A High-Quality Multilingual Dataset for Structured Documentation Translation

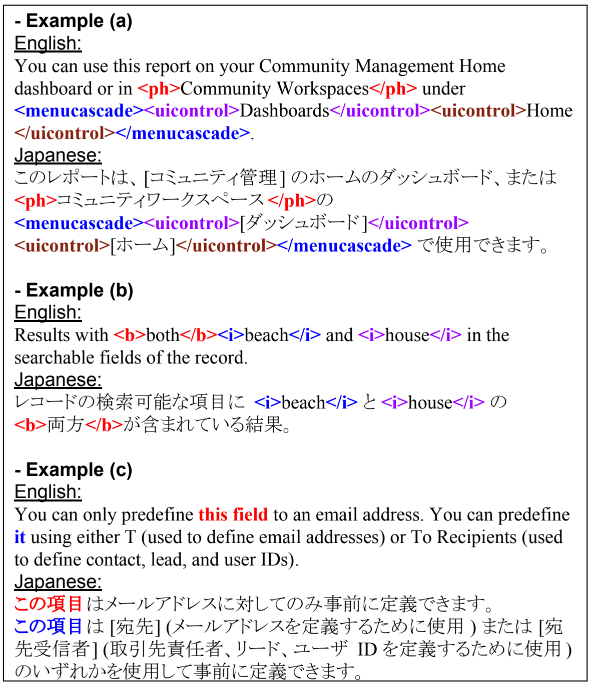
Kazuma Hashimoto **Raffaella Buschiazzo** **James Bradbury**[*]
Teresa Marshall **Richard Socher** **Caiming Xiong**
Salesforce
{k.hashimoto,rbuschiazzo,james.bradbury,
teresa.marshall,rsocher,cxiong}@salesforce.com

Abstract

This paper presents a high-quality multilingual dataset for the documentation domain to advance research on localization of structured text. Unlike widely-used datasets for translation of plain text, we collect XML-structured parallel text segments from the online documentation for an enterprise software platform. These Web pages have been professionally translated from English into 16 languages and maintained by domain experts, and around 100,000 text segments are available for each language pair. We build and evaluate translation models for seven target languages from English, with several different copy mechanisms and an XML-constrained beam search. We also experiment with a non-English pair to show that our dataset has the potential to explicitly enable 17 × 16 translation settings. Our experiments show that learning to translate with the XML tags improves translation accuracy, and the beam search accurately generates XML structures. We also discuss trade-offs of using the copy mechanisms by focusing on translation of numerical words and named entities. We further provide a detailed human analysis of gaps between the model output and human translations for real-world applications, including suitability for post-editing.

1 Introduction

Machine translation is a fundamental research area in the field of natural language processing (NLP). To build a machine learning-based translation system, we usually need a large amount of bilingually-aligned text segments. Examples of widely-used datasets are those included in WMT (Bojar et al., 2018) and LDC[1], while new evaluation datasets are being actively created (Michel and Neubig, 2018; Bawden et al., 2018; Müller et al., 2018). These existing datasets have mainly focused on translating plain text.



- Example (a)
English:
You can use this report on your Community Management Home dashboard or in <ph>Community Workspaces</ph> under <menucascade><uicontrol>Dashboards</uicontrol><uicontrol>Home </uicontrol></menucascade>.
Japanese:
このレポートは、[コミュニティ管理] のホームのダッシュボード、または <ph>コミュニティワークスペース </ph>の <menucascade><uicontrol>[ダッシュボード]</uicontrol> <uicontrol>[ホーム]</uicontrol></menucascade> で使用できます。

- Example (b)
English:
Results with <b>both</b><i>beach</i> and <i>house</i> in the searchable fields of the record.
Japanese:
レコードの検索可能な項目に <i>beach</i> と <i>house</i> の <b>両方</b>が含まれている結果。

- Example (c)
English:
You can only predefine **this field** to an email address. You can predefine **it** using either T (used to define email addresses) or To Recipients (used to define contact, lead, and user IDs).
Japanese:
この項目はメールアドレスに対してのみ事前に定義できます。
この項目は [宛先] (メールアドレスを定義するために使用) または [宛先受信者] (取引先責任者、リード、ユーザ ID を定義するために使用) のいずれかを使用して事前に定義できます。

Figure 1: English-Japanese examples in our dataset.

On the other hand, text data, especially on the Web, is not always stored as plain text, but often wrapped with markup languages to incorporate document structure and metadata such as formatting information. Many companies and software platforms provide online help as Web documents, often translated into different languages to deliver useful information to people in different countries. Translating such Web-structured text is a major component of the process by which companies localize their software or services for new markets, and human professionals typically perform the translation with the help of a *translation memory* (Silvestre Baquero and Mitkov, 2017) to increase efficiency and maintain consistent termi-

[*] Now at Google Brain.
[1] https://www.ldc.upenn.edu/

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 116–127
Florence, Italy, August 1-2, 2019. ©2019 Association for Computational Linguistics

nology. Explicitly handling such structured text can help bring the benefits of state-of-the-art machine translation models to additional real-world applications. For example, structure-sensitive machine translation models may help human translators accelerate the localization process.

To encourage and advance research on translation of structured text, we collect parallel text segments from the public online documentation of a major enterprise software platform, while preserving the original XML structures.

In experiments, we provide baseline results for seven translation pairs from English, and one non-English pair. We use standard neural machine translation (NMT) models, and additionally propose an XML-constrained beam search and several discrete copy mechanisms to provide solid baselines for our new dataset. The constrained beam search contributes to accurately generating source-conditioned XML structures. Besides the widely-used BLEU (Papineni et al., 2002) scores, we also investigate more focused evaluation metrics to measure the effectiveness of our proposed methods. In particular, we discuss trade-offs of using the copy mechanisms by focusing on translation of named entities and numerical words. We further report detailed human evaluation and analysis to understand what is already achieved and what needs to be improved for the purpose of helping the human translators (a post-editing context). As our dataset represents a single, well-defined domain, it can also serve as a corpus for domain adaptation research (either as a source or target domain). We will release our dataset publicly, and discuss potential for future expansion in Section 6.

2 Collecting Data from Online Help

This section describes how we constructed our new dataset for XML-structured text translation.

Why high quality? We start from the publicly-available online help of a major international enterprise software-as-a-service (SaaS) platform. The software is provided in many different languages, and its multilingual online documentation has been localized and maintained for 15 years by the same localization service provider and in-house localization program managers. Since the beginning they have been storing translations in a translation memory (i.e. computer-assisted translation tool) to increase quality and terminology consistency. The documentation makes frequent use of structured formatting (using XML) to convey information to readers, so the translators have aimed to ensure consistency of formatting and markup structure, not just text content, between languages.

How many languages? The web documentation currently covers 16 non-English languages translated from English. These 16 languages are Brazilian Portuguese, Danish, Dutch, Finnish, French, German, Italian, Japanese, Korean, Mexican Spanish, Norwegian, Russian, Simplified Chinese, Spanish, Swedish, and Traditional Chinese. In practice, the human translation has been done from English to the other languages, but all the languages could be potentially considered as both source and target because they contain the same tagging structure.

2.1 Bilingual Web Page Alignments

In this paper, we focus on each language pair separately, as an initial construction of our dataset. Each page of the online documentation in the different languages is already aligned in the following two ways:

– first, the same page has the same file name between languages; for example, if we have a page about "WMT", there would be `/English/wmt.xml` and `/Japanese/wmt.xml`, and

– second, most of the high-level XML elements are already aligned, because the original English files have been translated by preserving the same XML structures as much as possible in the localization process, to show the same content with the same formatting. Figure 2 shows a typical pair of files and the alignment of their high-level XML elements.

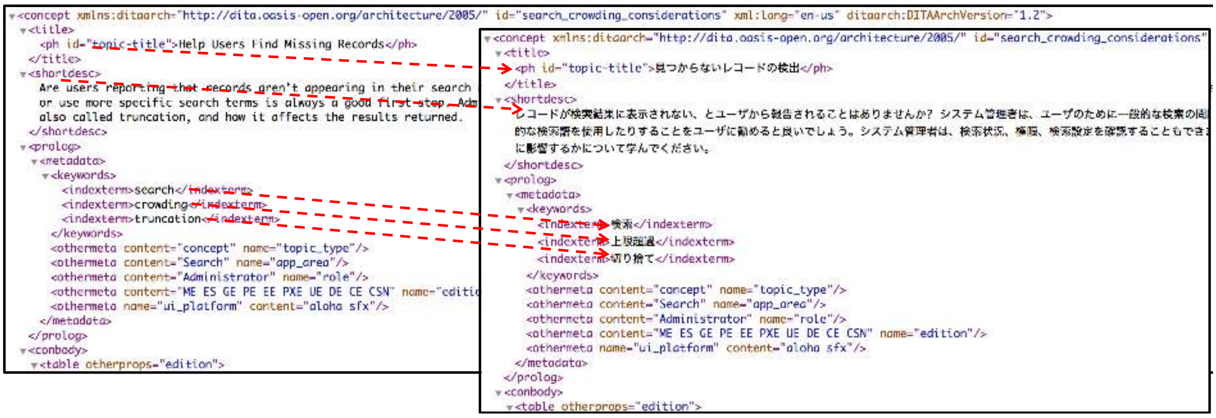
Our dataset contains about 7,000 pairs of XML files for each language pair; for example, there are 7,336 aligned files for English-{French, German, Japanese}, 7,160 for English-{Finnish, Russian}, and 7,927 for Finnish-Japanese.[2]

2.2 Extracting Parallel Text Segments

XML parsing and alignment For each language pair, we extract parallel text segments from XML structures. We use the `etree` module in a Python library called `lxml`[3] to process XML

[2] Some documents are not present, or not aligned, in all languages.

[3] https://lxml.de/



```
▾<concept xmlns:ditaarch="http://dita.oasis-open.org/architecture/2005/" id="search_crowding_considerations" xml:lang="en-us" ditaarch:DITAArchVersion="1.2">
  ▾<title>
     <ph id="topic-title">Help Users Find Missing Records</ph>
   </title>
  ▾<shortdesc>
     Are users reporting that records aren't appearing in their search
     or use more specific search terms is always a good first step. Adm
     also called truncation, and how it affects the results returned.
   </shortdesc>
  ▾<prolog>
    ▾<metadata>
      ▾<keywords>
         <indexterm>search</indexterm>
         <indexterm>crowding</indexterm>
         <indexterm>truncation</indexterm>
       </keywords>
       <othermeta content="concept" name="topic_type"/>
       <othermeta content="Search" name="app_area"/>
       <othermeta content="Administrator" name="role"/>
       <othermeta content="ME ES GE PE EE PXE UE DE CE CSN" name="editio
       <othermeta name="ui_platform" content="aloha sfx"/>
     </metadata>
   </prolog>
  ▾<conbody>
    ▾<table otherprops="edition">
```

```
▾<concept xmlns:ditaarch="http://dita.oasis-open.org/architecture/2005/" id="search_crowding_considerations"
  ▾<title>
     <ph id="topic-title">見つからないレコードの検出</ph>
   </title>
  ▾<shortdesc>
     レコードが検索結果に表示されない、とユーザから報告されることはありませんか? システム管理者は、ユーザのために一般的な検索の回
     的な検索語を使用したりすることをユーザに勧めると良いでしょう。システム管理者は、検索状況、権限、検索設定を確認することもでき
     に影響するかについて学んでください。
   </shortdesc>
  ▾<prolog>
    ▾<metadata>
      ▾<keywords>
         <indexterm>検索</indexterm>
         <indexterm>上限超過</indexterm>
         <indexterm>切り捨て</indexterm>
       </keywords>
       <othermeta content="concept" name="topic_type"/>
       <othermeta content="Search" name="app_area"/>
       <othermeta content="Administrator" name="role"/>
       <othermeta content="ME ES GE PE EE PXE UE DE CE CSN" name="edition"/>
       <othermeta name="ui_platform" content="aloha sfx"/>
     </metadata>
   </prolog>
  ▾<conbody>
    ▾<table otherprops="edition">
```

Figure 2: An aligned pair of English and Japanese XML files.

strings in the XML files. Since the XML elements are well formed and translators keep the same tagging structure as much as their languages allow it, as described in Section 2.1, we first linearize an XML-parsed file into a sequence of XML elements. We then use a pairwise sequence alignment algorithm for each bilingually-aligned file, based on XML tag matching. As a result, we have a set of aligned XML elements for the language pair.

Tag categorization Next, we manually define which XML elements should be translated, based on the following three categories:
– Translatable:
A translatable tag (e.g. p, xref, note) requires us to translate text inside the tag, and we extract translation pairs from this category. In general, the translatable tags correspond to standalone text, and are thus easy to align in the sequence alignment step.
– Transparent:
By contrast, a transparent tag (e.g. b, ph) is a formatting directive embedded as a child element in a translatable tag, and is not always well aligned due to grammatical differences among languages. We keep the transparent tags embedded in the translatable tags.
– Untranslatable:
In the case of untranslatable tags (e.g. sup), we remove the elements. The complete list of tag categorizations can be found in the supplementary material.

Text alignment Figure 3 shows how to extract parallel text segments based on the tag categorization. There are three aligned translatable tags, and they result in three separate translation pairs. The note tag is translatable, so the entire element is removed when extracting the translation pair of the p tag. However, we do not remove nested translatable tags (like the xref tag in this figure) when their *tail*[4] has text, to avoid missing phrases within sentences. Next, we remove the root tag from each translation pair, because the correspondence is obvious. We also remove fine-grained information such as attributes in the XML tags for the dataset; from the viewpoint of real-world usage, we can recover (or copy) the missing information as a post-processing step. As a result of this process, a translation pair can consist of multiple sentences as shown in Example (c) of Figure 1. We do not split them into single sentences, considering a recent trend of context-sensitive machine translation (Bawden et al., 2018; Müller et al., 2018; Zhang et al., 2018; Miculicich et al., 2018). One can use split sentences for training a model, but an important note is that there is no guarantee that all the internal sentences are perfectly aligned. We note that this structure-based alignment process means we do not rely on statistical alignment models to construct our parallel datasets.

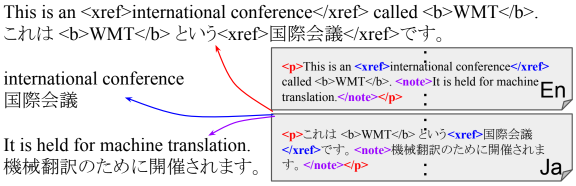

This is an <xref>international conference</xref> called <b>WMT</b>.
これは <b>WMT</b> という<xref>国際会議</xref>です。
international conference
国際会議
It is held for machine translation.
機械翻訳のために開催されます。
<p>This is an <xref>international conference</xref> called <b>WMT</b>. <note>It is held for machine translation.</note></p>
En
<p>これは <b>WMT</b> という<xref>国際会議</xref>です。<note>機械翻訳のために開催されます。</note></p>
Ja

Figure 3: Extracting parallel text segments from aligned XML elements.

[4] For example, the tail of the xref tag in the English example corresponds to the word "called."

Language pair	Training data	Aligned files
English-		
Dutch	100,756	7,160
Finnish	99,759	7,160
French	103,533	7,336
German	103,247	7,336
Japanese	101,480	7,336
Russian	100,332	7,160
Simplified Chinese	99,021	7,160
Finnish-Japanese	101,527	7,927

Table 1: The number of the translation examples in the training data used in our experiments.

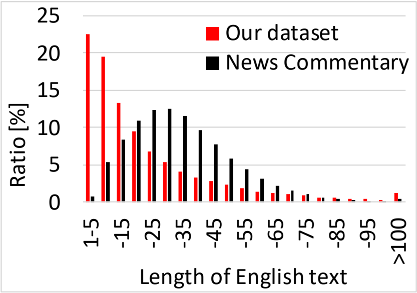


Figure 4: The length statistics of the English text in our English-French and the News Commentary datasets.

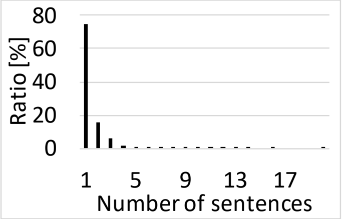


Figure 5: The statistics of the number of English sentences in the English-French translation pairs.

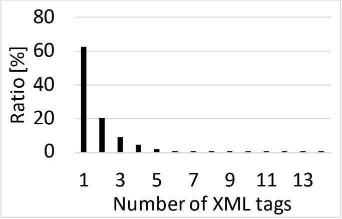


Figure 6: The statistics of the number of XML tags inside the English-French translation pairs.

Filtering We only keep translation pairs whose XML tag sets are consistent in both language sides, but we do not constrain the order of the tags to allow grammatical differences that result in tag reordering. We remove duplicate translation pairs based on exact matching, and separate two sets of 2,000 examples each for development and test sets. There are many possible experimental settings, and in this paper we report experimental results for seven English-based pairs, English-to-{Dutch, Finnish, French, German, Japanese, Russian, Simplified Chinese}, and one non-English pair, Finnish-to-Japanese. The dataset thus provides opportunities to focus on arbitrary pairs of the 17 languages. For each of the possible pairs, the number of training examples (aligned segments) is around 100,000.

2.3 Detailed Dataset Statistics

Table 1 and Figure 4, 5, 6 show more details about the dataset statistics. We take our English-French dataset to show some detailed statistics, but the others also have the consistent statistics because all the pairs are grounded in the same English files.

Text lengths Due to the XML tag-based extraction, our dataset includes word- and phrase-level translations as well as sentence- and paragraph-level translations, and we can see in Figure 4 that there are many short text segments. This is, for example, different from the statistics of the widely-used News Commentary dataset. The text length is defined based on the number of subword tokens, following our experimental setting described below.

Sentence counts Another characteristic of our dataset is that the translation pairs can consist of multiple sentences, and Figure 5 shows the statistics of the number of English sentences in the English-French translation pairs. The number of sentences is determined with the sentence splitter from the Stanford CoreNLP toolkit (Manning et al., 2014).

XML-tag counts As we remove the root tags from the XML elements in our dataset construction process, not all the text segments have XML tags inside them. More concretely, about 25.5% of the translation pairs have at least one internal XML tag, and Figure 6 shows the statistics. For example, Example (a) in Figure 1 has four XML tags, and Example (b) has three.

2.4 Evaluation Metrics

We consider multiple evaluation metrics for the new dataset. For evaluation, we use the *true-cased* and *detokenized* text, because our dataset is designed for an end-user, raw-document setting.

BLEU without XML We include the most widely-used metric, BLEU, without XML tags.

That is, we remove all the XML tags covered by our dataset and then evaluate BLEU. The metric is compatible with the case where we use the dataset for plain text translation without XML. To compute the BLEU scores, we use language-specific tokenizers; for example, we use Kytea (Neubig et al., 2011) for Simplified Chinese and Japanese, and the Moses (Koehn et al., 2007) tokenizer for English, Dutch, Finnish, French, German, and Russian.

Named entities and numbers The online help frequently mentions named entities such as product names and numbers, and accurate translations of them are crucial for users. Frequently, they are not translated but simply copied as English forms. We evaluate corpus-level precision and recall for translation of the named entities and numerical tokens. To extract the named entities and numerical words, we use a rule-based regex script, based on our manual analysis on our dataset. The numerical words are extracted by

“[0-9.,\’/:]*[0-9]+[0-9.,\’/:]*”.

The named entities are defined as

“[.,\’/:a-zA-Z$]*[A-Z]+[.,\’/:a-zA-Z$]*”

appearing in a non-alphabetic language, Japanese, because in our dataset we observe that the alphabetic words in such non-alphabetic languages correspond to product names, country names, function names, etc.

XML accuracy, matching, and BLEU For each output text segment, we use the `etree` module to check if it is a valid XML structure by wrapping it with a dummy root node. Then the XML accuracy score is the number of the valid outputs, divided by the number of the total evaluation examples. We further evaluate how many translation outputs have exactly the same XML structures as their corresponding reference text (an XML matching score). If a translation output matches its reference XML structure, both the translation and reference are split by the XML tags. We then evaluate corpus-level BLEU by comparing each split segment one by one. If an output does not match its reference XML structure, the output is treated as empty to penalize the irrelevant outputs.

3 Machine Translation with XML Tags

We use NMT models to provide competitive baselines for our dataset. This section first describes how to handle our dataset with a sequential NMT model. We then propose a simple constrained beam search for accurately generating XML structures conditioned by source information. We further incorporate multiple copy mechanisms to strengthen the baselines.

3.1 Sequence-to-Sequence NMT

The task in our dataset is to translate text with structured information, and therefore we consider using syntax-based NMT models. A possible approach is incorporating parse trees or parsing algorithms into NMT models (Eriguchi et al., 2016, 2017), and another is using sequential models on linearized structures (Aharoni and Goldberg, 2017). We employ the latter approach to incorporate source-side and target-side XML structures, and note that this allows using standard sequence-to-sequence models without modification.

We have a set of parallel text segments for a language pair $(\mathcal{X}, \mathcal{Y})$, and the task is translating a text segment $x \in \mathcal{X}$ to another $y \in \mathcal{Y}$. Each x in the dataset is represented with a sequence of tokens including some XML tags: $x = [x_1, x_2, \dots, x_N]$, where N is the length of the sequence. Its corresponding reference y is also represented with a sequence of tokens: $y = [y_1, y_2, \dots, y_M]$, where M is the sequence length. Any tokenization method can be used, except that the XML tags should be individual tokens.

To learn translation from x to y, we use a *transformer* model (Vaswani et al., 2017). In our K-layer transformer model, each source token x_i in the k-th $(k \in [1, K])$ layer is represented with

$$h_k^x(x_i) = f(i, h_{k-1}^x) \in \mathbb{R}^d, \qquad (1)$$

where i is the position information, d is the dimensionality of the model, and $h_{k-1}^x = [h_{k-1}^x(x_1), h_{k-1}^x(x_2), \dots, h_{k-1}^x(x_N)]$ is the sequence of the vector representations in the previous layer. $h_0^x(x_i)$ is computed as $h_0^x(x_i) = \sqrt{d} \cdot v(x_i) + e(i)$, where $v(x_i) \in \mathbb{R}^d$ is a token embedding, and $e(i) \in \mathbb{R}^d$ is a positional embedding.

Each target-side token y_j is also represented in a similar way:

$$h_k^y(y_j) = g(j, h_k^x, h_{k-1}^y) \in \mathbb{R}^d, \qquad (2)$$

where only $[h_{k-1}^y(y_1), h_{k-1}^y(y_2), \dots, h_{k-1}^y(y_j)]$ is used from h_{k-1}^y. In the same way as the source-side embeddings, $h_0^y(y_j)$ is computed as $h_0^y(y_j) =$

$\sqrt{d} \cdot v(y_j) + e(j)$. For more details about the parameterized functions f and g, and the positional embeddings, please refer to Vaswani et al. (2017).

Then $h^y_K(y_j)$ is used to predict the next token w by a softmax layer: $p_g(w|x, y_{\leq j}) = \text{softmax}(Wh^y_K(y_j) + b)$, where $W \in \mathbb{R}^{|\mathbb{V}| \times d}$ is a weight matrix, $b \in \mathbb{R}^{|\mathbb{V}|}$ is a bias vector, and $\mathbb{V}$ is the vocabulary. The loss function is defined as follows:

$$L(x, y) = -\sum_{j=1}^{M-1} \log p_g(w = y_{j+1}|x, y_{\leq j}), \quad (3)$$

where we assume that y_1 is a special token BOS to indicate the beginning of the sequence, and y_M is an end-of-sequence token EOS. Following Inan et al. (2017) and Press and Wolf (2017), we use W as an embedding matrix, and we share the single vocabulary $\mathbb{V}$ for both $\mathcal{X}$ and $\mathcal{Y}$. That is, each of $v(x_i)$ or $v(y_j)$ is equivalent to a row vector in W.

3.2 XML-Constrained Beam Search

At test time, standard sequence-to-sequence generation methods do not always output valid XML structures, and even if an output is a valid XML structure, it does not always match the tag set of its source-side XML structure. To generate source-conditioned XML structures as accurately as possible, we propose a simple constrained beam search method. We add three constrains to a standard beam search method. First, we keep track of possible tags based on the source input, and allow the model to open only a tag that is present in the input and has not yet been covered. Second, we keep track of the most recently opened tag, and allow the model to close the tag. Third, we do not allow the model to output EOS before opening and closing all the tags used in the source sentence. Algorithm 1 in the supplementary material shows a comprehensive pseudo code.

3.3 Reformulating a Pointer Mechanism

We consider how to further improve our NMT system, by using multiple *discrete* copy mechanisms. Since our dataset is based on XML-structured technical documents, we want our NMT system to copy (A) relevant text segments in the target language if there are very similar segments in the training data, and (B) named entities (e.g. product names), XML tags, and numbers directly from the source. For the copy mechanisms, we follow the general idea of the *pointer* used in See et al. (2017).

For the sake of discrete decisions, we reformulate the pointer method. Following the previous work, we have a sequence of tokens which are targets of our pointer method: $c = [c(z_1), c(z_2), \ldots, c(z_U)]$, where $c(z_i) \in \mathbb{R}^d$ is a vector representation of the i-th token z_i, and U is the sequence length. As in Section 3.1, we have $h^y_K(y_j)$ to predict the $(j + 1)$-th token. Before defining an attention mechanism between $h^y_K(y_j)$ and c, we append a parameterized vector $c(z_0) = c'$ to c. We expect c' to be responsible for decisions of "not copying" tokens, and the idea is inspired by adding a "null" token in natural language inference (Parikh et al., 2016).

We then define attention scores between $h^y_K(y_j)$ and the expanded c: $a(j, i) = score(h^y_K(y_j), c_i, c)$, where the normalized scoring function $score$ is implemented as a single-head attention model proposed in Vaswani et al. (2017). If the next reference token y_{j+1} is not included in the copy target sequence, the loss function is defined as follows:

$$L(x, y_{\leq j}, c) = -\log a(j, 0), \quad (4)$$

and otherwise the loss function is as follows:

$$L(x, y_{\leq j}, c) = -\log \sum_{i, \text{ s.t. } z_i = y_{j+1}} a(j, i), \quad (5)$$

and then the total loss function is $L(x, y) + \sum_{j=1}^{M-1} L(x, y_{\leq j}, c)$. The loss function solely relies on the cross-entropy loss for single probability distributions, whereas the pointer mechanism in See et al. (2017) defines the cross-entropy loss for weighted summation of multiple distributions.

At test time, we employ a discrete decision strategy for copying tokens or not. More concretely, the output distribution is computed as

$$\delta \cdot p_g(w|x, y_{\leq j}) + (1 - \delta) \cdot p_c(w|x, y_{\leq j}), \quad (6)$$

where $p_c(w|x, y_{\leq j})$ is computed by aggregating $[a(j, 1), \ldots, a(j, U)]$. δ is 1 if $a(j, 0)$ is the largest among $[a(j, 0), \ldots, a(j, U)]$, and otherwise δ is 0.

Copy from Retrieved Translation Pairs Gu et al. (2018) presented a retrieval-based NMT model, based on the idea of translation memory (Silvestre Baquero and Mitkov, 2017). Following Gu et al. (2018), we retrieve the most relevant translation pair (x', y') for each source text x

in the dataset. In this case, we set $[z_1, \ldots, z_U] = [y'_2, \ldots, y'_{M'}]$ and $c = [h^y_K(y'_1), \ldots, h^y_K(y'_{M'-1})]$, where M' is the length of y', and each vector in c is computed by the same transformer model in Section 3.1. For this retrieval copy mechanism, we denote p_c and δ as p_r and δ_r, respectively.

Copy from Source Text To allow our NMT model to directly copy certain tokens from the source text x when necessary, we follow See et al. (2017). We set $[z_1, \ldots, z_U] = [x_1, \ldots, x_N]$ and $c = [h^x_K(x_1), \ldots, h^x_K(x_N)]$, and we denote p_c and δ as p_s and δ_s, respectively.

We have the single vocabulary $\mathbb{V}$ to handle all the tokens in both languages $\mathcal{X}$ and $\mathcal{Y}$, and we can combine the three output distributions at each time step in the text generation process:

$$(1 - \delta_s)p_s + \delta_s(\delta_r p_g + (1 - \delta_r)p_r). \quad (7)$$

The copy mechanism is similar to the multi-pointer-generator method in McCann et al. (2018), but our method employs rule-based discrete decisions. Equation (7) first decides whether the NMT model copies a source token. If not, our method then decides whether the model copies a retrieved token.

4 Experimental Settings

This section describes our experimental settings. We will release the preprocessing scripts and the training code (implemented with PyTorch) upon publication. More details are described in the supplementary material.

4.1 Tokenization and Detokenization

We used the SentencePiece toolkit (Kudo and Richardson, 2018) for sub-word tokenization and detokenization for the NMT outputs.

Without XML tags If we remove all the XML tags from our dataset, the task becomes a plain MT task. We carried out our baseline experiments for the plain text translation task, and for each language pair we trained a joint SentencePiece model to obtain its shared sub-word vocabulary. For training each NMT model, we used training examples whose maximum token length is 100.

With XML tags For our XML-based experiments, we also trained a joint SentencePiece model for each language pair, where one important note is that all the XML tags are treated as user-defined special tokens in the toolkit. This allows us to easily implement the XML-constrained beam search. We also set the three tokens `&amp;`, `&lt;`, and `&gt;` as special tokens.

4.2 Model Configurations

We implemented the transformer model with $K = 6$ and $d = 256$ as a competitive baseline model. We trained three models for each language pair:

"OT" (trained only with text without XML),

"X" (trained with XML), and

"$\mathrm{X_{rs}}$" (XML and the copy mechanisms).

For each setting, we tuned the model on the development set and selected the best-performing model in terms of BLEU scores *without* XML, to make the tuning process consistent across all the settings.

5 Results

Table 2 and 4 show the detailed results on our development set, and for the $\mathrm{X_{rs}}$ model, we also show the results ($\mathrm{X_{rs}^{(T)}}$) on our test set to show our baseline scores for future comparisons. Simplified Chinese is written as "Chinese" in this section.

5.1 Evaluation without XML

We first focus on the two evaluation metrics: BLEU without XML, and named entities and numbers (NE&NUM). In Table 2, a general observation from the comparison of OT and X is that including segment-internal XML tags tends to improve the BLEU scores. This is not surprising because the XML tags provide information about explicit or implicit alignments of phrases. However, the BLEU score of the English-to-Finnish task significantly drops, which indicates that for some languages it is not easy to handle tags within the text.

Another observation is that $\mathrm{X_{rs}}$ achieves the best BLEU scores, except for English-to-French. In our experiments, we have found that the improvement of BLEU comes from the retrieval method, but it degrades the NE&NUM scores, especially the precision. Then copying from the source tends to recover the NE&NUM scores, especially for the recall. We have also observed that using beam search, which improves BLEU scores, degrades the NE&NUM scores. A lesson learned from these results is that work to improve BLEU scores can sometimes lead to degradation of other important metrics.

		NE&NUM		NE&NUM		NE&NUM		NE&NUM
	BLEU	Precision, Recall	BLEU	Precision, Recall	BLEU	Precision, Recall	BLEU	Precision, Recall
	English-to-Japanese		English-to-Chinese		English-to-French		English-to-German	
OT	61.61	89.84, 89.84	58.06	94.91, 93.62	64.07	88.64, 85.64	50.51	88.40, 86.55
X	62.00	92.54, 90.51	58.61	94.56, 93.44	63.98	87.48, 86.98	50.96	88.79, 86.43
X_{rs}	64.25	91.64, 90.98	60.05	94.44, 94.27	63.51	88.42, 85.64	52.91	88.00, 86.78
$\mathrm{X}_{\mathrm{rs}}^{(\mathrm{T})}$	64.34	93.39, 91.75	59.86	93.49, 93.11	65.04	88.98, 88.31	52.69	88.22, 88.45
	English-to-Finnish		English-to-Dutch		English-to-Russian		Finnish-to-Japanese	
OT	43.97	87.58, 84.99	59.54	90.89, 88.59	43.28	89.67, 85.26	54.55	90.45, 89.69
X	42.84	83.17, 85.55	60.18	90.41, 90.26	43.44	87.96, 88.35	54.69	93.47, 89.29
X_{rs}	45.10	86.41, 86.49	60.58	88.76, 90.11	46.73	88.65, 89.55	57.92	93.02, 89.03
$\mathrm{X}_{\mathrm{rs}}^{(\mathrm{T})}$	45.71	87.38, 88.91	61.01	87.66, 90.84	46.44	86.90, 89.59	57.06	93.39, 89.38

Table 2: Automatic evaluation results *without* XML on the development set, and the test set for X_{rs}.

Training data	Our dev set	newstest2014
Our dataset (no XML)	64.07	7.35
w/ 10K news	63.66	14.02
w/ 20K news	64.31	16.30
Only 10K news	0.90	2.66
Only 20K news	2.35	6.72

Table 3: Domain adaptation results (BLEU). The models are tuned on our development set.

Compatibility with other domains Our dataset is limited to the domain of online help, but we can use it as a seed corpus for domain adaptation if our dataset contains enough information to learn basic grammar translation. We conducted a simple domain adaptation experiment in English-to-French by adding 10,000 or 20,000 training examples of the widely-used News Commentary corpus. We used the newstest2014 dataset for evaluation in the news domain. From Table 3, we can see that a small amount of the news-domain data significantly improves the target-domain score, and we expect that our dataset plays a good role in domain adaptation for all the covered 17 languages.

5.2 Evaluation with XML

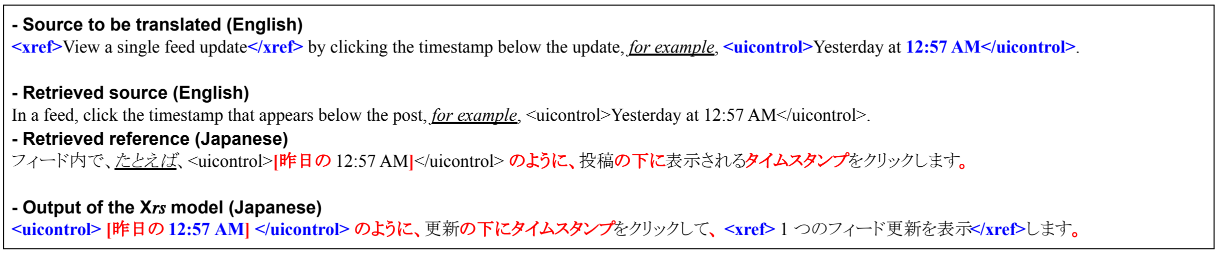
Table 4 shows the evaluation results with XML. Again, we can see that X_{rs} performs the best in terms of the XML-based BLEU scores, but the absolute values are lower than those in Table 2 due to the more rigid segment-by-segment comparisons. This table also shows that the XML accuracy and matching scores are higher than 99% in most of the cases. Ideally, the scores could be 100%, but in reality, we set the maximum length of the translations; as a result, sometimes the model cannot find a good path within the length limitation. Table 5 shows how effective our method is, based on the English-to-Japanese result, and we observed the consistent trend across the different languages. These results show that our method can accurately generate the relevant XML structures.

How to recover XML attributes? As described in Section 2.2, we removed all the attributes from the original XML elements for simplicity. However, we need to recover the attributes when we use our NMT model in the real-world application. We consider recovering the XML attributes by the copy mechanism from the source; that is, we can copy the attributes from the XML elements in the original source text, if the XML tags are copied from the source. Table 6 summarizes how our model generates the XML tags on the English-Japanese development set. We can see in the table that most of the XML tags are actually copied from the source.

Figure 7 shows an example of the output of the X_{rs} model. For this visualization, we merged all the subword tokens to form the standard words. The tokens in blue are explicitly copied from the source, and we can see that the time expression "12:57 AM" and the XML tags are copied as expected. The output also copies some relevant text segments (in red) from the retrieved translation. Like this, we can explicitly know which words are copied from which parts, by using our multiple discrete copy mechanisms. One surprising observation is that the underlined phrase "for example" is missing in the translation result, even though the BLEU scores are higher than those on other standard public datasets. This is a typical error called *under translation*. Therefore, no matter how large the BLEU scores are, we definitely need human corrections (or post editing) before providing the translation results to customers.

	BLEU	XML Acc., Match	BLEU	XML Acc., Match	BLEU	XML Acc., Match	BLEU	XML Acc., Match
	English-to-Japanese		English-to-Chinese		English-to-French		English-to-German	
X	59.77	99.80, 99.55	57.01	99.95, 99.70	61.81	99.60, 99.30	48.91	99.85, 99.25
X_{rs}	62.06	99.80, 99.40	58.43	99.90, 99.60	61.87	99.80, 99.50	51.16	99.75, 99.30
$X_{rs}^{(T)}$	62.27	99.95, 99.60	57.92	99.75, 99.40	63.19	99.80, 99.35	50.47	99.80, 99.20
	English-to-Finnish		English-to-Dutch		English-to-Russian		Finnish-to-Japanese	
X	41.98	99.65, 99.25	57.86	99.60, 99.25	40.72	99.60, 98.95	52.14	99.90, 99.30
X_{rs}	43.57	99.50, 99.25	58.51	99.70, 99.30	44.42	99.75, 99.25	55.20	99.65, 98.90
$X_{rs}^{(T)}$	44.22	99.90, 99.65	60.19	99.90, 99.85	44.25	99.80, 99.35	54.05	99.60, 98.75

Table 4: Automatic evaluation results *with* XML on the development set, and the test set for X_{rs}.



- Source to be translated (English)

<xref>View a single feed update</xref> by clicking the timestamp below the update, *for example*, <uicontrol>Yesterday at 12:57 AM</uicontrol>.

- Retrieved source (English)

In a feed, click the timestamp that appears below the post, *for example*, <uicontrol>Yesterday at 12:57 AM</uicontrol>.

- Retrieved reference (Japanese)

フィード内で、たとえば、<uicontrol>[昨日の 12:57 AM]</uicontrol> のように、投稿の下に表示されるタイムスタンプをクリックします。

- Output of the X*rs* model (Japanese)

<uicontrol> [昨日の 12:57 AM] </uicontrol> のように、更新の下にタイムスタンプをクリックして、<xref> 1 つのフィード更新を表示</xref>します。

Figure 7: An example of the translation results of the X_{rs} model on the English-Japanese test set.

	BLEU	XML Acc., Match
w/ XML constraint	59.77	99.80, 99.55
w/o XML constraint	58.02	98.70, 98.10

Table 5: Effects of the XML-constrained beam search.

	Count
Copied from source text	1,638
Copied from retrieved translation	24
Generated from vocabulary	11

Table 6: Statistics of the generated XML tags.

5.3 Human Evaluation by Professionals

One important application of our NMT models is to help human translators; translating online help has to be precise, and thus any incomplete translations need post-editing. We asked professional translators at a vendor to evaluate our test set results (with XML) for the English-to-{Finnish, French, German, Japanese} tasks. For each language pair, we randomly selected 500 test examples, and every example is given an integer score in $[1, 4]$. A translation result is rated as "4" if it can be used without any modifications, "3" if it needs simple post-edits, "2" if it needs more post-edits but is better than nothing, and "1" if using it is not better than translating from scratch.

Figure 8 shows the summary of the evaluation to see the ratio of each score, and the average scores are also shown. A positive observation for all the four languages is that more than 50% of the translation results are evaluated as complete or useful in post-editing. However, there are still many low-quality translation results; for example, around 30% of the Finnish and German results are evaluated as useless. Moreover, the German results have fewer scores of "4", and it took 12 hours for the translators to evaluate the German results, whereas it took 10 hours for the other three languages. To further make our NMT models useful for post-editing, we have to improve the translations scored as "1".

Detailed error analysis We also asked the translators to note what kinds of errors exist for each of the evaluated examples. All the errors are classified into the six types shown in Table 7, and each example can have multiple errors. The "Formatting" type is our task-specific one to evaluate whether the XML tags are correctly inserted. We can see that the Finnish results have significantly more XML-formatting errors, and this result agrees with our finding that handling the XML tags in Finnish is harder than in other languages, as discussed in Section 5.1. It is worth further investigating such language-specific problems.

The "Accuracy" type covers major issues of NMT, such as adding irrelevant words, skipping important words, and mistranslating phrases. As discussed in previous work (Malaviya et al., 2018), reducing the typical errors covered by the "Accuracy" type is crucial. We have also noticed

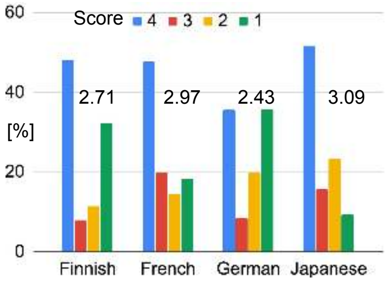


Figure 8: Human evaluation results for the X_{rs} model. "4" is the best score, and "1" is the worst.

	Finnish	French	German	Japanese
Accuracy	30.0	32.8	37.4	37.4
Readability	20.6	20.4	0.8	17.4
Formatting	10.6	0.0	0.8	1.0
Grammar	20.2	10.0	11.4	5.8
Structure	10.2	2.8	2.0	1.2
Terminology	12.0	3.0	2.4	0.6

Table 7: Ratio [%] of six error types.

that the NMT-specific errors would slow down the human evaluation process, because the NMT errors are different from translation errors made by humans. The other types of errors would be reduced by improving language models, if we have access to in-domain monolingual corpora.

Can MT help the localization process? In general, it is encouraging to observe many "4" scores in Figure 8. However, one important note is that it takes significant amount of time for the translators to verify the NMT outputs are good enough. That is, having better scored NMT outputs does not necessarily lead to improving the productivity of the translators; in other words, we need to take into account the time for the quality verification when we consider using our NMT system for that purpose. Previous work has investigated the effectiveness of NMT models for post-editing (Skadina and Pinnis, 2017), but it has not yet been investigated whether using NMT models can improve the translators' productivity alongside the use of a well-constructed translation memory (Silvestre Baquero and Mitkov, 2017). Therefore, our future work is investigating the effectiveness of using the NMT models in the real-world localization process where a translation memory is available.

6 Related Work and Discussions

Automatic extraction of parallel sentences has a long history (Varga et al., 2005), and usually statistical methods and dictionaries are used. By contrast, our data collection solely relies on the XML structure, because the original data have been well structured and aligned. Recently, collecting training corpora is the most important in training NLP models, and thus it is recommended to maintain well-aligned documents and structures when building multilingual online services. That will significantly contribute to the research of language technologies.

We followed the syntax-based NMT models (Eriguchi et al., 2016, 2017; Aharoni and Goldberg, 2017) to handle the XML structures. One significant difference between the syntax-based NMT and our task is that we need to output source-conditioned structures that are able to be parsed as XML, whereas the syntax-based NMT models do not always need to follow formal rules for their output structures. In that sense, it would be interesting to relate our task to source code generation (Oda et al., 2015) in future work.

Our dataset has significant potential to be further expanded. Following the context-sensitive translation (Bawden et al., 2018; Müller et al., 2018; Zhang et al., 2018; Miculicich et al., 2018), our dataset includes translations of multiple sentences. However, the translatable XML tags are separated, so the page-level global information is missing. One promising direction is thus to create page-level translation examples. Finally, considering the recent focus on multilingual NMT models (Johnson et al., 2017), multilingually aligning the text will enrich our dataset.

7 Conclusion

We have presented our new dataset for XML-structured text translation. Our dataset covers 17 languages each of which can be either source or target of machine translation. The dataset is of high quality because it consists of professional translations for an online help domain. Our experiments provide baseline results for the new task by using NMT models with an XML-constrained beam search and discrete copy mechanisms. We further show detailed human analysis to encourage future research focusing on how to apply machine translation to help human translators in practice.

Acknowledgements

We thank anonymous reviewers and Xi Victoria Lin for their helpful feedbacks.

References

Roee Aharoni and Yoav Goldberg. 2017. Towards String-To-Tree Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor. 2018. Proceedings of the Third Conference on Machine Translation: Shared Task Papers. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-Sequence Attentional Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to Parse and Translate Improves Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search Engine Guided Neural Machine Translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5133–5140.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling. In *Proceedings of the 5th International Conference on Learning Representations*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Chaitanya Malaviya, Pedro Ferreira, and André F. T. Martins. 2018. Sparse and Constrained Attention for Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 2*, pages 370–376.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The Natural Language Decathlon: Multitask Learning as Question Answering. *arXiv preprint arXiv:1806.08730*.

Paul Michel and Graham Neubig. 2018. MTNT: A Testbed for Machine Translation of Noisy Text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-Level Neural Machine Translation with Hierarchical Attention Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A Large-Scale Test Set for the Evaluation of Context-Aware Pronoun Translation in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533.

Y. Oda, H. Fudaba, G. Neubig, H. Hata, S. Sakti, T. Toda, and S. Nakamura. 2015. Learning to Generate Pseudo-Code from Source Code Using Statistical Machine Translation. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 574–584.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Ofir Press and Lior Wolf. 2017. Using the Output Embedding to Improve Language Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Andrea Silvestre Baquero and Ruslan Mitkov. 2017. Translation Memory Systems Have a Long Way to Go. In *Proceedings of the Workshop Human-Informed Translation and Interpreting Technology*, pages 44–51.

Inguna Skadina and Mārcis Pinnis. 2017. NMT or SMT: Case Study of a Narrow-domain English-Latvian Post-editing Project. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 373–383.

Daniel Varga, Laszlo Németh, Peter Halácsy, Andras Kornai, Viktor Trón, and Victor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the Transformer Translation Model with Document-Level Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.