A Study on the Effectiveness of Large Language Models for Translation with Markup

Raj Dabre raj.dabre@nict.go.jp
Hideki Tanaka hideki.tanaka@nict.go.jp
National Institute of Information and Communications Technology, Japan

Bianka Buschbeck bianka.buschbeck@sap.com
Miriam Exel miriam.exel@sap.com
SAP SE, Walldorf, Germany

Abstract

In this paper we evaluate the utility of large language models (LLMs) for translation of text with markup in which the most important and challenging aspect is to correctly transfer markup tags while ensuring that the content, both, inside and outside tags is correctly translated. While LLMs have been shown to be effective for plain text translation, their effectiveness for structured document translation is not well understood. To this end, we experiment with BLOOM and BLOOMZ, which are open-source multilingual LLMs, using zero, one and few-shot prompting, and compare with a domain-specific in-house NMT system using a detag-and-project approach for markup tags. We observe that LLMs with in-context learning exhibit poorer translation quality compared to the domain-specific NMT system, however, they are effective in transferring markup tags, especially the large BLOOM model (176 billion parameters). This is further confirmed by our human evaluation which also reveals the types of errors of the different tag transfer techniques. While LLM-based approaches come with the risk of losing, hallucinating and corrupting tags, they excel at placing them correctly in the translation.

1 Introduction

Recent work involving Large Language Models (LLMs) has shown impressive performance in various Natural Language Processing (NLP) tasks. These models have the ability to perform few-shot (or in-context) learning based on prompts, an alternative to fine-tuning, requiring only a forward pass of the neural network (Brown et al., 2020). Prompts are instructions in natural language given as input to LLMs along with a test sequence, allowing a few examples (i.e. few-shot) to be fed to the model at test time. Researchers have shown that LLMs via prompting can be effective as Machine Translation (MT) systems (Brown et al., 2020; Wei et al., 2022; Chowdhery et al., 2022; Zhang et al., 2023; Hendy et al., 2023; Bawden and Yvon, 2023), whose quality approaches that of traditional encoder-decoder neural MT (NMT) systems trained or fine-tuned on parallel corpora. The majority of the aforementioned research has been conducted on plain text, neglecting the practical application of MT for text containing markup, see Table 1, where the challenge is to properly transfer markup tags *within* the translatable content from the source to the target language. Given that a significant portion of web-based content and proprietary or business documents requiring translation comes in structured for-

Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track, pages 148–159
September 4–8, 2023, Macau SAR, China.

en	Click <uicontrol>Prepayment</uicontrol>.
ja	<uicontrol>前払</uicontrol>をクリックします。

Table 1: Example with inline markup (in gray), taken from Buschbeck et al. (2022).

mats like HTML pages or Microsoft Office files, it is important to understand the effectiveness of LLMs in handling this task.

In this paper, we conduct the first of its kind study on the use of LLMs for translation of text with markup where the transfer of markup tags, or tag placement, is as important as the translation of the content inside and outside the tags. We use SAP's Asian language dataset (Buschbeck et al., 2022) focusing on translation involving Japanese, Chinese, Korean and English and experiment with zero, one and few-shot prompting of the open-source multilingual BLOOM and BLOOMZ LLMs (Le Scao et al., 2022; Muennighoff et al., 2022). We compare our results against those obtained via a general-domain MT system, M2M[1] (Fan et al., 2021), as well as a domain-specific in-house NMT system that handles markup tags via a detag-and-project approach. Our multi-metric evaluations using BLEU, chrF and COMET reveal that while LLMs exhibit relatively poorer translation quality compared to the domain-specific NMT system, they are often competitive with a general-domain MT system, and that the degree to which LLMs are able to transfer markup tags out-of-the-box depends on the prompting strategy and the model size. This is further confirmed by our human evaluation that reveals the various error types associated with different tag transfer approaches. Notably, the 176 billion parameter model employing few-shot prompting outperforms the detag-and-project strategy in terms of tag positioning, demonstrating its strong potential. Our study focuses on the impact of example retrieval approaches, number of shots and their ordering. It provides insights for MT practitioners, and should encourage further research in this area.

2 Related Work

This paper focuses on an evaluation of LLMs for the translation of text with markup. We briefly review the related work in this area.

2.1 Structured Document Translation

Hashimoto et al. (2019) present a data set from the IT domain that features structure via inline markup, and corresponding MT results using a constrained beam search approach for decoding. Further, Hanneman and Dinu (2020) compare different data augmentation methods with a detag-and-project approach, and evaluate on data from legal documents from the European Union. The methods for tag transfer in Zenkel et al. (2021) are also related, even though they focus on inserting the tags into a fixed human translation. In contrast to these works, Buschbeck et al. (2022), who also release an evaluation dataset for structured document translation of Asian languages, propose to use existing multilingual pre-trained NMT models as black-boxes for translating texts with inline elements directly. They show that these models perform surprisingly well at transferring markup tags during translation despite not being explicitly trained to handle structured content. In this paper, we further investigate black-box approaches for structured document translation, focusing specifically on LLMs.

2.2 Language Model Prompting

Ever since the intoduction of GPT-3 (Brown et al., 2020), which showed that LLMs are excellent zero and few-shot text learners, there has been a lot of interest in using LLMs for various NLP tasks. GPT-3 has been followed by models like BLOOM (Le Scao et al., 2022) and XGLM (Lin

[1] M2M is not explicitly trained to handle markup tags.

et al., 2022) which are multilingual supporting between 40 and 120 languages. These LLMs have shown that by providing them with some examples of a downstream task, in what is known as prompting, they are able to produce outputs of reasonably high quality. We specifically focus on their ability to handle structured content, something that has not been explored so far. Muennighoff et al. (2022) have shown that multi-task fine-tuning of LLMs can improve their performance, especially in a zero-shot setting, which we also study with BLOOMZ which is an extension of BLOOM.

3 Methodology

The methodology employed in this work focuses on prompting approaches, namely, the template or format of instructions fed to the LLMs along with input sequences to be translated, as well as example retrieval techniques.

3.1 Prompting Approach

For our experiments, we use an N-shot approach, selecting N translation pairs (S_i, T_i) from an example pool. We then use these examples (or shots) in a templated form to prompt the LLM. The template is of the following form for all experiments in this paper:

> "Translate the following sentence from E to F: [S_1] [T_1] $\cdots$ Translate the following sentence from E to F: [S_N] [T_N] Translate the following sentence from E to F: [S_t]"

where E is the source language, F is the target language, and S_t is the test example for which we want to obtain a translation. We use structure-aware prompting, where we retrieve examples containing markup tags for test sentences with tags, and examples without markup tags for test sentences without tags. Unless explicitly mentioned, few-shot results are reported with 4 examples. Note that in the template each source and target language sentence is wrapped in opening and closing square brackets ([,]). After the model produces outputs, we remove the prompted prefix and retain the first segment produced by the model within the [and] brackets as the model's translation.

3.2 Example Retrieval

In this paper, we primarily use LABSE-based embedding similarity[2] (Feng et al., 2022) to extract fitting examples from the example pool. We compute cosine similarity between the LABSE representations of the test sentence and the source side of the example set, and retrieve N pairs such that their sources have the highest similarity. We employ the LABSE model because it is a multilingual model capable of calculating the similarity between sentences in any language. In our analyses, we also use BM25[3] (Robertson et al., 1995) and the chrF metric (Popović, 2015) for retrieval. BM25 is a bag-of-words[4] based retrieval algorithm which is widely used for information retrieval. It is a probabilistic model which computes the similarity between a query and a document as a function of the term frequencies in the document and the query. In our case, the query is the test sentence and the document is the source side of the example set. chrF is a character level n-gram based metric which is used for machine translation evaluation. We calculate it between the test sentence and the source sides of the example set, and extract examples that maximize chrF. We would like to investigate whether leveraging chrF for example retrieval can improve the translations' chrF scores.

[2] https://huggingface.co/setu4993/LaBSE

[3] https://github.com/dorianbrown/rank_bm25

[4] Since Japanese, Chinese and Korean are unsegmented, for simplicity we treat each character as a word.

4 Experimental Setup

In this section, we describe the datasets, language models and baselines used in our experiments to evaluate the utility of LLMs for structured document translation.

4.1 Datasets

We experiment with the Software Documentation Data Set (Buschbeck et al., 2022), henceforth the SAP dataset, which covers Japanese, Chinese, Korean translation from/to English.[5] It belongs to the domain of enterprise software documentation and consists of high-quality, n-way parallel structured documents in form of XML or XLIFF files. Using this dataset allows us to show how LLMs perform on domain-specific technical data, and whether LLMs can preserve the structural markup during translation. For the experiments, we use the data in the provided `text-dita-translatables` format, with 2,011 and 2,002 segments as development and test data respectively. We use the development set as example pool for example retrieval and report results on the test set.

4.2 Language Models

Our main results focus on the BLOOM model and its multi-task fine-tuned variant BLOOMZ, both of which support 46 languages and contain around 7.1 billion parameters. We also employ the BLOOM model with 176 billion parameters for analysis focusing on model size and translation quality. Note that BLOOM is not officially trained for Japanese and Korean but it is still able to handle them potentially due to unintentional inclusion of these languages. We use the Transformers library (version 4.27.0.dev0) by HuggingFace which supports decoding using BLOOM and BLOOMZ. We apply 32-bit floating point precision for greedy search with batch sizes of 2 and generate 128 additional tokens on a 40GB-A100 GPU. For the 176 billion parameter model, we use a batch size of 1 and 8 GPUs. 8-bit decoding is employed via Transformers' integration of the bitsandbytes[6] library (Dettmers et al., 2022).

4.3 Baseline and Upperbound

We compare against two MT baselines: one that is publicly available but markup-agnostic, and another that is an in-house system that can be considered in-domain for software documentation and thus serves as an upper bound for the performance achievable with current NMT systems. The publicly available system is the M2M 1.2 billion parameter model, and we use a beam of size 4 for decoding. The in-house system is a corporate MT engine by SAP that uses the Transformer architecture and that is trained on a multitude of data sources including the contents of company-internal translation memories. These comprise parallel texts from the test domain of software documentation; however, note that it is a multi-domain system that has not been fine-tuned to the test domain specifically. For the tag transfer, a detag-and-project approach along the lines of Hanneman and Dinu (2020) is used.

4.4 Evaluation Metrics

We follow the evaluation method which encompasses both lexical and structural content, as presented in Buschbeck et al. (2022), wherein the MT output and its reference are decomposed into lexical content (sequences are stripped from XML tags, noted *lex*) and structural content (sequences are stripped from lexical content, noted *tag*) before running the automatic metrics. We also compute automatic scores for the unmodified translations (mix of lexical and structural content, noted *raw*). The automatic metrics we report in this paper are BLEU (Papineni et al.,

[5] `https://github.com/SAP/software-documentation-data-set-for-machine-translation`

[6] `https://github.com/TimDettmers/bitsandbytes`

2002) and chrF[7] (Popović, 2015) obtained using the SacreBLEU toolkit (Post, 2018). We apply appropriate tokenization for *raw*, *lex* and *tag*[8] BLEU. The *raw* and *lex* tokenizations depend on the target language and are chosen correspondingly for English[9], Japanese[10], Chinese[11] and Korean[12]. We also report COMET (Rei et al., 2020) using the WMT'22[13] model for the *lex* content as it is the current best practice in MT evaluation.

5 Results and Analysis

We now present our results for translation with markup for the experimental setup lined out in Section 4. We provide a detailed analysis of the impact of various factors on the performance of LLMs on this task. A human evaluation will follow in Section 6.

5.1 Main Results

Table 2 contains the main results of translating text with markup, comparing the LLMs BLOOM and BLOOMZ with and without in-context learning with the multilingual translation model M2M and the corporate in-house MT model. Overall, across the metrics and language pairs, zero-shot configurations lead to poor results, with BLOOMZ, being multilingually fine-tuned, having an advantage over BLOOM. However, including one and four translation examples with the model input (one-shot and few-shot) consistently improves the performances of both BLOOM and BLOOMZ. Both lexical and structural scores improve, showing that the LLMs learn from the provided examples. Note that the relative improvements as well as absolute scores observed with BLOOM in one- and few-shot configurations are larger compared to those obtained with BLOOMZ for all translation directions. See also Section 5.2 for further discussion of this phenomenon. Interestingly, although BLOOM is not officially trained for Japanese and Korean, it still performs well on these languages, especially in the few-shot configuration.

When comparing to the baselines, we can observe that few-shot BLOOM, on average, seems to be roughly on par with M2M according to the reported metrics, with M2M performing better for some language pairs (e.g. en↔ko) and BLOOM for others (e.g. en↔zh). The in-house MT model, that has likely seen more in-domain training data than the other models, outperforms all other models across all metrics and translation directions.

With regards to the metrics themselves, we can see that *lex* BLEU, chrF and COMET are roughly correlated with each other. However, note that the difference in translation quality between the LLMs and the in-house system looks a lot larger with the string-based metrics than with COMET. Given that COMET is known to have the highest correlation with human annotations, BLEU and chrF can be used as reasonable approximates, at least in this paper, which is why we rely mainly on BLEU for the rest of the paper.

5.2 Analysis: Impact of the number of examples

We observed that increasing the number of examples from 1 to 4 had a positive impact on the results of both BLOOM and BLOOMZ. Therefore, taking Japanese to English and English to Japanese translation as a case study, we explore the impact of an increasing number of examples. Specifically, we consider up to 16 retrieved examples when prompting the models, the results for which are shown in Figure 1. We observe that, for both translation directions,

[7] `nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.3.1`

[8] `nrefs:1|case:mixed|eff:no|tok:none|smooth:exp|version:2.3.1`

[9] `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`

[10] `nrefs:1|case:mixed|eff:no|tok:ja-mecab-0.996-IPA|smooth:exp|version:2.3.1`

[11] `nrefs:1|case:mixed|eff:no|tok:zh|smooth:exp|version:2.3.1`

[12] `nrefs:1|case:mixed|eff:no|tok:ko-mecab-0.996/ko-0.9.2-KO|smooth:exp|version:2.3.1`

[13] `https://huggingface.co/Unbabel/wmt22-comet-da`

	en→ja	en→ko	en→zh	ja→en	ko→en	zh→en
M2M	42.1 (35.3, 76.8)	34.6 (27.1, 75.2)	49.2 (43.4, 79.5)	29.0 (24.8, 13.1)	37.0 (25.9, 61.3)	40.2 (29.8, 61.1)
In-house	73.8 (71.3, 91.5)	69.6 (64.8, 90.5)	80.4 (78.2, 93.8)	60.8 (47.4, 80.2)	56.2 (43.0, 71.9)	63.9 (51.2, 77.4)
zero-shot						
BLOOM	3.0 (0.2, 13.9)	2.9 (0.3, 18.7)	3.0 (0.3, 16.4)	3.0 (0.9, 15.0)	5.8 (1.1, 34.5)	8.0 (1.3, 53.4)
BLOOMZ	12.6 (6.9, 47.5)	7.2 (2.6, 30.4)	30.8 (28.0, 42.4)	15.7 (13.5, 7.8)	11.8 (7.9, 8.8)	25.6 (24.0, 17.7)
one-shot						
BLOOM	31.3 (21.8, 75.4)	20.7 (11.4, 66.7)	49.7 (42.6, 88.6)	30.5 (18.2, 66.6)	24.4 (12.4, 51.7)	41.9 (30.6, 76.5)
BLOOMZ	22.3 (14.6, 64.2)	10.1 (5.5, 27.9)	45.1 (38.4, 84.1)	24.8 (15.4, 50.8)	14.6 (8.0, 30.2)	37.8 (28.1, 70.3)
few-shot						
BLOOM	36.0 (26.3, 79.1)	24.1 (13.9, 67.0)	53.8 (46.6, 94.1)	33.5 (20.3, 69.2)	27.4 (14.2, 56.8)	44.4 (31.7, 76.2)
BLOOMZ	27.3 (19.6, 62.4)	17.1 (8.8, 56.0)	47.8 (41.6, 81.1)	27.9 (17.8, 51.5)	20.3 (11.2, 37.2)	41.1 (30.7, 71.2)
M2M	53.2 (40.2, 92.1)	50.3 (34.2, 95.8)	57.5 (37.5, 93.5)	56.1 (53.8, 45.7)	60.2 (54.7, 89.7)	63.6 (58.4, 91.5)
In-house	81.4 (75.8, 99.9)	78.5 (69.2, 99.9)	82.6 (72.9, 99.9)	80.1 (77.2, 98.1)	77.4 (74.2, 97.5)	81.9 (79.4, 98.1)
zero-shot						
BLOOM	10.0 (0.7, 54.2)	10.6 (1.0, 57.0)	11.8 (0.7, 57.5)	16.1 (12.5, 34.9)	18.5 (12.1, 58.0)	19.0 (10.8, 72.9)
BLOOMZ	22.9 (11.0, 60.7)	15.9 (4.3, 48.9)	35.4 (24.8, 56.3)	37.9 (39.0, 31.4)	28.2 (27.0, 34.1)	49.2 (50.5, 42.9)
one-shot						
BLOOM	43.6 (27.9, 89.1)	34.5 (16.5, 80.7)	58.7 (37.5, 93.9)	51.8 (45.7, 84.3)	42.1 (35.0, 79.6)	63.7 (58.6, 90.9)
BLOOMZ	33.4 (18.8, 77.5)	19.7 (8.0, 51.5)	54.2 (33.6, 89.7)	45.4 (40.6, 72.6)	30.4 (26.4, 53.6)	59.2 (54.6, 85.3)
few-shot						
BLOOM	47.9 (32.2, 91.4)	39.0 (20.1, 84.6)	62.4 (41.0, 97.1)	54.5 (48.1, 88.4)	45.4 (38.0, 83.9)	65.7 (60.2, 94.1)
BLOOMZ	38.2 (24.3, 78.9)	28.1 (11.3, 74.7)	55.4 (36.3, 87.6)	49.1 (44.7, 73.6)	36.9 (31.9, 64.1)	63.0 (58.6, 86.3)
M2M	0.846	0.799	0.844	0.795	0.802	0.806
In-house	0.945	0.919	0.923	0.901	0.886	0.895
zero-shot						
BLOOM	0.435	0.438	0.436	0.531	0.575	0.546
BLOOMZ	0.681	0.604	0.775	0.745	0.650	0.780
one-shot						
BLOOM	0.796	0.679	0.854	0.810	0.747	0.851
BLOOMZ	0.756	0.592	0.837	0.771	0.684	0.828
few-shot						
BLOOM	0.817	0.712	0.867	0.823	0.765	0.859
BLOOMZ	0.783	0.653	0.849	0.806	0.731	0.850

Table 2: BLEU (top), chrF (middle) and COMET (bottom) scores obtained with BLOOM and BLOOMZ pretrained models in zero-, one- and few-shot (4) configurations, compared to the pretrained M2M model and the in-house MT engine. Scores are presented as *raw* (*lex*, *tag*) following the metrics presented in Section 4.4. COMET scores are only computed for *lex*.

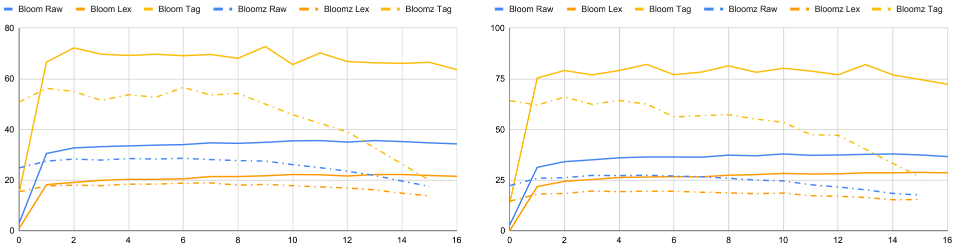

Figure 1: Impact of the number of examples/shots (0 to 16) on the *raw*, *lex* and *tag* BLEU scores of translations obtained by BLOOM and BLOOMZ for ja→en (left) and en→ja (right).

while increasing the number of examples beyond 4 results in a slight improvement in translation quality using BLOOM, the opposite happens with BLOOMZ. Specifically, beyond 5 to 6 examples the quality of BLOOMZ starts dropping with lowest scores for 16 examples. Note

Model	zero-shot	one-shot	few-shot
BLOOM 7b1	3.0 (0.2, 13.9)	31.3 (21.8, 75.4)	36.0 (26.3, 79.1)
BLOOM 176b	3.6 (0.3, 14.1)	41.4 (32.3, 85.9)	45.3 (35.9, 91.9)
M2M		42.1 (35.3, 76.8)	
In-house		73.8 (71.3, 91.5)	

Table 3: *raw*, *lex* and *tag* BLEU scores for the 7.1 billion (7b1) and 176 billion (176b) parameter BLOOM models in comparison to the baselines for en→ja.

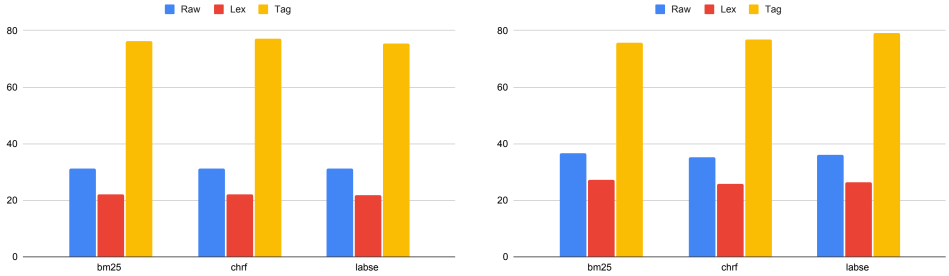

Figure 2: Impact of the example retrieval approach on the *raw*, *lex* and *tag* BLEU scores of English to Japanese translations obtained by BLOOM, for 1-shot (left) and 4-shot (right).

that the BLOOMZ model is a fine-tuned version of BLOOM on xP3 (Muennighoff et al., 2022) which is a multilingual multitask dataset. There is a key difference between the training styles of BLOOM and BLOOMZ, namely that BLOOM is trained on long documents with no specific task in mind, whereas BLOOMZ is trained on supervised task-specific data. Therefore, the latter is not well suited for handling increasing lengths of inputs since the fine-tuning step causes it to forget how to rely on longer context. Although BLOOMZ is superior to BLOOM in a zero-shot setting, it is not suitable for use when large number of examples are available.

5.3 Analysis: Impact of model size

All aforementioned results use BLOOM(Z) models of 7.1 billion parameters, but the largest BLOOM model contains 176 billion parameters and we now study the impact of increasing the model size. We evaluate again for 0, 1 and 4 shots, focusing only on English to Japanese translation due to computational constraints. We present the results in Table 3. It is clear that using the large BLOOM model brings about a large jump in the *raw*, *lex* and *tag* scores as compared to the small BLOOM model. By using four examples, the large model is able to surpass the M2M model; however, it falls far behind the in-house model in terms of *raw* and *lex* BLEU. This is not much of a surprise as BLOOM and M2M are general-domain models, whereas the corporate in-house model has seen substantial training data from the software documentation domain and related domains. Note that in terms of *tag* BLEU the large few-shot BLOOM model can well compete with the detag-and-project approach of the corporate in-house model, indicating that it has the ability to transfer structure effectively from the source to the translation. A more fine-grained analysis for exactly the four presented models will follow in Section 6.

5.4 Analysis: Impact of the example retrieval approach

For the results presented so far, we used LABSE to select the examples for one- and few-shot translation. We now compare to BM25 and chrF (cf. Section 3.2) for English to Japanese translation. See Figure 2 for the results. Overall, we observe minor differences between the retrieval approaches. However, in a few-shot setting, LABSE tends to give the best *tag* BLEU scores.

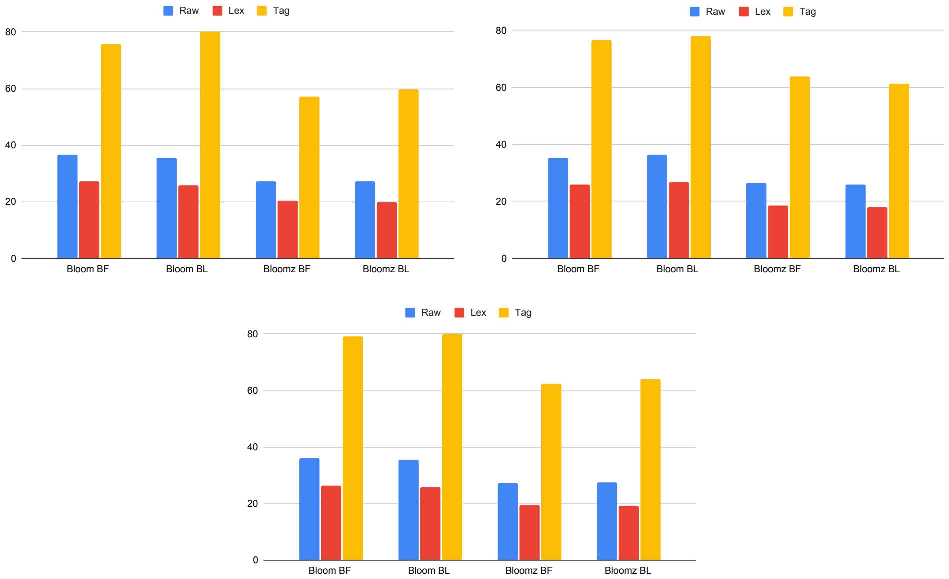

Figure 3: Impact of the order of examples on the *raw*, *lex* and *tag* BLEU scores of English to Japanese few-shot translations obtained by BLOOM and BLOOMZ, for different example retrieval approaches: BM25 (top-left), chrF (top-right) and LABSE (bottom).

5.5 Analysis: Impact of the order of examples

In the few-shot experiments presented thus far, the examples were always ordered *best first* (BF), meaning the best examples (according to the example retrieval approach) are at the beginning of the prompt and the worse example at the end. We now explore the impact of this ordering. Specifically, we reverse this order for few-shot translation for English to Japanese, which we call *best last* (BL). We report the results in Figure 3. We observe that while the *raw* BLEU scores are not largely affected, the *lex* BLEU scores are often reduced by keeping the best examples closest to the test sentence being translated. However, an opposite effect is observed on the *tag* BLEU scores. For BM25 for example, we observe that the *tag* BLEU scores for BL are higher than BF by 6.1 points for translation with BLOOM. Therefore, we recommend that the appropriate ordering be used depending on what evaluation metric is most important for the task at hand. However, further investigation is required to understand why this ordering has such a large impact on the *tag* BLEU scores.

6 Human Evaluation

As the automatic lexical matching metrics used in this paper have their limitations in measuring MT quality (Freitag et al., 2022), and evaluating tag placement automatically is a non-trivial task without a standardized methodology, we perform human evaluation to assess the correctness of translation and tag placement. We focus on the language pair English to Japanese, for which translations of BLOOM 7b1 and BLOOM 176b both few-shot, M2M and the in-house NMT system (see Table 3) were assessed regarding translation quality (Section 6.1) and tag placement (Section 6.2). From the test set, we randomly selected 200 source sentences containing tags and their corresponding translations of the four selected systems. For translation quality assessment, the tags were removed, as tag placement was evaluated separately. Assessing text quality and tag handling separately enables a more accurate understanding.

	Tester 1		**Tester 2**		**Average**	
Model	CharacTER	TER	CharacTER	TER	CharacTER	TER
BLOOM 7b1 few-shot	41.65	61.14	43.03	63.36	42.34	62.25
BLOOM 176b few-shot	26.56	48.02	28.86	52.49	27.71	50.26
M2M	44.12	58.55	45.42	62.59	44.77	60.57
In-house	8.15	13.86	10.17	20.33	**9.16**	**17.09**

Table 4: Results of minimal post-editing of 200 sentences by two translators for English to Japanese measured in CharacTER ↓ and TER ↓

6.1 Post-editing evaluation

MT quality can be efficiently measured using minimal post-editing. It is more reliable than rating as translators are required to edit the translations, which at the same time reveals the encountered problems. By measuring the edit distance between the MT and its post-edited version – a common praxis in the translation industry – the quality of different models can be ranked. We report two metrics: TER (translation edit rate) (Snover et al., 2006) that measures the post-editing effort on the token level and CharacTER (Wang et al., 2016) for character-level edit distance. For TER, the implementation of the SacreBLEU toolkit (Post, 2018)[14] is used. The four sets of 200 translations were post-edited by two professional translators specialized in the domain. Segments were presented in random order. Table 4 shows the outcome.

Assessing the post-editing effort, there is a consensus among testers, with tester 2 being marginally stricter. The inter-annotator agreement, calculated as the Pearson correlation coefficient, yields 0.83 for TER and 0.86 for CharacTER. Both edit distance metrics confirm that the smaller BLOOM model and M2M require significantly more post-editing than the large BLOOM model. The least edits were required for the in-house model, our upperbound baseline. As post-edition was performed on the text without tags, these result could be related to the *lex* BLEU scores of the four selected models in Table 3. Knowing that the data selected for human evaluation is only a subset of the test data, it is still surprising that M2M, being of comparable quality to few-shot BLOOM 176b according to BLEU, was found on the same quality level of few-shot BLOOM 7b1. For both models, M2M and BLOOM 7b1, post-editing effort is massive. Although translations from BLOOM 176b necessitate significantly less post-editing, they cannot be considered practically valuable translations.

6.2 Tag placement evaluation

To assess tag placement independently from translation quality, we also chose post-editing as evaluation method, but this time only tags could be added, moved, renamed, or removed by the testers. The instructions included to never modify any target text so that the editing was restricted to opening and closing tags, their names and syntax. If the translation did not contain the content where the tags should be placed, the testers were instructed to skip the segment. Additionally, testers were asked to indicate whether the content inside tag pairs was indeed translated or just copied from the source. Tag placement was evaluated by 5 testers, but each segment was only evaluated once, as the task was rather deterministic and did not allow the variance one would expect in translation.

The results of the tag placement evaluation of the four systems are shown in Table 5. We report the percentage of tags that were not modified during the post-editing task (*correct*), tags that the testers could not place because the translation did not allow for it (*skipped*), and tags

[14] Signature: `nrefs:1|case:lc|tok:tercom|norm:yes|punct:yes|asian:yes|version:2.3.1`

Model	%Tags						Untranslated
	Correct	Skipped	Wrong				
			Missing	Position	Tag	Hallucinated	
BLOOM 7b1	81.73	14.19	2.28	0.49	1.31	3.43	3.5
BLOOM 176b	**92.66**	6.53	**0.00**	**0.33**	0.49	1.96	3.5
M2M	85.64	8.81	0.65	1.14	3.75	0.16	74.0
In-house	86.46	**2.28**	**0.00**	11.26	**0.00**	**0.00**	**1.5**

Table 5: Results of human tag placement evaluation for English to Japanese

that were modified by the testers (*wrong*). For the latter, we further analyse the post-editing modifications, and report in which way the tags are problematic: tags can be missing in the MT output (*missing*), they can be placed in the wrong position (*position*), the tag itself can be corrupted in some way and/or have the wrong name (*tag*), and the tag can be hallucinated in the MT output (*hallucinated*). We furthermore report the percentage of segments that contain *untranslated* (copied) content between tag pairs.

The results reveal that the large few-shot BLOOM model effectively transfers and accurately places markup tags in translations. However, it may occasionally hallucinate tags or use incorrect tag names. These effects are more pronounced with the small BLOOM model, which looses some tags, while being quite accurate for the transferred tags. In contrast, the in-house MT model's detag-and-project method avoids losing, hallucinating, or corrupting tags but is less precise in placing them accurately in the translation. M2M struggles to perform translation and tag transfer simultaneously, often failing to translate content between markup tags and just copying the source. This issue affects 74% of M2M translations. We should also note the number of tags in skipped translations, which correspond directly to the translation quality, see Section 6.1. As testers could not place tags in translations due to low quality and missing content, we assume that the system's tag placement was rather off.

This tag post-editing study is complementary to the automatic evaluation scores presented in Section 5. In contrast to the *raw* metrics, it evaluates tag transfer and placement independent of translation quality. The *tag* metrics only cover the transfer of tags to the translation and their order to some extend, but not their placement within the translation. This detailed human analysis provides valuable insights into the specific shortcomings of each approach, from which improvement measures or fall-back strategies can be derived.

7 Conclusion

We explored various LLMs and a specialized MT system to assess their ability to translate structured documents in the software documentation domain, focusing on both the translation quality and the transfer of markup elements. The investigation of different prompting approaches showed that LLMs learn from in-domain examples and are capable to produce correct text markup in the target language. With this respect, the foundation model BLOOM is more responsive to prompting than its fine-tuned variant BLOOMZ. We also observed that the large-scale BLOOM model with few-shot prompting largely outperforms its smaller cousins in both translation quality and tag placement. However, this comes at a higher price and with subpar performance, which raises doubts about its practical usefulness for commercial translation purposes. While LLMs excel at transferring structural markup, most likely because they were trained on it, none of the investigated models achieve the translation accuracy of a dedicated machine translation system. Nevertheless, this opens up interesting possibilities for future research, such as the combination of LLMs and MT systems to achieve the best of both worlds.

References

Bawden, R. and Yvon, F. (2023). Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Buschbeck, B., Dabre, R., Exel, M., Huck, M., Huy, P., Rubino, R., and Tanaka, H. (2022). A multilingual multiway evaluation data set for structured document translation of Asian languages. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 237–245.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. (2022). Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M., and Joulin, A. (2021). Beyond English-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2022). Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Freitag, M., Rei, R., Mathur, N., Lo, C.-k., Stewart, C., Avramidis, E., Kocmi, T., Foster, G., Lavie, A., and Martins, A. F. (2022). Results of WMT22 Metrics shared task: Stop using BLEU–neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68.

Hanneman, G. and Dinu, G. (2020). How should markup tags be translated? In *Proceedings of the Fifth Conference on Machine Translation*, pages 1160–1173.

Hashimoto, K., Buschiazzo, R., Bradbury, J., Marshall, T., Socher, R., and Xiong, C. (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.

Hendy, A., Abdelrehim, M., Sharaf, A., Raunak, V., Gabr, M., Matsushita, H., Kim, Y. J., Afify, M., and Awadalla, H. H. (2023). How good are GPT models at machine translation? A comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Lin, X. V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., Pasunuru, R., Shleifer, S., Koura, P. S., Chaudhary, V., O'Horo, B., Wang, J., Zettlemoyer, L., Kozareva, Z., Diab, M., Stoyanov, V., and Li, X. (2022). Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T. L., Bari, M. S., Shen, S., Yong, Z.-X., Schoelkopf, H., et al. (2022). Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

Popović, M. (2015). chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. (2020). COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Robertson, S. E., Walker, S., and Hancock-Beaulieu, M. M. (1995). Large test collection experiments on an operational, interactive system: Okapi at TREC. *Information Processing & Management*, 31(3):345–360.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Wang, W., Peter, J.-T., Rosendahl, H., and Ney, H. (2016). CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Zenkel, T., Wuebker, J., and DeNero, J. (2021). Automatic bilingual markup transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533.

Zhang, B., Haddow, B., and Birch, A. (2023). Prompting large language model for machine translation: A case study. *arXiv preprint arXiv:2301.07069*.