A Parallel Evaluation Data Set of Software Documentation with Document Structure Annotation

Bianka Buschbeck and **Miriam Exel**
SAP SE
Dietmar-Hopp-Allee 16, 69189 Walldorf, Germany
{bianka.buschbeck,miriam.exel}@sap.com

Abstract

This paper accompanies the *software documentation data set for machine translation*, a parallel evaluation data set of data originating from the *SAP Help Portal*, that we released to the machine translation community for research purposes. It offers the possibility to tune and evaluate machine translation systems in the domain of corporate software documentation and contributes to the availability of a wider range of evaluation scenarios. The data set comprises of the language pairs English to Hindi, Indonesian, Malay and Thai, and thus also increases the test coverage for the many low-resource language pairs. Unlike most evaluation data sets that consist of plain parallel text, the segments in this data set come with additional metadata that describes structural information of the document context. We provide insights into the origin and creation, the particularities and characteristics of the data set as well as machine translation results.

1 Introduction

The *software documentation data set for machine translation* is created by SAP[1] as evaluation data for the machine translation (MT) research community. The data originates from the *SAP Help Portal*[2] that contains documentation for SAP products and user assistance for product-related questions. The current language scope is English (EN) to Hindi (HI), Indonesian (ID), Malay (MS) and Thai (TH). The data has been processed in a way that makes it suitable as development and test data for machine translation purposes. For each language pair about 4k segments are available, split into development and test data.

The segments are provided in their document context and are annotated with additional metadata about the document structure. The metadata provides information such as document and paragraph boundaries as well as the segment's text type, for example whether it is a title or table element. Such information will surely be valuable when developing and evaluating document-level MT or for tuning systems for specific text types in order to increase the overall translation quality in this domain.

The *software documentation data set for machine translation* as described in this paper is available under the Creative Commons license Attribution-Non Commercial 4.0 International (CC BY-NC 4.0). It is available on GitHub under https://github.com/SAP/software-documentation-data-set-for-machine-translation. It has been released by SAP for the 7th Workshop on Asian Translation (WAT 2020).[3]

We will first provide some context, explaining the role of test data in machine translation and referring to related work (Section 2). We will then describe the origin of the *software documentation data set* in Section 3, including the data preparation and data selection. Section 4 is dedicated to the characteristics of the data set. Benchmarking results of MT systems on the test sets are provided in Section 5. Section 6 concludes.

2 Context

Test sets are typically used for comparison in MT evaluation campaigns, such as WMT[4] and WAT[5],

[1] https://www.sap.com

[2] https://help.sap.com

[3] https://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2020/index.html

[4] Yearly *Conference on Machine Translation*, hosting a number of shared tasks. See http://www.statmt.org/wmt19/ and the findings paper (Barrault et al., 2019) for the 2019 occurrence.

[5] Yearly *Workshop on Asian Translation*, hosting several shared translation tasks. See http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2019/index.html and the overview paper (Nakazawa et al., 2019) for the 2019 occurrence.

Proceedings of the 7th Workshop on Asian Translation, pages 160–169
December 4, 2020. ©2020 Association for Computational Linguistics

in which different participants, or rather their systems, compete against each other on specific tasks. Subsequently, those test sets are typically also used in research publications to demonstrate the effectiveness of the approach at hand and to compare to previous results. As such, test sets play a crucial role in showing the progress of machine translation.

For many years, test sets have been prevalently drawn from news articles.[6] However, to be able to assess machine translation quality in a wider range of usage scenarios, it is important to also evaluate in other domains than news, and thus to create and establish test sets from a wider range of domains. Clearly, specific usage scenarios have other challenges than what is represented in the news domain. Thus, quality results, and claims about human parity (e.g. Hassan et al., 2018), that have been achieved in the news domain can usually not be directly transferred to other domains. Accordingly, data sets and shared tasks have been created for other domains as well, e.g. biomedical[7] and patents[8]. With the *software documentation data set*, we provide the possibility to tune and evaluate MT systems in the domain of corporate software documentation, and thus contribute to a clearer picture of the quality of machine translation across domains. Similarly, the focus of machine translation has often been on high-resource language pairs, such as English-German. With an evaluation data set for four language pairs that are rather on the lower end of availability of resources, we contribute to a better test coverage for the many low-resource language pairs.

With the recent improvements in machine translation quality, up to claims of human parity, flaws in the evaluation setups and interpretation of results have been pointed out (Toral et al., 2018; Läubli et al., 2018; Bojar et al., 2018). Subsequently, more emphasis has been put on carefully evaluating machine translation, in particular to be able to evaluate segments within their document context, e.g. by Barrault et al. (2019). By creating data sets that consist of documents corresponding to help pages, we contribute to this endeavor. The document structure annotation can also provide additional useful information during human evaluation. Similarly, machine translation approaches have started to look beyond translating independent sentences. Methods for taking more context into account have emerged, with the goal to improve the translation quality (Miculicich et al., 2018; Maruf and Haffari, 2018; Yu et al., 2020, amongst others). By providing development and test data with document context and metadata, we hope to strengthen such developments.

Resources that are related to the data set at hand in terms of the covered domain are the data sets from the WMT16 shared task of machine translation of IT domain (Bojar et al., 2016, Section 4) and the documentation data set by Salesforce (Hashimoto et al., 2019). The data set from the IT translation shared task consists of answers from a help desk, thus it covers a different text type than software documentation that likely also comes with a different style. Furthermore, the focus of the data set is on European languages, and it does not contain more context than short one-paragraph answers. The data set described and experimented with by Hashimoto et al. (2019) is very similar in nature to ours. Note however that the language scope is different: all language pairs in the data set by Salesforce are rather high-resource.

3 Origin of the data

3.1 Data sources

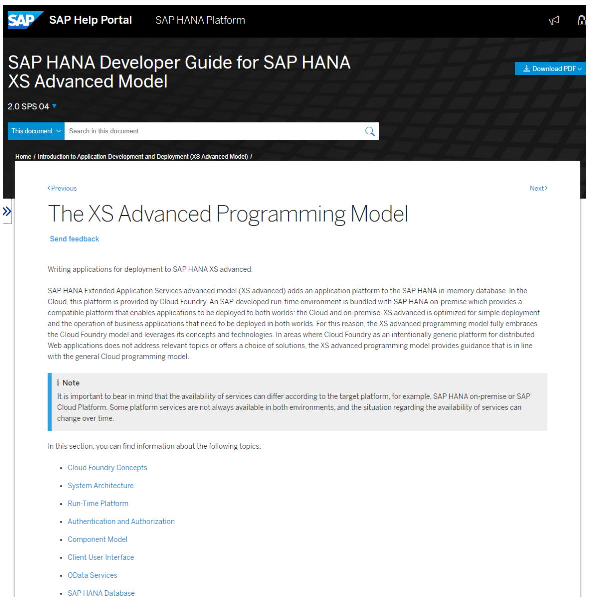
The contents of the *software documentation data set for machine translation* originate from the *SAP Help Portal* that contains SAP product documentation and user assistance for product-related questions. As it describes the use of software, it is rather technical in nature. In contrast to general textual data, it is highly structured, i.e. it contains many tables, lists, links, examples as well as code snippets. The textual presentation and page layout follow a similar structure across documents to obtain a coherent appearance of corporate help pages. This explains some of the particularities of this data set, described in more detail in Section 3.3. Figure 1 shows an example of such a help page.

The content of the help pages is authored by domain experts and then translated by professional translators that are specialized in the translation of SAP content. Hence, the source data is of excellent quality as well as its translations. Furthermore, the translations of the proposed documentation data set were created without machine translation in the

[6] See `http://matrix.statmt.org/test_sets/list` for example.

[7] See, for example, the biomedical translation task at WMT19: `http://www.statmt.org/wmt19/biomedical-translation-task.html`

[8] See, for example, the JBO Patent corpus used at WAT: `http://lotus.kuee.kyoto-u.ac.jp/WAT/patent/`



SAP HANA Developer Guide for SAP HANA XS Advanced Model

The XS Advanced Programming Model

Writing applications for deployment to SAP HANA XS advanced.

SAP HANA Extended Application Services advanced model (XS advanced) adds an application platform to the SAP HANA in-memory database. In the Cloud, this platform is provided by Cloud Foundry. An SAP-developed run-time environment is bundled with SAP HANA on-premise which provides a compatible platform that enables applications to be deployed to both worlds: the Cloud and on-premise. XS advanced is optimized for simple deployment and the operation of business applications that need to be deployed in both worlds. For this reason, the XS advanced programming model fully embraces the Cloud Foundry model and leverages its concepts and technologies. In areas where Cloud Foundry as an intentionally generic platform for distributed Web applications does not address relevant topics or offers a choice of solutions, the XS advanced programming model provides guidance that is in line with the general Cloud programming model.

i Note
It is important to bear in mind that the availability of services can differ according to the target platform, for example, SAP HANA on-premise or SAP Cloud Platform. Some platform services are not always available in both environments, and the situation regarding the availability of services can change over time.

In this section, you can find information about the following topics:

- Cloud Foundry Concepts
- System Architecture
- Run-Time Platform
- Authentication and Authorization
- Component Model
- Client User Interface
- OData Services
- SAP HANA Database

Figure 1: Screenshot of a page from *SAP Help Portal*

loop, so there is no bias to any MT system.

3.2 Data preparation

In this section, we will describe the source format of the data and how we processed it for the *software documentation data set for machine translation*.

English source texts are edited using DITA[9], an XML-based format, well suited for authoring, structuring and publishing content with a high potential of reuse. For translation, SAP uses computer-assisted translation (CAT) tools, such as SDL Trados Studio[10]. which transform DITA-XML format into XLIFF (XML Localization Interchange File Format)[11] used in translation. As it keeps track of the text structure and inline markup of the source texts, this information can be transferred to the target language after translation. For its use in SDL Trados Studio, SDL developed SDLXLIFF[12], a special flavor of XLIFF. SDLXLIFF files are highly structured bilingual files that contain both the source document text and its translation.

Figure 2 shows a fragment of an SDLXLIFF document that demonstrates the information used to provide parallel text as well as structural annotation of the document context. SDLXLIFF files usually cover one document, the content of which is presented in textual order. A translation unit `<trans-unit>` is a sequence of consecutive text for the source and the target language, in this case for English and Hindi. It is split into sentences by the Trados sentence segmenter, as shown under `<seg-source>` and `<target>` in Figure 2. Segments are enumerated using the `mid` attribute. We use this information to order the translation pairs consecutively for each document and to count segments that belong to a text unit or paragraph (see metadata columns 2 and 4 in Table 1).

The information about the structural type of a translation unit in the document is conveyed by the `<sdl:cxts>` context value. Text can be used in a title, a section, a table, an example or an itemized list. In the example in Figure 2, the translation unit occurs in the context `<sdl:cxt id="4"/>` which corresponds to an unordered list, see Figure 3 for the text element declarations.

Contextual text types are declared for each XLIFF file and vary depending on the document content and its source. To reduce the number of text types that come with naming variants and different levels of granularity, we mapped them to six common and self-explanatory categories for the *software documentation data set*: `title`, `section`, `table_element`, `list_element`, `example`, `unspecified`.

Parallel segments, positional metadata and text type were extracted from each SDLXLIFF document using the Saxon parser[13] with an XSLT stylesheet. We provide the resulting data in text format, as it is common practice in machine translation, in three sentence-parallel files: source text, target text and document context metadata. The metadata file contains the following five columns:

1. Document ID
2. Segment ID in the document that indicates the contextual order (restarts from 1 in each document)
3. Text Unit ID in the document that indicates segments that occur in consecutive order (starts from 1 in each document). Segments with the same Text Unit ID make up one text block consisting of multiple sentences, for example a paragraph.

[9] https://en.wikipedia.org/wiki/Darwin_Information_Typing_Architecture

[10] https://www.sdl.com/software-and-services/translation-software/sdl-trados-studio/

[11] http://xml.coverpages.org/xliff.html

[12] http://producthelp.sdl.com/sdl%20trados%20studio/client_en/Edit_View/XLIFF_File_Format.htm

[13] http://saxon.sourceforge.net/

```
<sdl:cxts>
  <sdl:cxt id=”4”/>
</sdl:cxts>
<trans-unit id=”e64c34f4-7383-4ffd-8a68-a3bc3e90c9ec”>
  <source>If you cannot find the course, then your course is an external event.
          Start over and select External event.</source>
  <seg-source>
    <mrk mtype=”seg” mid=”11”>If you cannot find the course, then your course is an
                              external event.</mrk>
    <mrk mtype=”seg” mid=”12”>Start over and select External event.</mrk>
  </seg-source>
  <target>
    <mrk mtype=”seg” mid=”11”>यदि आप पाठ्यक्रम नहीं ढूंढ पा रहे हैं, तो आपका पाठ्यक्रम बाहरी
                              सामग्री है.</mrk>
    <mrk mtype=”seg” mid=”12”>फिर से शूरू करें और बाहरी ईवेंट का चयन करें.</mrk>
  </target>
</trans-unit>
```

Figure 2: Example of a translation unit in XLIFF format

```
<cxt-defs xmlns=”http://sdl.com/FileTypes/SdlXliff/1.0”>
  <cxt-def id=”1” type=”sdl:title”>
    <fmt id=”1”/>
  </cxt-def>
  <cxt-def id=”2” type=”link text” descr=”Line of text for a link.”>
    <fmt id=”2”/>
  </cxt-def>
  <cxt-def id=”3” type=”section” descr=”Organizational division of a topic.”>
    <fmt id=”4”/>
  </cxt-def>
  <cxt-def id=”4” type=”unordered list” descr=”List of items”>
    <fmt id=”4”/>
  </cxt-def>
</cxt-defs>
```

Figure 3: Example of a definition of textual elements in an XLIFF file

4. Segment ID in Text Unit (starts from 1 in each Text Unit)

5. Textual element that describes the structural type of the segment. Values are `title`, `section`, `table_element`, `list_element`, `example`, `unspecified`

After the XLIFF processing, the contextual annotation of the content of the SAP Help page in Figure 1 would look as shown in Table 1. It is document 79 with 17 segments and 12 text units. There is a paragraph marked as text unit 3 consisting of 6 sentences. Each list element is considered an individual text unit.[14]

3.3 Particularities of the data

As pointed out in Section 3.1, help pages are composed in a way that allows for high reuse of textual content and patterns. For coherent appearance, their structure is intended to be clear and uniform. This has some impact on the kind of text segments we find in software documentation documents.

1. There is a lot of *redundancy*, i.e. source-target pairs occur several times across documents or even within the same document. This concerns titles, table headers, table values or even complete sentences.

2. As the help pages on the *SAP Help Portal* contain many tables and list items, many *translation segments are short*, sometimes consisting of just a number or one word. List and table elements are presented as individual text units and are translated independently within their document context.

3. There is a large number of *short documents* reflecting the segmentation of help page content into reusable units.

These particularities impact the creation of the evaluation data sets (Section 3.4) and the characteristics of the final data set (Section 4).

[14] The *Note* displayed on the help page in Figure 1 is not part of the document the data was extracted from. It is inserted at some later stage of the publishing process.

English source	Metadata				
	1	2	3	4	5
The XS Advanced Programming Model	79	1	1	1	title
Writing applications for deployment to SAP HANA XS advanced.	79	2	2	1	section
SAP HANA Extended Application Services advanced model (XS advanced) adds an application platform to the SAP HANA in-memory database.	79	3	3	1	section
In the Cloud, this platform is provided by Cloud Foundry.	79	4	3	2	section
An SAP-developed run-time environment is bundled with SAP HANA on-premise which provides a compatible platform that enables applications to be deployed to both worlds: the Cloud and on-premise.	79	5	3	3	section
XS advanced is optimized for simple deployment and the operation of business applications that need to be deployed in both worlds.	79	6	3	4	section
For this reason, the XS advanced programming model fully embraces the Cloud Foundry model and leverages its concepts and technologies.	79	7	3	5	section
In areas where Cloud Foundry as an intentionally generic platform for distributed Web applications does not address relevant topics or offers choice, the XS advanced programming model provides guidance that is in line with the general Cloud programming model.	79	8	3	6	section
In this section, you can find information about the following topics:	79	9	4	1	section
Cloud Foundry Concepts	79	10	5	1	list_element
System Architecture	79	11	6	1	list_element
Run-Time Platform	79	12	7	1	list_element
Authentication and Authorization	79	13	8	1	list_element
Component Model	79	14	9	1	list_element
Client User Interface	79	15	10	1	list_element
OData Services	79	16	11	1	list_element
SAP HANA Database	79	17	12	1	list_element

Table 1: Presentation of source segments and text structure annotation

3.4 Data selection

Ideally, test and development sets for machine translation meet the requirements of being[15] (i) *representative* for a given test or usage scenario, in our case for a given domain, covering well its specific terminology, its syntax and style, (ii) *free of duplicates* and redundancy, (iii) *balanced*, i.e., ideally sampled from a larger set of data, so that the content is spread over various topics.

When building evaluation sets as collections of single sentences (or sentence pairs), it is rather straightforward to adhere to these criteria. However, when creating them for whole documents, the absence of duplicates and redundancy as well as content balance are more challenging. This is particularly true for our help page content that displays similar structuring and repetitions, see Section 3.3. Obviously, duplicate sentence pairs cannot simply be removed if we want to keep the contextual order of segments.

Let us define *redundancy* as the ratio of all source-target pairs to unique source-target pairs in a data set. Figure 4 shows the redundancy for all data at our disposal (in blue). We see that it differs depending on the language pair. To some extent, this can be explained by the amount of documents used for extraction. While for English to Malay (EN-MS) and to Thai (EN-TH) we had several thousands of original documents at hand, for English to Hindi (EN-HI) and to Indonesian (EN-ID) only a couple of hundred documents were available that had less overlap and thus show less redundancy.

To meet the requirements of test and development data, we made an effort to reduce this redundancy by selecting documents that are less prone to have content present in other documents. The following indicators were calculated to be used in the selection process:

- Document redundancy ratio: percentage of unique parallel segments to all parallel segments in a document (to flag documents that contain duplicates).

[15] General guidance for assembling (test) data can be found in Megerdoomian (2003, Sec. 1.6.5), Jurafsky and Martin (2008, Sec. 4.3), Resnik and Lin (2010, Sec. 2.6), amongst others.

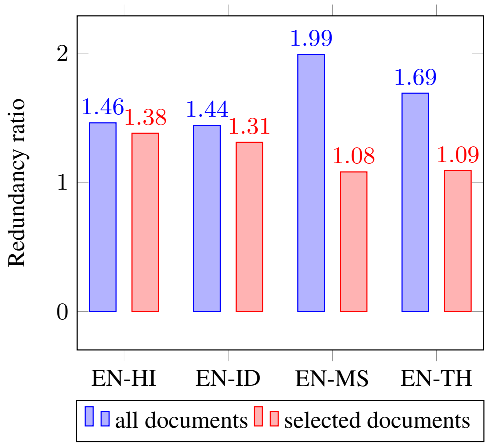


Figure 4: Redundancy reduction: redundancy in all data vs. the data that was selected for the data set.

- Number of segments in the document (to flag documents with little content, and hence context).
- Average number of source words per segment (to keep documents with longer segments).
- Cross-document redundancy of a document with respect to all documents (to flag documents that contain a large number of segments that occur in many other documents). We first created a frequency list of source segments of all documents. Then, for each segment of a document, their overall document frequencies were summed up and divided by the number of segments in the document. This ratio is high if the document contains many segments that occur in many documents.
- Document double indicator. It turned out that for EN-MS and EN-TH, many documents were almost identical but for one or two segments. Overall cross-document redundancy does not help in this case, as source-target pairs occur only twice. The document double indicator flags documents that contain a large percentage of source-target pairs that occur exactly twice in the complete data.

For each language pair, we selected a subset of all available documents that contains about 4k sentences that meets the requirements as much as possible by calibrating the indicators. For EN-MS and EN-TH, all five indicators were used to reduce the redundancy as much as possible. For EN-HI and EN-ID, only the document redundancy ratio and the number of segments per document were considered, as there were less documents to choose from and there was less redundancy to start with. With this approach, we successfully obtained a data set with less duplicates across documents, see Figure 4 (in red).

4 Characteristics of the data set

The selected documents with reduced redundancy, see Section 3.4, were divided into development and test data sets and constitute the *software documentation data set for machine translation*. We will look into its characteristics in this section. Table 2 provides an overview over the size of and redundancy within the respective data set.

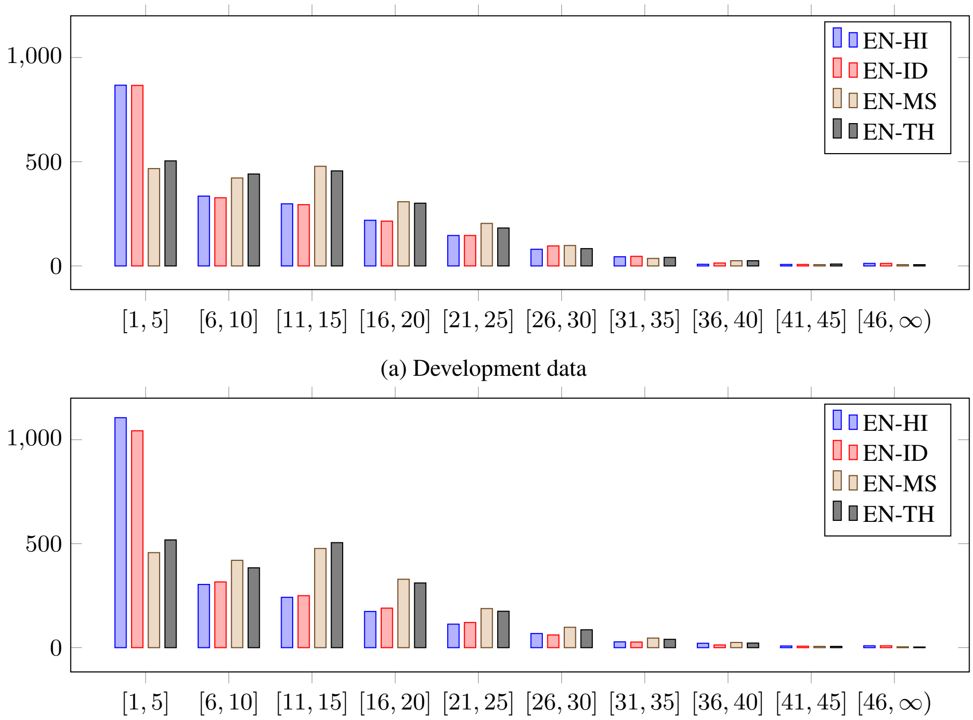
While the number of segments of the development and test sets are in the same range across language pairs, the number of documents and the total amount of words are different for EN-HI and EN-ID compared to the other two language pairs. This difference is also reflected in the distribution of words per segment, see Figure 5: there is a larger number of short segments for EN-HI and EN-ID. For EN-MS and EN-TH, we see a more balanced distribution of short and medium length segments in both, development and test sets. Figure 6 shows the distributions of textual element annotations in the data sets' metadata. They explain, to some extent, the distribution of segment length: We see a larger number of `section` segments for EN-MS and EN-TH. Sections usually contain longer segments than table elements, which are frequent for EN-HI and EN-ID.

Finally, we look at the redundancy in the released data set, i.e. the number of all source-target pairs over the number of unique source-target pairs, shown in Table 2. As expected from Figure 4, there is more redundancy for EN-HI and EN-ID, which ties in with the larger number of shorter segments. They are more likely to reoccur across documents.

In summary, we conclude that the data sets for EN-HI and EN-ID are comparable concerning the criteria analyzed in this section. They differ somewhat from the EN-MS and EN-TH data sets that also have characteristics in common. We would have preferred to provide a more homogeneous data set. However, given the different sizes and features of the original resources and the constraints imposed by adding contextual metadata, this was not feasible. On the other hand, the charts in this section indicate that the development and test sets of each language pair share the same characteris-

	# of documents		# of parallel segments		# of source words		Data set redundancy	
	dev	test	dev	test	dev	test	dev	test
EN-HI	78	76	2,016	2,073	20,662	18,128	1.33	1.14
EN-ID	66	74	2,023	2,037	21,159	18,164	1.26	1.11
EN-MS	210	197	2,050	2,050	26,654	26,758	1.04	1.05
EN-TH	207	205	2,048	2,050	25,759	25,426	1.03	1.05

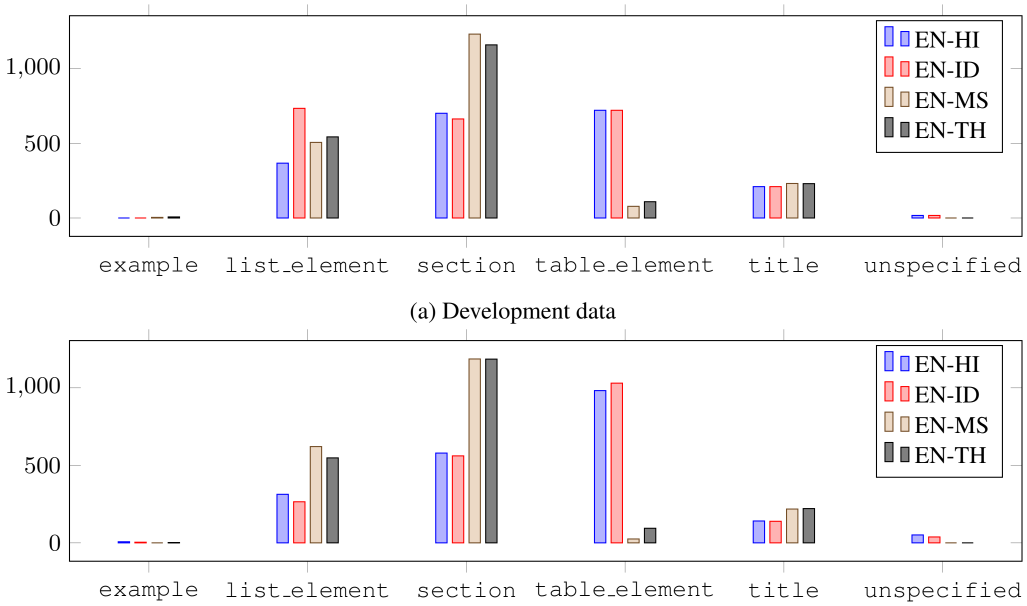
Table 2: Statistics on development and test data sets



(a) Development data

(b) Test data

Figure 5: Length distributions of source segments



(a) Development data

(b) Test data

Figure 6: Distribution of textual element annotations

	Provider	BLEU	ChrF	BLEU*
EN-HI	WAT	13.67	0.3193	13.73
	Online	32.97	0.5681	
EN-ID	WAT	30.39	0.5828	30.11
	Online	58.18	0.7662	
EN-MS	WAT	31.85	0.5968	31.95
	Online	42.31	0.7005	
EN-TH	WAT	31.29	0.2933	31.28
	Online	68.80	0.6443	
HI-EN	WAT	14.54	0.3987	14.39
	Online	50.19	0.7375	
ID-EN	WAT	23.25	0.4917	23.05
	Online	52.94	0.7552	
MS-EN	WAT	25.32	0.5120	25.36
	Online	48.52	0.7313	
TH-EN	WAT	9.56	0.3244	9.56
	Online	27.73	0.5717	

Table 3: Machine translation results on the test set

tics, i.e. their segment length distribution, the types of textual elements as well as their word counts are comparable. This makes the development sets well suited to optimize an MT model towards the translation of the corresponding test set.

5 Machine translation results

In this section, we provide reference MT results on the test sets of the *software documentation data set for machine translation*. They should serve as comparison for future research evaluated on this data set. The results can be found in Table 3.

The first reference MT result is based on the baseline system of the WAT 2020 *NICT-SAP IT and Wikinews Task*[16], provided by the organizers of the task. There are two multilingual systems, one for EN to {HI, ID, MS, TH} and one for {HI, ID, MS, TH} to EN. They are trained on Wikinews data from the Asian Language Treebank (ALT) project (Riza et al., 2016) and IT data from Opus[17] (Ubuntu, GNOME and KDE4) (Tiedemann, 2012). A transformer-big configuration (Vaswani et al., 2017) was used. More details can be found in the workshop overview (Organizers, to appear). The second reference MT system that we report on is a popular general purpose online MT provider. The contextual metadata of the test set is not used in either reference system.

We report case-sensitive BLEU (Papineni et al., 2002) and ChrF (Popović, 2016) scores as calculated by *sacrebleu*[18] (Post, 2018). The BLEU scores for EN-TH are generated based on character-segmented input. For the WAT system, we additionally report the official BLEU scores as provided by the task evaluation (cf. column BLEU* in Table 3).[19] It uses *Moses' multi-bleu.perl*.[20]

The online MT provider outperforms the WAT baseline for all language pairs by a wide margin on our test sets. Given the low-resource setting of the baseline WAT system and its role as a simple baseline, this is not very surprising. We expect that by using more data, in particular in-domain data, may it be monolingual or parallel, the gap could be narrowed or even closed. Other interesting questions are whether bilingual systems would perform better than the multilingual baseline, how to better exploit small quantities of in-domain data, how to leverage the available contextual metadata, and whether neural network parametrization could help to improve results in this low-resource setting.

6 Conclusion

We released the *software documentation data set for machine translation* to the MT research community, a high-quality real-world data set with content from the *SAP Help Portal*. To our knowledge, it is the first data collection with explicit text structure annotation and the first IT-specific evaluation data set for English to Hindi, Indonesian, Malay and Thai. It will advance automatic quality assessment of context-aware MT systems, giving users the flexibility to consider all or only selected or no text structure metadata. Moreover, it facilitates the development and testing of machine translation systems for low-resource language pairs for the translation of software documentation in a corporate context. As a starting point, we provided MT results that serve as benchmarks in future research.

[16] See http://lotus.kuee.kyoto-u.ac.jp/WAT/NICT-SAP-Task/ for the details.

[17] http://opus.nlpl.eu/

[18] Version strings are BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.13 and chrF2+numchars.6+space.false+version.1.4.13.

[19] See the SOFTWARE* links on http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html. Pages accessed on September 15th, 2020.

[20] See http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/automatic_evaluation_systems/tools.html for details on the used tools.

Acknowledgments

We would like to thank Raj Dabre, Masao Utiyama and Eiichiro Sumita from NICT for their support, for training the baseline WAT systems and for providing us with the translations. We would also like to thank our colleagues Dominic Jehle and Matthias Huck for comments on the draft.

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Kazuma Hashimoto, Raffaella Buschiazzo, James Bradbury, Teresa Marshall, Richard Socher, and Caiming Xiong. 2019. A high-quality multilingual dataset for structured documentation translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 116–127, Florence, Italy. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*, 2nd edition. Prentice Hall.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? A case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284, Melbourne, Australia. Association for Computational Linguistics.

Karine Megerdoomian. 2003. Text mining, corpus building and testing. In *Handbook for Language Engineers*, chapter 6. CSLI Publications.

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, Brussels, Belgium. Association for Computational Linguistics.

Toshiaki Nakazawa, Nobushige Doi, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th Workshop on Asian Translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 1–35, Hong Kong, China. Association for Computational Linguistics.

WAT Organizers. to appear. Overview of the 7th Workshop on Asian Translation. In *Proceedings of the 7th Workshop on Asian Translation*.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2016. chrF deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504, Berlin, Germany. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Philip Resnik and Jimmy Lin. 2010. Evaluation of nlp systems. In *The Handbook of Computational Linguistics and Natural Language Processing*, chapter 11, pages 271–295. John Wiley & Sons, Ltd.

H. Riza, M. Purwoadi, Gunarso, T. Uliniansyah, A. A. Ti, S. M. Aljunied, L. C. Mai, V. T. Thang, N. P. Thai, V. Chea, R. Sun, S. Sam, S. Seng, K. M. Soe, K. T. Nwet, M. Utiyama, and C. Ding. 2016. Introduction of the Asian Language Treebank. In *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? Reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Better document-level machine translation with Bayes' rule. *Transactions of the Association for Computational Linguistics*, 8:346–360.