Domain Adaptation of Machine Translation with Crowdworkers

Makoto Morishita[1], **Jun Suzuki**[2], **Masaaki Nagata**[1]
NTT Communication Science Laboratories, NTT Corporation[1]
Tohoku University[2]
{makoto.morishita.gr, masaaki.nagata.et}@hco.ntt.co.jp
jun.suzuki@tohoku.ac.jp

Abstract

Although a machine translation model trained with a large in-domain parallel corpus achieves remarkable results, it still works poorly when no in-domain data are available. This situation restricts the applicability of machine translation when the target domain's data are limited. However, there is great demand for high-quality domain-specific machine translation models for many domains. We propose a framework that efficiently and effectively collects parallel sentences in a target domain from the web with the help of crowdworkers. With the collected parallel data, we can quickly adapt a machine translation model to the target domain. Our experiments show that the proposed method can collect target-domain parallel data over a few days at a reasonable cost. We tested it with five domains, and the domain-adapted model improved the BLEU scores to +19.7 by an average of +7.8 points compared to a general-purpose translation model.

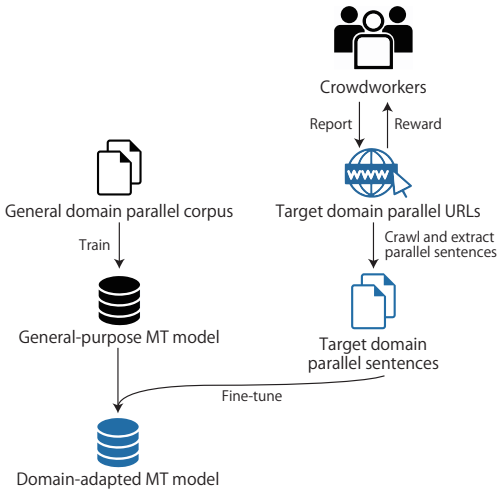

Figure 1: Overview of proposed domain-adaptation method with crowdworkers who collected URLs that included parallel sentences of target domain. We then fine-tuned a general-purpose model with the collected target domain parallel sentences. See Section 3 for details.

1 Introduction

Although recent Neural Machine Translation (NMT) methods have achieved remarkable performance, their translation quality drastically drops when the input domain is not covered by training data (Müller et al., 2020). One typical approach for translating such inputs is adapting the machine translation model to a domain with a small portion of in-domain parallel sentences (Chu and Wang, 2018). Such sentences are normally extracted from a large existing parallel corpus (Wang et al., 2017; van der Wees et al., 2017) or created synthetically from a monolingual corpus (Chinea-Ríos et al., 2017). However, the existing parallel/monolingual data may not include enough sentences relevant to the target domain.

There is a real-world need for a method that can adapt a machine translation model to any domain. For example, users reading or writing in such specific fields as scientific, medical or patent domains, may experience satisfaction if they have access to a domain-adapted machine translation model. Unfortunately, the often limited availability of in-domain parallel data complicates this task. For example, it is difficult to adapt a model to the COVID-19 domain because this issue is too new, and the current available data do not sufficiently cover it.

To alleviate the issue, we propose a method that rapidly adapts a machine translation model to many domains at reasonable costs and time periods with crowdworkers. Fig. 1 shows an overview of our framework. We hypothesize that a small number of in-domain parallel sentences of the target domain are available on the web, and we ask crowdworkers to report these web URLs as a web mining task. Our task does not require translation skills, unlike some previous research (Zaidan and Callison-Burch, 2011; Behnke et al., 2018; Kalimuthu et al., 2019) that attempted manual translations of in-

Proceedings of EMNLP 2022 Industry Track, pages 616–628
December 9–11, 2020.

domain monolingual sentences by crowdworkers. Thus, workers who are not professional translators can participate.

Furthermore, to collect effective parallel sentences, we also vary the crowdworkers' rewards based on the quality of their reported URLs. After collecting parallel sentences by our method, we adapted the machine translation model with the collected, target-domain parallel sentences. Our method has the advantage of being applicable to many domains, in contrast to previous works that use existing parallel/monolingual data.

We experimentally show that our method quickly collects in-domain parallel sentences and improves the translation performance of the target domains in a few days and at a reasonable cost.

Our contributions can be summarized as follows:

- We proposed a new domain-adaptation method that quickly collects in-domain parallel sentences from the web with crowdworkers.
- We empirically showed that crowdworkers are motivated by variable rewards to find more valuable web sites and achieved better performance than under the fixed reward system.

2 Related Work

2.1 Domain Adaptation

Domain adaptation is a method that improves the performance of a machine translation model for a specific domain. The most common method for neural machine translation models is to fine-tune the model with target-domain parallel sentences (Chu and Wang, 2018). Kiyono et al. (2020), who ranked first in the WMT 2020 news shared task (Barrault et al., 2020), fine-tuned a model with a news domain parallel corpus and improved the BLEU scores by +2.2 points. Since the availability of a target-domain parallel corpus is limited, we typically select similar domain sentences from a large parallel corpus (Moore and Lewis, 2010; Axelrod et al., 2011). However, its applicability remains limited because some domains are not covered by existing parallel corpora.

We take a different approach that freshly collects target-domain parallel sentences from the web. Since we do not rely on an existing corpus, our method can be applied to many domains.

2.2 Collecting Parallel Sentences from the Web

Recently, some works successfully built a large-scale parallel corpus by collecting parallel sentences from the web. The BUCC workshop organized shared-tasks of extracting parallel sentences from the web (Sharoff et al., 2015; Zweigenbaum et al., 2017). The ParaCrawl project successfully created a large-scale parallel corpus between English and other European languages by extensively crawling the web (Bañón et al., 2020). Typical bitext-mining projects, including ParaCrawl, took the following steps to identify parallel sentences from the web (Resnik and Smith, 2003): (1) find multilingual websites, which may contain parallel sentences, from the web (Papavassiliou et al., 2018; Bañón et al., 2020); (2) find parallel documents from websites (Thompson and Koehn, 2020; El-Kishky and Guzmán, 2020); (3) extract parallel sentences from parallel web URLs (Thompson and Koehn, 2019; Chousa et al., 2020). Our work focuses on the first step: finding bilingual target-domain web URLs. Bañón et al. (2020) analyzed all of the CommonCrawl data to find crawl candidate websites that contain a certain amount of both source and target language texts. Their method efficiently collected parallel sentences from the web. However, since CommonCrawl only covers a small portion of the web, it may overlook websites that contain valuable resources. Thus, the current web-based corpora (Bañón et al., 2020; Morishita et al., 2020) may not cover all the domains we want to adapt. It is also difficult to focus on a specific topic. In contrast, our work does not rely on CommonCrawl but on crowdworkers who can search the whole web and focus on specific domains.

2.3 Creating Parallel Corpus with Crowdworkers

Some researchers have used crowdsourcing platforms to create new language resources (Roit et al., 2020; Jiang et al., 2018). Some work created a parallel corpus for domain-adaptation by asking crowdworkers to translate in-domain monolingual sentences (Zaidan and Callison-Burch, 2011; Behnke et al., 2018; Kalimuthu et al., 2019). Although this approach is straightforward, it does suffer from several drawbacks. For example, it is often difficult to find a sufficient amount of crowdworkers since translation tasks often require an understanding of both the languages that are actu-

ally being used. Note that although we also use a crowdsourcing platform, our approach entirely differs from the approach introduced in this section, such as asking crowdworkers to do translation tasks.

3 Collecting Parallel URLs with Crowdworkers

Fig. 1 shows an overview of our collecting protocol. Our method asks workers to find URLs that are related to the target domain and written in parallel. We then extract the parallel sentences from these URLs and fine-tune the general-purpose machine translation model with the collected data.

This section is organized as follows: In Section 3.1, we explain why we focus on collecting parallel URLs and describe their advantages. We overview the details of our crowdsourcing task definition in Section 3.2. In Section 3.3, we describe how we extract parallel sentences from the reported URLs. We describe the details of our reward setting in Section 3.4.

3.1 Advantages

Previous works, which adapted a machine translation model to a specific domain, created resources by asking crowdworkers to translate text (Lewis et al., 2011; Anastasopoulos et al., 2020; Zaidan and Callison-Burch, 2011; Behnke et al., 2018; Kalimuthu et al., 2019). In contrast, our method asks workers to find web URLs (instead of translating sentences) that have parallel sentences in the target domain.

This method has two advantages. The first concerns task difficulty. To achieve rapid domain adaptation, the task must be easy enough that many crowdworkers can participate. Thus, we do not assume that the workers fluently understand both the source and target languages. Finding potential web URLs that have parallel sentences is relatively easy and can be done by any crowdworker.

The other advantage involves task efficiency. We asked workers to collect the URLs of parallel web pages instead of parallel sentences because recent previous works successfully extracted parallel sentences from parallel URLs (Bañón et al., 2020). Efficiency is important for our method, since we focus on speed to create a domain-specific model.

3.2 Crowdsourcing Task Definition

We focus on collecting the parallel sentences of languages e and f. We created a web application to accept reports from the crowdworkers and extracted parallel sentences from the reported web URLs. We prepared a development set (a small portion of the parallel sentences) of the target domain and distribute it to the workers as examples of the type of sentences we want them to collect. The crowdworkers are asked to find pairs of web URLs that contain parallel sentences of the target domain. We call this URL pair a parallel URL. Note that we collect the URLs of pages written in parallel; this means that workers act as parallel document aligners. We do not accept parallel URLs that have already been reported by others.

3.3 Parallel Sentence Extraction

After obtaining parallel URLs from workers, we extract parallel sentences from the reported URLs. First, we downloaded the reported web URLs and extracted the texts[1] and removed the sentences that are not in the e or f language based on `CLD2`[2]. Then we used `vecalign` (Thompson and Koehn, 2019) to extract the parallel sentences, a step that aligns them based on the multi-lingual sentence embeddings LASER (Artetxe and Schwenk, 2019). We discard noisy sentence pairs based on sentence alignment scores[3] and do not use them for model training.

3.4 Reward Settings

To bolster the crowdworkers' motivation, reward setting is one of the most important issues (Posch et al., 2019). In this paper, we tested two types of rewards: fixed or variable. In the following, we describe both reward settings.

3.4.1 Fixed Reward

Fixed reward pays a set amount for each reported URL if we can extract at least one parallel sentence from it. This fixed reward setting is one very typical setting for crowdsourcing.

[1] Since we expect the workers to act as document aligners, we focus on the reported URLs and do not crawl the links in the reported URLs.

[2] https://github.com/CLD2Owners/cld2

[3] Since `vecalign` outputs a scoring cost where a lower score means better alignment, our implementation removes a sentence pair if its cost exceeds 0.7.

3.4.2 Variable Reward

The key motivation of crowdworkers is probably to earn money (Antin and Shaw, 2012), and thus they try to maximize their earnings (Horton and Chilton, 2010). Since the fixed reward setting only considers the number of reported URLs, workers may report noisy URLs whose texts are not parallel or not in the target domain in an effort to maximize their number of reports.

To alleviate this concern, we tested another reward setting: varying rewards based on the quality of their reported parallel URLs. We hypothesize that the workers will improve their work performance when we pay more for good work and less for poor work.

We defined parallel URLs as those satisfying the following criteria that help improve the translation performance in the target domain: (1) they contain a large number of parallel sentences, (2) the parallel sentences are correctly translated, and (3) the parallel sentences are in the target domain. To reflect these criteria in the reward, we set variable reward r:

$$r = \min(r_{\max}, r_{\min} + \sum_{(x_i,y_i)\in\mathbb{D}} S_a(x_i, y_i) + S_d(x_i)), \quad (1)$$

where $\mathbb{D}$ is a set of parallel sentences extracted from the reported URLs, x_i and y_i are parallel sentences of languages e and f, $r_{\min}$ and $r_{\max}$ are the minimum and maximum reward per report, and $S_a(\cdot)$ and $S_d(\cdot)$ are the sentence alignment and domain similarity scores, which are explained below.

Sentence Alignment Score Suppose n parallel sentences $\mathbb{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ extracted from the reported URLs. Sentence alignment score S_a is calculated as follows:

$$S_a = \sum_{(x_i,y_i)\in\mathbb{D}} \varsigma(-\mathrm{V}(x_i, y_i)), \quad (2)$$

where $\mathrm{V}(\cdot)$ is an alignment cost function of `vecalign`, where lower is better, and $\varsigma(\cdot)$ is a sigmoid function that converts the score into the range 0 to 1.

Domain Similarity Score The domain similarity score is based on cross-entropy (Moore and Lewis, 2010):

$$S_d = \sum_{x_i\in\mathbb{D}} \varsigma(H_I(x_i) - H_N(x_i)), \quad (3)$$

where I and N are in-domain and non-domain-specific language models and $H(x_i)$ is the per-word cross-entropy of sentence x_i.

Through our web application, workers can check the results (of their previous reports), which include the reward amounts, the scores, and the number of extracted parallel sentences. These results are available a few minutes after we accept their reports so that they can improve their work and maximize their scores and their payments.

4 Experiments

We carried out experiments to confirm whether different reward settings influenced the workers' performance and translation accuracy. Prior to them, we conducted a preliminary experiment to check the effect of our method in smaller settings. Refer to Section B in the Appendix for this preliminary experiment. In this section, we empirically confirm the effectiveness of our method by focusing on five domains.

4.1 Experimental Settings

In this experiment, we tested English-Japanese translations on five domains: COVID-19, news, science, patents, and legal matters[4]. The details of the domains and the corpus statistics of the development/test sets are shown in Section A.2 in the Appendix. We hired 97 crowdworkers through a crowdsourcing platform called Crowdworks[5]. Each worker was randomly assigned to a single target domain.

We used both the fixed and variable reward setting for the science and patent domains, and only the variable reward setting for the other three domains, since we confirmed that the variable reward setting is effective in the following experiment (see Section 4.2.1). We set the fixed reward at 25 JPY ($\simeq$ 0.23 USD), $r_{\min}$ to 10 JPY ($\simeq$ 0.09 USD), and $r_{\max}$ to 100 yen ($\simeq$ 0.91 USD) for the variable reward[6]. Since our task is much easier than translating sentences, we pay our workers much less than such translators of sentences[7]. Data collection continued for 13 days. We trained

[4]We chose these domains because they require special domain knowledge and are difficult to translate by current models.

[5]https://crowdworks.jp/

[6]We paid the workers in JPY since they mainly live in Japan. They are guaranteed at least the minimum wage.

[7]Typically, it requires around 0.15 USD to translate an English word into Japanese.

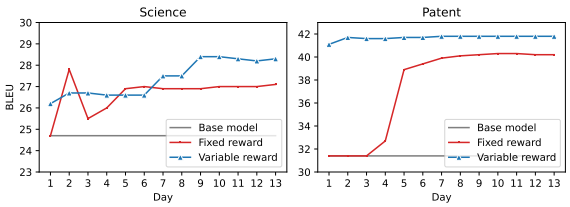

Figure 2: Transition of test set BLEU scores on science and patent domains

in-domain language models with each development set to calculate the domain similarity scores. We used `KenLM` as an implementation of the n-gram language model (Heafield, 2011) to calculate the domain similarity scores. We trained the in-domain language model with the development set of TICO-19 and the non-domain-specific model with JParaCrawl v2.0 (Morishita et al., 2020).

Translation Model Settings As a neural machine translation model, we employed the Transformer model with its base settings (Vaswani et al., 2017). To train the general-purpose baseline model, we used JParaCrawl v2.0 (Morishita et al., 2020), which contains 10 million English-Japanese parallel sentences and tokenized the training data into subwords with the `sentencepiece` (Kudo and Richardson, 2018) toolkit. We set the vocabulary size to 32,000 for each language side and removed sentences that exceeded 250 subwords to reduce the noisy sentence pairs.

We trained the baseline model with JParaCrawl until it converged and then fine-tuned it with the newly collected in-domain parallel sentences. See Section A.1 in the Appendix for the detailed hyperparameter settings.

We used `SacreBLEU` (Post, 2018) to evaluate the translation performance and report the BLEU scores[8] (Papineni et al., 2002).

4.2 Experimental Results

4.2.1 Fixed or Variable Reward Comparison

First, we address whether the variable reward setting encouraged the workers to find more valuable data. Fig. 2 compares the BLEU scores between the fixed and variable reward settings in the science and patent domains. The variable reward setting achieved higher BLEU scores than the fixed reward setting in both domains. Combined with the preliminary experiments described in Section B in the Appendix, we conclude that the variable reward setting collects beneficial data. Thus, the following sections mainly discuss the results of the variable reward setting.

4.2.2 Data Collection

Table 1 shows the experimental results on the variable reward setting, including the number of URLs and collected parallel sentences. Our framework collected a large number of parallel sentences for all five domains. The lower half of Fig. 3 shows the transitions of the number of sentences collected with crowdsourcing on the COVID-19, news, and legal domains. For the other domains, see Fig. 6 in the Appendix. The number of collected sentences linearly increased as we continued crowdsourcing.

We carried out the task on the five domains and assigned roughly the same number of workers to each task, but we found that the number of reports differed. This implies that the task's difficulty might differ depending on the target domain. For example, the science task might be easier than the others because several scientific journals translate abstracts (and make them available on the web) into other languages.

4.2.3 Translation Performance

Table 1 shows the BLEU scores of the baseline and the fine-tuned models with the collected in-domain parallel sentences. The fine-tuned models achieved significantly better accuracy with an average of +7.8 points than the baseline model on all five domains. In particular, our legal domain model improved by +19.7 points. One likely reason is that the legal domain frequently uses words that do not appear in other domains, and the collected in-domain data improved these translations.

The top of Fig. 3 shows the transitions of the BLEU scores as we continued the data collection for the COVID-19, news, and legal domains (see the Base Model and w/Crawled lines). Fig. 6 in the Appendix shows the results of the other domains. All the domains show identical tendencies. Their performance surpassed the baseline on the first or second day of crowdsourcing and continued growing as we collected more data. This supports our assumption that our method can achieve rapid domain adaptation for many domains.

[8] We used NFKC to normalize both the Japanese translations and references since JParaCrawl is normalized by the same procedure.

Domain	#URLs	#Sentences	Cost (USD)	Development BLEU			Test BLEU		
				Base model	w/Crawled		Base model	w/Crawled	
COVID-19	6,841	165,838	1,807.7	25.9	**28.7**	(+2.8)	31.7	**34.3**	(+2.6)
News	10,712	220,559	2,765.5	19.3	**21.2**	(+1.9)	20.5	**23.1**	(+2.6)
Science	10,948	390,303	3,217.8	25.0	**27.9**	(+2.9)	24.7	**28.3**	(+3.6)
Patent	4,135	307,104	1,431.3	27.4	**36.6**	(+9.2)	31.4	**41.8**	(+10.4)
Legal	5,438	302,747	2,088.0	22.9	**42.0**	(+19.1)	22.8	**42.5**	(+19.7)

Table 1: Experimental results for five domains. Model fine-tuned with newly crawled data significantly improved BLEU scores on all of them.

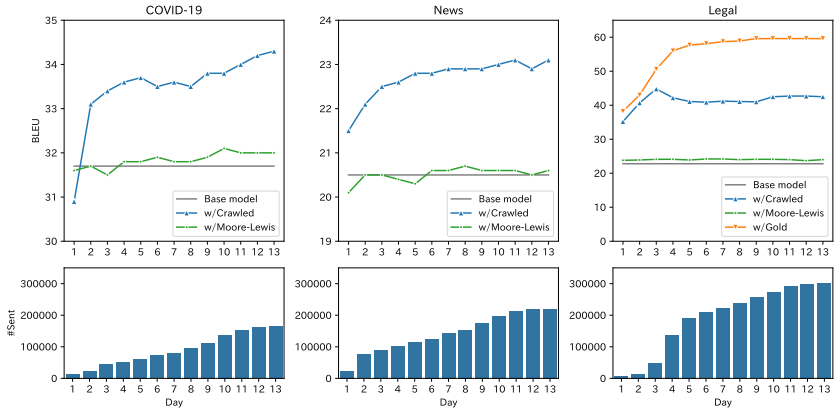

Figure 3: Transition of BLEU scores (top) and sentences collected (bottom) as we continued data collection for the COVID-19, news, and legal domains. Model named w/Moore-Lewis is fine-tuned with domain-relevant sentences extracted from existing general-purpose corpus, as described in Section 4.3. As an upper bound of fine-tuning, we show the scores of the w/Gold model, which was fine-tuned with existing target-domain parallel corpus, as described in Section D.2 in the Appendix.

Domain	Test BLEU			
	Base	w/Crawled		w/ML
COVID-19	31.7	34.3	(+2.6)	32.0 (+0.3)
News	20.5	23.1	(+2.6)	20.6 (+0.1)
Science	24.7	28.3	(+3.6)	25.3 (+0.6)
Patent	31.4	41.8	(+10.4)	32.0 (+0.6)
Legal	22.8	42.5	(+19.7)	24.0 (+1.2)

Table 2: BLEU score comparisons with Moore-Lewis (w/ML)

4.3 Comparison: Selecting In-domain Data from Existing Parallel Corpus

In this section, we compare our method with the existing domain adaption method to answer the following question: Do we really need to collect new data with crowdworkers?

Currently, the most common domain-adaptation method is to find target domain sentences from existing parallel corpora (Chu and Wang, 2018). As with the existing method, we used the one proposed by Moore and Lewis (2010)[9]. We scored all the sentences in JParaCrawl and used those considered most relevant to the target domain. We selected the same number of sentences as in our collected data.

Table 2 shows the BLEU scores of each model, and the top of Fig. 3 shows the transition of the BLEU scores (see w/Moore-Lewis). The Moore-Lewis method surpassed the baseline on all five domains, but by a narrow margin. Although their method does not require additional cost, our method achieved significantly better performance with just a small additional cost. Thus the answer to the above question is yes: our method outperformed the existing domain-adaptation method.

5 Conclusion

We introduced a new framework for domain adaptation in machine translation. Our method asks crowdworkers to find parallel URLs related to the target domain. Such a task does not require any professional skills and can be done cheaply by many people. We then fine-tuned the machine translation model with parallel sentences in the target domain extracted from the reported URLs. Through experiments, we empirically confirmed that our

[9]Some may be concerned that this method is outdated, but it is still considered a strong domain-adaptation method, since the recent first-ranked system among WMT submissions uses it for selecting relevant data (Junczys-Dowmunt, 2018).

framework significantly improved the translation performance for a target domain within a few days of crowdsourcing and at a reasonable cost. We also confirmed that our variable reward function, which is based on the quality of parallel sentences, changed the behavior of the workers who began to collect more effective parallel sentences, increasing the translation accuracy.

Limitations

We assume that websites containing in-domain parallel sentences are available on the web, which might not be true for some difficult domains. However, since we believe that parallel sentences in neighboring domains are available on the web, we expect our method to improve the translation accuracy on these domains.

We conducted English-Japanese experiments. We expect our method to work on most major language pairs, including German-English and Chinese-Japanese, since there are many parallel websites on these language pairs. However, we haven't yet confirmed whether it does works on very minor language pairs, because finding parallel websites for them is difficult.

Ethics Statement

In the experiments, our crawler strictly followed the "robots.txt" and crawled only from allowed websites. During the experiments, we also ensured that the crowdworkers earned at least the minimum wage.

Acknowledgements

We thank the three anonymous reviewers for their insightful comments.

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*.

Judd Antin and Aaron Shaw. 2012. Social desirability bias and self-reports of motivation: A study of amazon mechanical turk in the US and India. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2925–2934.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics (TACL)*, 7:597–610.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–362.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4555–4567.

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 5th Conference on Machine Translation (WMT)*, pages 1–55.

Maximiliana Behnke, Antonio Valerio Miceli Barone, Rico Sennrich, Vilelmini Sosoni, Thanasis Naskos, Eirini Takoulidou, Maria Stasimioti, Menno van Zaanen, Sheila Castilho, Federico Gaspari, Panayota Georgakopoulou, Valia Kordoni, Markus Egg, and Katia Lida Kermanidis. 2018. Improving machine translation of educational content via crowdsourcing. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*.

Mara Chinea-Ríos, Álvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. In *Proceedings of the 2nd Conference on Machine Translation (WMT)*, pages 138–147.

Katsuki Chousa, Masaaki Nagata, and Masaaki Nishino. 2020. SpanAlign: Sentence alignment method based on cross-language span prediction and ILP. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 4750–4761.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1304–1319.

Ahmed El-Kishky and Francisco Guzmán. 2020. Massively multilingual document alignment with cross-lingual sentence-mover's distance. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 616–625.

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 the 10th NTCIR Conference on Evaluation of Information Access Technologies*, pages 260–286.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*, pages 187–197.

John Joseph Horton and Lydia B. Chilton. 2010. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 209–218.

Youxuan Jiang, Catherine Finegan-Dollak, Jonathan K. Kummerfeld, and Walter Lasecki. 2018. Effective crowdsourcing for a new type of summarization task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 628–633.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, pages 425–430.

Marimuthu Kalimuthu, Michael Barz, and Daniel Sonntag. 2019. Incremental domain adaptation for neural machine translation in low-resource settings. In *Proceedings of the 4th Arabic Natural Language Processing Workshop*, pages 1–10.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-AIP-NTT at WMT 2020 news translation task. In *Proceedings of the 5th Conference on Machine Translation (WMT)*, pages 145–155.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 66–71.

William Lewis, Robert Munro, and Stephan Vogel. 2011. Crisis MT: Developing a cookbook for MT in crisis situations. In *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*, pages 501–511.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 220–224.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pages 3603–3609.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 48–53.

Vassilis Papavassiliou, Prokopis Prokopidis, and Stelios Piperidis. 2018. Discovering parallel language resources for training MT engines. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Lisa Posch, Arnim Bleier, Clemens M. Lechner, Daniel Danner, Fabian Flöck, and Markus Strohmaier. 2019. Measuring motivations of crowdworkers: The multidimensional crowdworker motivation scale. *ACM Transactions on Social Computing*, 2(2).

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, pages 186–191.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics (CL)*, 29(3):349–380.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation.

In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7008–7013.

Serge Sharoff, Pierre Zweigenbaum, and Reinhard Rapp. 2015. BUCC shared task: Cross-language document similarity. In *Proceedings of the Eighth Workshop on Building and Using Comparable Corpora*, pages 74–78.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 2818–2826.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1348.

Brian Thompson and Philipp Koehn. 2020. Exploiting sentence order in document alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1400–1410.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 6000–6010.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 560–566.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1220–1229.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 60–67.

Base model	
Architecture	Transformer (base)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$) (Kingma and Ba, 2015)
Learning rate schedule	Inverse square root decay
Warmup steps	4,000
Max learning rate	0.001
Dropout	0.3 (Srivastava et al., 2014)
Gradient clipping	1.0
Label smoothing	$\epsilon_{ls} = 0.1$ (Szegedy et al., 2016)
Mini-batch size	320,000 tokens
Updates	24,000 updates
Averaging	Save checkpoint every 200 steps and average the last eight
Implementation	`fairseq` (Ott et al., 2019)
Parameters	93.2 million
Fine-tuning	
Learning rate	1×10^{-5} (Fixed)
Mini-batch size	32,000 tokens
Updates	8 epochs[10]
Averaging	Save checkpoint every epoch and average the last eight

Table 3: List of hyperparameters

	Development		Test	
Domain	**#Sentences**	**#Tokens**	**#Sentences**	**#Tokens**
COVID-19	971	21,085	2,100	49,490
News	1,998	45,318	1,000	22,141
Science	1,790	39,377	1,812	39,573
Patent	2,000	60,312	2,300	71,847
Legal	1,313	46,922	1,310	46,842

Table 4: Number of sentences and English tokens in development and test sets

A Detailed Experimental Settings

A.1 Hyperparameters

Table 3 shows the hyperparameter settings used to train a general-purpose machine translation model and fine-tune it with target domain sentences. We did not conduct a hyperparameter search, and almost all the settings were borrowed from previous works (Morishita et al., 2020; Kiyono et al., 2020).

A.2 Datasets

We used TICO-19 (Anastasopoulos et al., 2020) as development and test sets to evaluate the translation performance of the COVID-19 domain. Since the original TICO-19 does not include Japanese translations, professional translators translated the English sentences to create a Japanese reference. We used the development/test sets from the WMT20 news shared task (Barrault et al., 2020) for the news domain and the NTCIR-10 patent translation task for the patent domain. For the science domain, we used ASPEC (Nakazawa et al., 2016), which contains

[10]One epoch means the model sees the entire corpus once. Thus the number of updates depends on the data size. We chose this setting because a fixed number of updates has a risk of over-fitting if the fine-tuning data are too small.

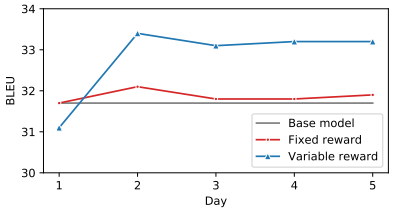

Figure 4: Relationship between BLEU scores and crowdsourcing days for small-scale experiment

excerpts of scientific papers. For the legal domain, we used the Japanese-English legal parallel corpus[11]. Since it is not divided into development and test sets, we created them by randomly choosing sentences from the entire corpus. The details of the development and test set corpus statistics are shown in Table 4.

B Preliminary Experiments

We carried out a preliminary experiment to determine how the different reward settings influenced the workers' performance and translation accuracy.

B.1 Experimental Settings

Target Domain and Crowdsourcing Settings In this experiment, we focused on English-Japanese translations in the COVID-19 domain. We assigned ten crowdworkers to each reward setting and asked them to find websites that contained parallel sentences related to the COVID-19 domain. The crowdsourcing continued for five days.

We set the fixed reward at 70 JPY ($\simeq$ 0.64 USD) per report. For the variable reward setting, we paid r JPY for each report, as shown by Eq. 1. We set $r_{\min}$ to 20 JPY ($\simeq$ 0.18 USD) and $r_{\max}$ to 100 JPY($\simeq$ 0.91 USD). Other model training settings, including the hyperparameters, are identical as in Section 4.1.

B.2 Experimental Results

B.2.1 Data Collection

Table 5 shows the results of crowdsourcing, including the number of reports, extracted parallel sentences, and the payments to the workers. We received almost the same number of reports in both reward settings. However, there was a significant difference in the average number of sentences per report: 10.4 for the fixed rewards and 13.4 for the variable rewards. One likely reason is that the

[11]http://www.phontron.com/jaen-law/index.html

Reward	#URLs	#Sentences	Cost (USD)	Development BLEU		Test BLEU	
				Base model	w/Crawled	Base model	w/Crawled
Fixed	504	5,220	322.8	25.9	26.3 (+0.4)	31.7	31.9 (+0.2)
Variable	503	6,722	284.3		**27.1** (+1.2)		**33.2** (+1.5)

Table 5: Small-scale experiment's results (five days of crowdsourcing), including crowdsourcing results and BLEU scores of baseline and model fine-tuned with newly collected in-domain corpus.

workers tried to maximize their rewards. We believe the number of in-domain parallel sentences is one crucial key for improving accuracy, and we reflected this idea in our reward function. Thus it improved the workers' performance more than the fixed reward setting. With the variable reward setting, we also reduced the cost and obtained even more parallel sentences by reducing the payments to low-quality workers and increasing them to good workers.

B.2.2 Translation Performance

Table 5 shows the BLEU scores of the baseline model and the fine-tuned model with our crawled in-domain parallel data. The model fine-tuned with variable reward data achieved better results than using fixed rewards. We believe the quality of the collected data caused the difference in addition to the number of parallel sentences, as previously mentioned. We compared the domain similarity scores described in Section 3.4.2 to check whether the collected data are related to the target domain and found that the data collected with the variable reward setting achieved higher scores than with the fixed rewards. This implies that the variable reward setting motivated the workers to find parallel web URLs related to the target domain, increasing the accuracy of the fine-tuned model.

Fig. 4 shows how the BLEU scores changed as crowdsourcing continued, and Fig. 5 in the Appendix shows the number of sentences used for this experiment. The fine-tuned model with the variable reward data outperformed the baseline model, even by the second day of crowdsourcing. This result supports our claim that our method helps provide a domain-adapted model in a few days, which is critical in such urgent situations as COVID-19.

From this experiment, we found that a variable reward setting encouraged workers to find more valuable parallel URLs, improved their translation performance in the target domain over a few days, and reduced the cost more than the fixed reward setting.

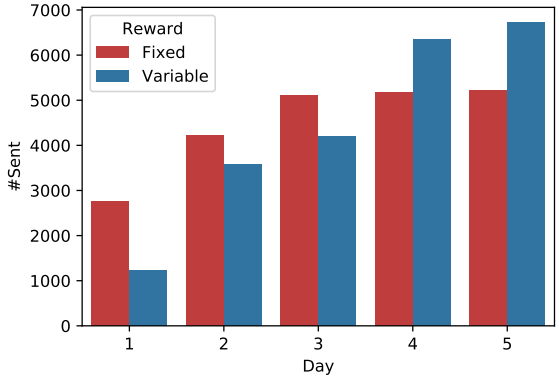

Figure 5: Collected sentences used for fine-tuning in experiment of Fig. 4. See Section B for details.

	Alignment	Domain	Both
Top 20%	34.9	34.2	34.4
Middle 20%	32.9	33.3	33.1
Bottom 20%	30.5	32.3	31.8

Table 6: BLEU scores of model fine-tuned with top/middle/bottom 20% scored sentences on COVID-19 domain test set

C Additional Experimental Results

Fig. 5 shows the numbers of sentences used for fine-tuning in the preliminary experiment (Section B). Fig. 6 shows the transitions of the BLEU scores in the experiment described in Section 4 and the number of sentences collected in the variable reward setting.

D Additional Analysis

D.1 Analysis: Reward Function

We varied the rewards to the workers with the reward function based on the sentence alignment and domain similarity scores. We pondered whether this reward function could correctly measure the data quality. To confirm this, we ordered the collected data with respect to the sentence alignment scores (Eq. 2), the domain similarity scores (Eq. 3), or the sum of both scores. Then we fine-tuned the model with the top/middle/bottom 20% of the sorted data.

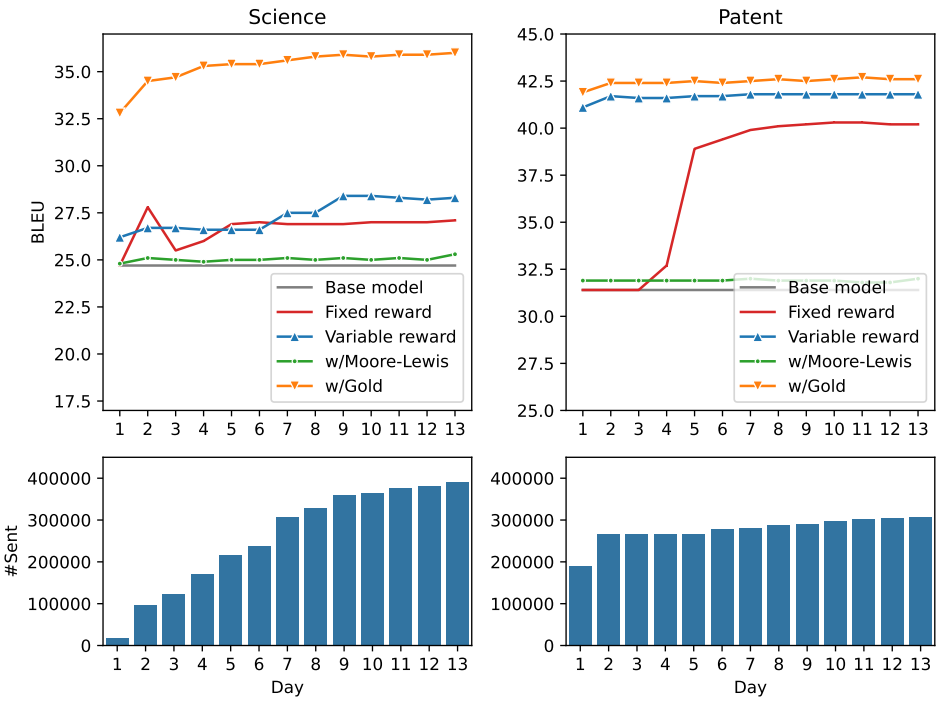

Figure 6: Transitions of BLEU scores (upper-side) and number of collected sentences (lower-side) as we continued data collection for science and patent domains. Detailed explanations can be found in Section 4.

	Test BLEU				
Domain	**Base**	**w/Crawled**		**w/Gold**	
Science	24.7	28.3	(+3.6)	36.0	(+11.3)
Patent	31.4	41.8	(+10.4)	42.6	(+11.2)
Legal	22.8	42.5	(+19.7)	59.6	(+36.8)

Table 7: BLEU score comparison with Gold data (w/Gold)

Table 6 shows the BLEU scores of the fine-tuned models on the COVID-19 domain. There is a clear trend that the model fine-tuned with high-scored data achieved higher accuracy, and there is a large gap between the top and the bottom for all the score functions. From this result, we conclude that our reward function correctly measured the quality of the data, and we paid more for high-quality works and less for low-quality works.

D.2 Comparison: Gold In-domain Parallel Corpus

In this section, we compare our collected data with the existing domain-specific parallel corpus. Among the five domains from which we collected sentences, there is a domain-specific parallel corpus for the science, patent, and legal domains. Note that the availability of domain-specific data is quite limited since creating such parallel data requires professionals, thus incurring heavy costs. Accordingly, this experiment resembles a comparison between our method and the upper bound. In the following, we call this domain-specific parallel corpus the gold data.

As gold data, we used ASPEC (Nakazawa et al., 2016) for the science domain, NTCIR (Goto et al., 2013) for the patent domain, and the Japanese-English legal parallel corpus for the legal domain. For a fair comparison, we randomly selected the same number of sentences as our collected data.

Table 7 shows the BLEU scores of the model fine-tuned with the gold data. Unsurprisingly, the w/Gold models achieved better accuracy than the w/Crawled models. However, the results of some of the latter were close to those of the former, such as the patent domain.

From Fig. 3, we compared the transition of the BLEU scores (see w/Crawled and w/Gold in the legal domain). From the first to third days, our method's performance resembled that of the w/Gold model. Since room remains for improvements after the fourth day, future work will refine the crowdsourcing protocol.

E Links to Data and Software

E.1 Data

JParaCrawl https://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/

TICO-19 https://tico-19.github.io/

WMT20 news shared task http://www.statmt.org/wmt20/translation-task.html

ASPEC http://orchid.kuee.kyoto-u.ac.jp/ASPEC/

NTCIR-10 http://research.nii.ac.jp/ntcir/permission/ntcir-10/perm-en-PatentMT.html

Legal parallel corpus http://www.phontron.com/jaen-law/index.html

E.2 Software

vecalign https://github.com/thompsonb/vecalign

CLD2 https://github.com/CLD2Owners/cld2

KenLM https://github.com/kpu/kenlm

fairseq https://github.com/pytorch/fairseq

SacreBLEU https://github.com/mjpost/sacreBLEU

sentencepiece https://github.com/google/sentencepiece