original	accuracy
2B-2A	177	473	0.7277
1A-2A	269	381	0.5862
1A-2B-2A	183	467	0.7185
1A-1B-2B-2A	179	471	0.7246

Table 4 The classification accuracy and results on the test data containing only **the original 2A.**

4 Results and Analysis

The accuracy of the classification models on the validation data is listed in Table 2.

We used the 1,300 lines of test data mentioned in section 3.1 to test the four classifiers. The results and accuracy on the separated test data are shown in Table 3 and Table 4. With only the test data of model-generated 2A, each model's performance is relatively similar. However, we notice that all the four classifiers are somehow weak on distinguishing the test data containing the original 2A. We consider the possible reason to be the low quality of OpenSubtitles. Among them, the 1A-2A's performance is worse than others with the test data of original 2A. Combined with the model's accuracy on validation data, we can conclude that the accuracy of 1A-2A is inferior to the remaining three.

When tracking the differences between 2B-2A's results and 1A-2A's results, we find that 1A-2A distinguish 107 more pairs of chat originally from the corpus as model-generated rather than 2B-2A. We consider that 2B-2A behaves better in predicting whether 2A is model-generated compared to 1A-2A. Yet, we conclude 1A-2A is better at distinguishing whether the chat is well-connected while looking at the detailed texts. In the Table 5 with an example chat, it is possible to say that 2A is a good and human-like translation if we only focus on 2B and 2A. However, if we

1A	Yes, I haven't said, have I?
2B	言ってないよ
2A	I didn't.

Table 5 An example of chat with a model-generated 2A.

1A	He is a shithead.
2B	毎年　彼は息子の誕生日を忘れるの
2A	Every year he forgets his birthday.

Table 6 An example of chat with a model-generated 2A.

focus on 1A and 2A, the chat of 1A and 2A is not transparent because the subject of 2A does not match to the preceding context 1A.

But the 1A-2A classifier incorrectly labeled some of the model-generated data as original-from-corpus when looking into the detailed outputs. As shown in Table 6 with an example of chat, if we look at 1A and 2A, the conversation is connected neatly; but the meaning of 2B is translated incorrectly, missing the information of 'his son.' Fortunately, the 1A-2B-2A classifier can take care of it. 1A-2B-2A relatively combines the features of the first two classifiers. Though the example chat in Table 6 is marked as obtained from the corpus via the 1A-2A classifier, it is correctly marked as model-generated through the 1A-2B-2A classifier. We consider that the 1A-2B-2A can distinguish whether 2A is model-generated and whether the translation of 2A is accurate as the same time. Moreover, we suggest that the 1A-2B-2A classifier can indicate the cohesion of the chat and the accuracy of the translation to ensure that the two parties in the conversation can understand each other well.

Observing the 1A-1B-2B-2A and the 1A-2B-2A classifiers' results, we found that 1A-1B-2B-2A can better determine the chat that the second sentence, 2B (or 2A), is shorter. The 1A-2B-2A classifier labeled the chat shown in Table 7 as model-generated, considering 2A is an incorrect translation of 2B. But with the additional reference 1B, the 1A-1B-2B-2A classifier correctly recognized that it was from the corpus. In this type of chat, the second sentence is more often just a response to the first sentence. Hence, the first sentence is more significant, containing further information. When judging 2A, not only 2B but also the preceding context 1B of 2B indirectly determines the quality of 2A. Moreover, due to the difference between English and Japanese grammar rules, fluency is more decisive in chat translation than accuracy when the sentence length is short. However, the difference between the results of 1A-1B-2B-2A and 1A-2B-2A is considerably insignificant. Compared to 1A-2B-2A, 1A-1B-2B-2A only correctly predicted 18 more of the 1300 lines of data originally from the corpus. From this, we believe that the overall performance of 1A-1B-2B-2A and 1A-2B-2A are equivalent. Meanwhile, the performance of 1A-1B-2B-2A is not outstanding with longer sentences. We consider the possible reason to be the weakness of the BERT model since BERT can only assign two token ids to mark the order of sequences in the settings.

1A	I don't believe you.
1B	信じられないな
2B	そう？
2A	No?

Table 7 An example of chat with a 2A originally from the corpus.

5 Conclusions and Future Work

Overall, the translation model's performance in translating chat can be evaluated to a certain extent through the four classifiers established in this research. However, the classifiers still have inaccurate predictions and could not certainly determine the data originally from the corpus on the test data. We consider the reason to be the quality of OpenSubtitles2018. In future research, we will try to find or create a more proper corpus to train our models. We also hope to further predict the translation results by improving our classification models in the future to identify specific problems.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19H04425.

References

[1] 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. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 1–61, 2019.

[2] 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. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pp. 1–55, 2020.

[3] Samuel Läubli, Rico Sennrich, and Martin Volk. 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*, pp. 4791–4796, 2018.

[4] Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 113–123, 2018.

[5] M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, pp. 65–75, 2020.

[6] Jörg Tiedemann and Yves Scherrer. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pp. 82–92, 2017.

[7] Jörg Tiedemann. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3518–3522, 2016.

[8] Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 923–929, 2016.

[9] Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 1742–1748, 2018.

[10] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, pp. 48–53, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45.