Adaptive Machine Translation with Large Language Models

Yasmin Moslem
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
yasmin.moslem@adaptcentre.ie

Rejwanul Haque
ADAPT Centre
Department of Computing
South East Technological University
Carlow, Ireland
rejwanul.haque@adaptcentre.ie

John D. Kelleher
ADAPT Centre
School of Computer Science
Technological University Dublin
Dublin, Ireland
john.kelleher@adaptcentre.ie

Andy Way
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
andy.way@adaptcentre.ie

Abstract

Consistency is a key requirement of high-quality translation. It is especially important to adhere to pre-approved terminology and adapt to corrected translations in domain-specific projects. Machine translation (MT) has achieved significant progress in the area of domain adaptation. However, real-time adaptation remains challenging. Large-scale language models (LLMs) have recently shown interesting capabilities of in-context learning, where they learn to replicate certain input-output text generation patterns, without further fine-tuning. By feeding an LLM at inference time with a prompt that consists of a list of translation pairs, it can then simulate the domain and style characteristics. This work aims to investigate how we can utilize in-context learning to improve real-time adaptive MT. Our extensive experiments show promising results at translation time. For example, LLMs can adapt to a set of in-domain sentence pairs and/or terminology while translating a new sentence. We observe that the translation quality with few-shot in-context learning can surpass that of strong encoder-decoder MT systems, especially for high-resource languages. Moreover, we investigate whether we can combine MT from strong encoder-decoder models with fuzzy matches, which can further improve translation quality, especially for less supported languages. We conduct our experiments across five diverse language pairs, namely English-to-Arabic (EN-AR), English-to-Chinese (EN-ZH), English-to-French (EN-FR), English-to-Kinyarwanda (EN-RW), and English-to-Spanish (EN-ES).

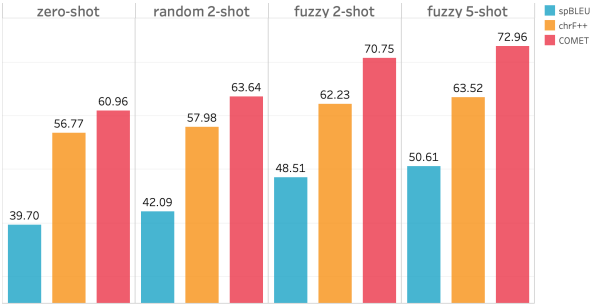

Figure 1: Evaluation results for GPT-3.5 zero-shot, and few-shot translation with random context or fuzzy matches. Average scores across EN-AR, EN-ES, EN-FR, and EN-ZH language pairs. While using a random context outperforms zero-shot translation, using fuzzy matches reveals the best results.

1 Introduction

Adaptive MT is a type of machine translation that utilizes feedback from users to improve the quality of the translations over time. Feedback usually includes corrections to previous translations, terminology and style guides, as well as ratings of the quality of the translations. This can be particularly useful for domain-specific scenarios, where baseline MT systems may have insufficient relevant data to accurately translate certain terms or phrases. There are still several challenges to effectively incorporate user feedback into the translation process, especially at inference time. In this work, we use a relatively wide definition of adaptive MT to refer to learning from similar translations (fuzzy matches) found in approved translation memories (TMs) on the fly (Farajian et al., 2017; Wuebker et al., 2018; Peris and Casacuberta, 2019; Etchegoyhen et al., 2021), as well as real-time terminology-constrained MT (Hokamp and Liu, 2017; Post and Vilar, 2018; Dinu et al., 2019; Michon et al., 2020).

Autoregressive decoder-only LLMs, such as GPT-3 (Brown et al., 2020; Ouyang et al., 2022), BLOOM (BigScience Workshop et al., 2022), PaLM (Chowdhery et al., 2022), and LLaMA (Touvron et al., 2023) are trained to predict the

next word given the previous context. During unsupervised pre-training, a language model develops a broad set of pattern recognition abilities. It then uses these abilities at inference time to rapidly recognize and adapt to the desired task. In their experiments, Brown et al. (2020) use the term "in-context learning" to describe a scenario where a pre-trained language model at inference time learns to replicate certain input-output text generation patterns without further fine-tuning. They show that autoregressive LLMs such as GPT-3 can perform well on diverse tasks, through zero-shot, one-shot, and few-shot in-context learning without weight updates. Instead of asking the model to directly perform a given task, the input can be augmented with relevant examples, which help the model adapt its output. The key idea of in-context learning is to learn from analogy. The model is expected to learn the pattern hidden in the demonstration and accordingly make better predictions (Dong et al., 2022).

Previous researchers investigated using neural language models for MT through few-shot in-context learning (Vilar et al., 2022) and even in zero-shot settings (Wang et al., 2021). Other researchers proposed using LLMs for generating synthetic domain-specific data for MT domain adaptation (Moslem et al., 2022). Recently, researchers (Agrawal et al., 2022; Zhang et al., 2023) confirmed the importance of in-context example selection for the quality of MT with LLMs.

The main contribution of this paper is investigating the capabilities of LLMs such as GPT-3.5, GPT-4 (including ChatGPT), and BLOOM for real-time adaptive MT through in-context learning. As illustrated by Figure 1, such LLMs can achieve better translation quality through adapting its output to adhere to the terminology and style used in previously approved translation pairs. In particular, we would like to understand the quality with which such models can perform the following tasks, without any further training:

- Adapting new translations to match the terminology and style of previously approved TM fuzzy matches, at inference time;
- Matching or outperforming the quality of translations generated by encoder-decoder MT models across a number of languages;
- Fixing translations from stronger encoder-decoder MT systems using fuzzy matches, which is especially useful for low-resource languages; and
- Terminology-constrained MT, by first defining terminology in the relevant sentences or dataset, and then forcing new translations to use these terms.

2 Experimental Setup

In all our experiments, we use GPT-3.5 *text-davinci-003* model via its official API.[1] For parameters, we use *top-p* 1, with *temperature* 0.3 for the three translation tasks, and 0 for the terminology extraction task.[2] For the maximum length of tokens, we observe that French and Spanish tokens can be 3–4 times the number of English source words, while other languages can be longer. Hence, we roughly choose a length multiplier value, which we set to 8 for Arabic, 5 for Chinese and Kinyarwanda, and 4 for French and Spanish. We used batch requests with a batch size of 20 segments.[3] Our scripts are publicly available.[4]

As we aim to simulate a document-level scenario where translators are required to adhere to a project's or client's TM, we use the domain-specific dataset, TICO-19 (Anastasopoulos et al., 2020), which includes 3070 unique segments. From now on, we will refer to it as the "context dataset". We focus on a range of languages with diverse scripts and amounts of resources, namely English as the source language, and Arabic, Chinese, French, Kinyarwanda, and Spanish as the target languages.

3 Adaptive MT with Fuzzy Matches

In translation environments, similar approved translated segments are usually referred to as "fuzzy matches", and are stored in parallel datasets, known as translation memories (TMs).[5] Researchers have investigated the possibilities of improving MT quality and consistency with fuzzy matches (Knowles et al., 2018; Bulte and Tezcan, 2019; Xu et al., 2020). Incorporating fuzzy matches into the MT process can help the system generate more accurate translations, and try to ensure adherence to pre-approved terminology and preferred style requirements.

In this set of experiments, we investigate the possibility of forcing the translation of a new sentence pair to adapt to fuzzy matches in the context dataset. To extract fuzzy matches, we use embedding similarity-based retrieval. Previous researchers have shown that approaches that depend

[1] `https://openai.com/api/`

[2] To avoid over-generation, the option *stop* can be set to ['\n']. However, if a new line is generated by the model before the translation, this might result in not generating a translation. Alternatively, over-generation can be manually handled.

[3] For higher values of few-shot translation into Arabic using *text-davinci-003*, we had to decrease the batch size to avoid exceeding the tokens-per-minute limit.

[4] `https://github.com/ymoslem/Adaptive-MT-LLM`

[5] Segments stored in a TM can be smaller than a full sentence (e.g. a title) or larger. However, as most segments in a TM are supposed to be sentence pairs, we use the two words interchangeably throughout the paper.

Lang	Context	spBLEU ↑	chrF++ ↑	TER ↓	COMET ↑
EN-AR	zero-shot	27.6	48.36	70.6	41.28
	random 2-shot	28.94	49.35	70.55	43.32
	fuzzy 1-shot	36.38	55.08	63.99	55.1
	fuzzy 2-shot	38.41	56.57	62.31	57.36
	fuzzy 3-shot	39.75	57.52	61.12	59.68
	fuzzy 4-shot	40.84	58.27	60.39	62.16
	fuzzy 5-shot	41.33	58.64	59.95	62.65
	fuzzy 7-shot	**41.81**	**59.1**	**59.38**	**64.01**
EN-ES	zero-shot	53.91	72.61	36.86	84.0
	random 2-shot	54.78	73.12	36.09	85.25
	fuzzy 2-shot	59.64	75.83	32.56	90.37
	fuzzy 5-shot	61.24	76.73	31.32	91.51
	fuzzy 10-shot	**61.77**	**77.05**	**30.9**	**92.0**
EN-FR	zero-shot	44.87	65.29	50.34	58.67
	random 2-shot	45.91	65.4	49.92	57.6
	fuzzy 1-shot	48.39	66.58	48.18	59.49
	fuzzy 2-shot	49.79	67.41	46.79	61.38
	fuzzy 3-shot	50.96	68.06	45.85	61.97
	fuzzy 4-shot	51.89	68.5	44.94	62.7
	fuzzy 5-shot	51.94	68.43	45.09	62.81
	fuzzy 10-shot	**53.72**	**69.39**	**43.82**	**63.57**
EN-RW	zero-shot	2.82	22.53	143.12	N/A
	random 2-shot	3.8	25.19	129.88	N/A
	fuzzy 2-shot	12.23	36.66	105.54	N/A
	fuzzy 5-shot	14.96	39.84	100.11	N/A
	fuzzy 10-shot	**17.87**	**41.44**	**92.84**	N/A
EN-ZH	zero-shot	32.41	40.82	99.45	59.87
	random 2-shot	38.72	44.06	87.56	68.39
	fuzzy 2-shot	46.18	49.12	69.0	73.9
	fuzzy 5-shot	47.94	50.28	64.96	74.86
	fuzzy 10-shot	**49.11**	**51.22**	**63.14**	**75.3**

Table 1: Adaptive MT with fuzzy matches for GPT-3.5 few-shot in-context learning outperforms using random sentence pairs as context examples. Increasing the number of fuzzy matches can improve the translation quality further. The table shows consistent results for EN-AR, EN-ES, EN-FR, EN-RW, and EN-ZH language pairs.

on embeddings to retrieve fuzzy matches can outperform those that use Edit Distance (Hosseini et al., 2020; Pham et al., 2020). To this end, we employ the paraphrase mining module from the Sentence-Transformers library (Reimers and Gurevych, 2019). We use the *all-MiniLM-L6-v2* model because of its high accuracy and efficiency.[6] For each sentence, we retrieve up to top_k other sentences. We experiment with diverse values of 1 to 10 sentence(s) from the context dataset.[7] Table 2 elaborates on the statistics of fuzzy matches based on their similarity to the new source sentence in 2-shot and 5-shot scenarios.[8]

The following illustrations show the difference between zero-shot and few-shot translation prompts. In the zero-shot prompt, only the source sentence and language names are provided, encouraging the model to generate the translation. The few-shot prompt incorporates translation examples to influence the style of the output.

[6] https://www.sbert.net/

[7] For Arabic, we could only integrate up to 7 matches (not 10 matches) because the tokenizer used by GPT-3.5 generates many more tokens for some Unicode languages, which can easily hit the max length of 4097 tokens. We observe that the issue has been alleviated by newer models.

[8] While creating prompts, we arrange fuzzy matches in descending order, making higher matches closer to the segment to be translated. We experimented with reversing the order, and there was no significant difference in terms of translation quality.

Prompt: EN-AR zero-shot translation

English: <source_segment>
Arabic:

Prompt: EN-AR two-shot translation

English: <source_fuzzy_match$_2$>
Arabic: <target_fuzzy_match$_2$>
English: <source_fuzzy_match$_1$>
Arabic: <target_fuzzy_match$_1$>
English: <source_segment>
Arabic:

Results illustrated by Figure 1 show that few-shot translation with GPT-3.5 using fuzzy matches as context outperforms few-shot translation with random examples, although using random sentence pairs outperforms zero-shot translation. As demonstrated by Table 1, across five language pairs, adding more fuzzy matches improves translation quality further. At some point, there might be diminishing returns of adding more similar sentences as their similarity score decreases. In other words, increasing the number of fuzzy matches from 2 sentences to 5 or 10 sentences incrementally improves translation quality, but with smaller quality gains.

Similarity Score	Segment Statistics			
	fuzzy 2-shot		fuzzy 5-shot	
>90%	167	2.7%	168	1.1%
89-80%	751	12.2%	1,103	7.2%
79-70%	1,593	25.9%	3,143	20.5%
69-60%	1,825	29.7%	4,661	30.4%
<60%	1,804	29.4%	6,275	40.9%
Total	6,140 = 3,070*2		15,350 = 3,070*5	

Table 2: Numbers and percentages of segments based on their similarity to the new source segment, in the 2-shot and 5-shot experiments using fuzzy matches for in-context learning. The English source is used to calculate similarity across the 5 language pairs.

4 GPT-3 vs Encoder-Decoder MT Models

In this section, we aim to compare evaluation results we obtained from various MT encoder-decoder Transformer-based systems (Vaswani et al., 2017) with those from GPT-3.5. To this end, we translate our context dataset with a range of open-source and commercial MT models, including DeepL Translate API,[9] Google Cloud Translation API, OPUS (Tiedemann, 2020),[10] and NLLB-200 (NLLB Team et al., 2022). We converted OPUS and NLLB models to the CTranslate2 (Klein et al., 2020) format with int8 quantization for efficiency. Inference parameters include

[9] DeepL supports French, Spanish and Chinese, but not Arabic and Kinyarwanda.

[10] We use OPUS models from the Tatoeba-Challenge, specifically the models augmented with back-translation, and trained with Transformer-Big.

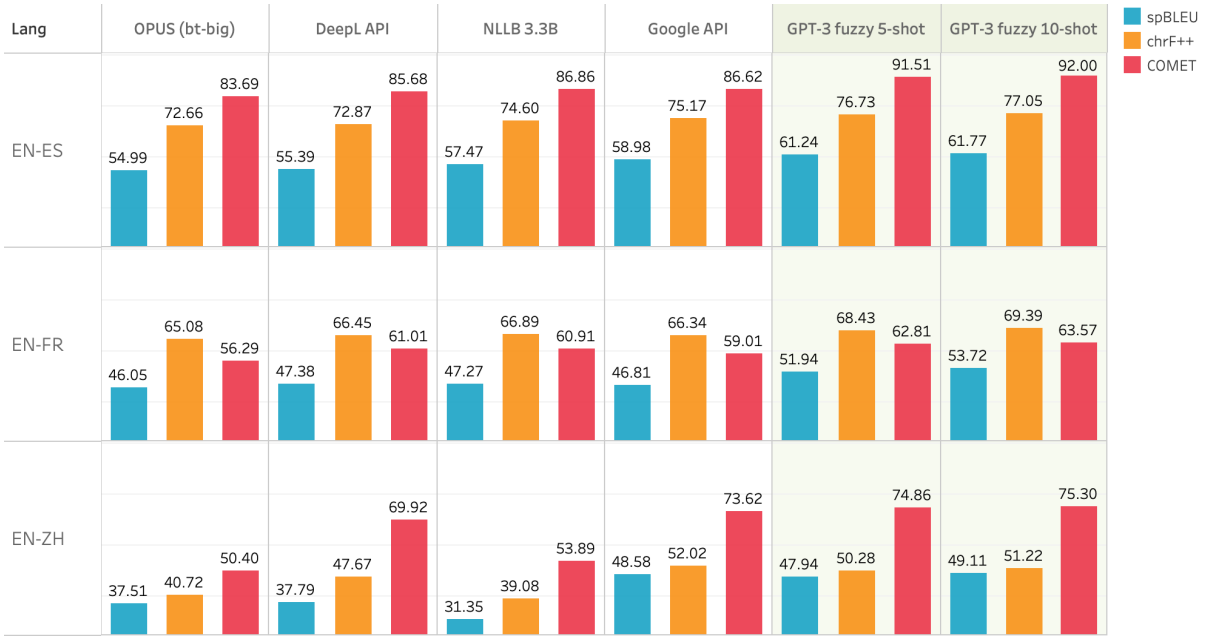

Figure 2: Evaluation results for GPT-3.5 few-shot translation with 5 or 10 fuzzy matches compared to encoder-decoder MT models (DeepL, Google, OPUS, and NLLB). Specifically, for EN-ES, EN-FR, and EN-ZH language pairs, few-shot translation with GPT-3.5 outperforms conventional systems.

beam_size 4 and *max_batch_size 2024*, on a GPU *A100-SXM4-40GB* (Google Colab Pro). For tokenization, we used SentencePiece (Kudo and Richardson, 2018) with the source and target subword models provided for each OPUS model, and the multilingual model provided by NLLB for tokenization.[11]

We observe that for high-resource languages, adaptive MT with fuzzy matches using GPT-3.5 few-shot in-context learning (cf. Section 3) can outperform strong encoder-decoder MT systems. For the English-to-French and English-to-Spanish language pairs, few-shot translation with GPT-3.5 incorporating only 5 fuzzy matches outperforms strong encoder-decoder MT models, as demonstrated by Figure 2. For English-to-Chinese translation, only when we used 10 fuzzy matches could we achieve better results. However, for English-to-Arabic and English-to-Kinyarwanda translations, results were not on par with the other three language pairs. The results are detailed in Table 3.

Among the popular adaptive encoder-decoder MT systems is ModernMT.[12] Originally, the system adopted the instance-based adaptation approach proposed by Farajian et al. (2017). To control our experiments with ModernMT to match those with GPT-3.5 few-shot translation, we created a new TM for each segment to include only the top-10 fuzzy matches for this segment. Table 3 illustrates the evaluation results of ModernMT translation with and without a TM. In general, using a TM with ModernMT improves translation quality. Moreover, we observe that zero-shot translation performance (without a TM) of ModernMT outperforms GPT-3.5 for the 4 supported language pairs. However, except for English-to-Arabic, few-shot translation with GPT-3.5 using either 5 or 10 fuzzy matches outperforms the translation quality of ModernMT using a TM with 10 fuzzy matches per segment, for English-to-Chinese, English-to-French, and English-to-Spanish language pairs.

5 Incorporating Encoder-Decoder MT

As we demonstrated in the previous section, encoder-decoder MT models have achieved high translation quality for several language pairs. Nevertheless, adaptive MT with LLM few-shot in-context learning can surpass such quality, especially for high-resource languages. In this section, we investigate whether we can utilize encoder-decoder MT models to further improve adaptive translation with GPT-3.5. In the next subsections, we study two scenarios:

- appending fuzzy matches with MT from an encoder-decoder model to enhance in-context learning.
- translating the source side of fuzzy matches, and using these MT translations for few-shot in-context learning along with the original translations.

[11] *flores200_sacrebleu_tokenizer_spm.model* is used for both tokenization for NLLB and also for spBLEU (Goyal et al., 2022) in sacreBLEU.

[12] `https://www.modernmt.com/`

5.1 Fuzzy matches + new segment MT

Incorporating a translation from an encoder-decoder MT model with fuzzy matches, we could achieve substantial improvements over the baseline MT performance. As illustrated by Table 5, although OPUS English-to-Arabic translation quality outperforms GPT-3.5 few-shot translation with 5 fuzzy matches, appending these fuzzy matches with OPUS translation outperforms both OPUS translation only and GPT-3.5 translation with fuzzy matches only. Similarly, adding Google English-to-Chinese translation to 5 fuzzy matches outperforms both baselines. Even for the very low-resource English-to-Kinyarwanda language pair, we relatively notice a similar behaviour, using MT outputs of OPUS or NLLB models.

However, we observe that if the translation with only fuzzy matches is significantly better than the encoder-decoder MT baseline, we may not achieve further gains. For example, the GPT-3.5 translations with 5 fuzzy matches are already much better than the OPUS translation for English-to-French or Google translation for English-to-Spanish. That is why incorporating the MT output from OPUS or Google did not enhance the GPT-3.5 translation quality for these language pairs.

5.2 Fuzzy matches + all segments MT

In Section 5.1, we added MT of the new segment from an encoder-decoder model to fuzzy matches, which enhanced GPT-3.5 in-context learning. In this experiment, we include MT for all fuzzy matches and also for the new source segment to be translated. For the English-to-Kinyarwanda and English-to-Spanish language pairs, it is not clear whether including MT for all in-context examples can significantly outperform including MT for only the new source segment to be translated. Again, this depends on the quality of the original MT and requires further investigation.

6 Bilingual Terminology Extraction

Terminology extraction is the task of automatically defining domain-specific terms in a dataset. Extracted terms are naturally used for building glossaries to help translators. Furthermore, it is possible to improve MT performance through finding sentences that include these terms and fine-tuning the system with them (Hu et al., 2019; Haque et al., 2020).

In this set of experiments, we ask GPT-3.5 to extract 5 bilingual terms from each sentence pair in the context dataset. For parameters, we use temperature 0 and top_p 1.

Lang	System	spBLEU ↑	chrF++ ↑	TER ↓	COMET ↑
EN-AR	OPUS (bt-big)	43.11	60.79	57.24	63.64
	NLLB 600M	35.66	54.6	62.07	54.53
	NLLB 1.2B	41.1	58.51	57.15	63.85
	NLLB 3.3B	43.42	60.11	55.58	66.8
	Google API	43.56	61.58	57.79	65.5
	ModernMT (no TM)	47.17	62.82	53.53	66.64
	ModernMT (TM)	**50.33**	**65.19**	**50.19**	**71.0**
	GPT-3 zero-shot	27.6	48.36	70.6	41.28
	GPT-3 fuzzy 5-shot	41.33	58.64	59.95	62.65
	GPT-3 fuzzy 7-shot	41.81	59.1	59.38	64.01
EN-ES	OPUS (bt-big)	54.99	72.66	36.26	83.69
	NLLB 600M	53.31	72.19	37.13	83.09
	NLLB 1.2B	56.1	73.85	34.96	85.91
	NLLB 3.3B	57.47	74.6	33.99	86.86
	DeepL API	55.39	72.87	36.21	85.68
	Google API	58.98	75.17	32.46	86.62
	ModernMT (no TM)	57.09	74.2	34.27	85.53
	ModernMT (TM)	59.22	75.4	32.79	86.99
	GPT-3 zero-shot	53.91	72.61	36.86	84.0
	GPT-3 fuzzy 5-shot	61.24	76.73	31.32	91.51
	GPT-3 fuzzy 10-shot	**61.77**	**77.05**	**30.9**	**92.0**
EN-FR	OPUS (bt-big)	46.05	65.08	49.8	56.29
	NLLB 600M	43.25	64.17	51.28	56.16
	NLLB 1.2B	46.3	66.25	48.68	59.76
	NLLB 3.3B	47.27	66.89	48.19	60.91
	DeepL API	47.38	66.45	48.47	61.01
	Google API	46.81	66.34	47.01	59.01
	ModernMT (no TM)	47.17	66.28	47.91	58.46
	ModernMT (TM)	49.24	67.41	46.17	59.84
	GPT-3 zero-shot	44.87	65.29	50.34	58.67
	GPT-3 fuzzy 5-shot	51.94	68.43	45.09	62.81
	GPT-3 fuzzy 10-shot	**53.72**	**69.39**	**43.82**	**63.57**
EN-RW	OPUS (Tatoeba 2021)	1.38	15.32	153.58	N/A
	OPUS (2020)	5.58	27.05	101.25	N/A
	NLLB 600M	19.46	47.61	80.01	N/A
	NLLB 1.2B	23.6	50.73	74.53	N/A
	NLLB 3.3B	**25.17**	**52.59**	**73.06**	N/A
	Google API	20.63	48.37	73.54	N/A
	GPT-3 zero-shot	2.82	22.53	143.12	N/A
	GPT-3 fuzzy 5-shot	14.96	39.84	100.11	N/A
	GPT-3 fuzzy 10-shot	17.87	41.44	92.84	N/A
EN-ZH	OPUS (bt-big)	37.51	40.72	121.49	50.4
	NLLB 600M	24.9	33.87	109.37	39.28
	NLLB 1.2B	29.02	37.45	110.22	50.05
	NLLB 3.3B	31.35	39.08	109.52	53.89
	DeepL API	37.79	47.67	100.83	69.92
	Google API	48.58	**52.02**	70.87	73.62
	ModernMT (no TM)	37.61	48.46	102.18	67.45
	ModernMT (TM)	39.85	50.95	101.53	69.64
	GPT-3 zero-shot	32.41	40.82	99.45	59.87
	GPT-3 fuzzy 5-shot	47.94	50.28	64.96	74.86
	GPT-3 fuzzy 10-shot	**49.11**	51.22	**63.14**	**75.3**

Table 3: Comparing GPT-3.5 few-shot translation using fuzzy matches with encoder-decoder MT systems, DeepL Translate API, Google Cloud Translation API, OPUS (Tatoeba-Challenge, with back-translation and Transformer-Big), and NLLB-200 (600M, 1.2B & 3.3B parameters).

Lang	Sentences	Terms	Correct	%
EN-AR	500	2,500	2,427	97.08
EN-ES	500	2,500	2,397	95.88
EN-FR	500	2,500	2,382	95.28

Table 4: Human evaluation results for the terminology extraction task for English-to-Arabic (EN-AR), English-to-Spanish (EN-ES), and English-to-French (EN-FR) language pairs. The majority of the terms that GPT-3 extracted ($> 95\%$) were accurate.

Human evaluation was performed for Arabic, French,[13] and Spanish. We provided the evaluators with a random sample of 500 sentences and their extracted terms. They were asked to use a 0-1 scale

[13]We observe that the original English-to-French TICO-19 dataset includes several misaligned translation pairs. This can negatively affect the quality of tasks using such sentences. That is why it is important to filter parallel datasets to remove possible misalignments. The evaluation sample has been manually refined to include only well-aligned translation pairs. Automatic semantic filtering approaches can be applied to large datasets.

to determine whether each source and target term were equivalent, and whether the extracted terms were actually in the sentence pair (relevant inflexions are acceptable). In several cases where the evaluators marked the extracted term pair with 0, the model had made up either the source, target, or both; although it might be correct, it was not in the provided sentence pair. In other cases, the extracted term was partial, sometimes due to reaching the maximum length of tokens. Nevertheless, as Table 4 illustrates, the majority of the terms in the provided sample were accurately extracted by the model.

7 Terminology-Constrained MT

As observed in Section 3, adding more fuzzy matches enhances in-context learning and hence improves translation quality. However, early in a real-world translation project, we might not have so many fuzzy matches. By incorporating domain-specific terminology, the system can produce translations that are more accurate and consistent with the terminology used in that field. In this section, we investigate integrating terms in the process when there are N fuzzy matches. For example, if we have only two fuzzy matches, we either extract terms from these similar sentences or from a glossary, and use those that match up to 5-gram phrases in the source sentence to be translated. In this work, we use the terminology extraction process elaborated in Section 6. Obviously, if a pre-approved glossary is available, it can be used instead. We investigate three scenarios:

- Few-shot translation with 2 fuzzy matches and their terms. As we do not have terms for the segment to be translated, we use terms from the 2 fuzzy matches if they are found in a set of n-grams (1-5) of the source segment to be translated. Integrating terms into two-shot prediction, i.e. using both terms and two fuzzy matches for in-context learning, outperforms using fuzzy matches only.
- We automatically compile a glossary including all terms from the dataset, with 2+ frequency, and up to 5-grams. If there are multiple targets for the same source, the term pair with the highest frequency is selected. Stop words and terms with empty source or target sides are excluded. The list is sorted by n-gram length, so terms with longer n-grams are prioritized. As illustrated by Table 6, integrating terms from a glossary outperforms adding terms from only two fuzzy matches, most likely due to the diversity that this option offers. In prompts (cf. Appendix A), we use terms found in a set of n-grams (1-5) of the source segment to be translated. We experiment with adding maximum 5 terms and maximum 10 terms, which does not show a huge difference in performance; in some cases only a smaller number of terms is available in the glossary.
- Zero-shot translation, i.e. without any fuzzy matches. This is similar to the previous scenario, except that we only use terms from the glossary. In zero-shot prediction, adding terms from the glossary improves translation quality. As shown in Table 6, improvements are significant across all 5 language pairs.

We conducted human evaluation for English-to-Arabic, English-to-French, and English-to-Spanish terminology-constrained MT, to see to what extent the model adheres to the required terms, and how this affects the overall translation quality. The evaluators are professional linguists in the respective languages. We provided the evaluators with 4 sets of 100 randomly selected sentence pairs (zero-shot, zero-shot with glossary terms, fuzzy two-shot, and fuzzy two-shot with glossary terms). They were asked to evaluate the sentence-level translation quality on a 1-4 scale (Coughlin, 2003) and the usage of each provided term in the translation on a 0-1 scale, as elaborated by Table 7.

Lang	GPT-3 Context	Human Eval. ↑	Terms ↑
EN-AR	Zero-shot	2.80	0.67
	Zero-shot + glossary terms	**3.19**	**0.94**
	Fuzzy two-shot	2.89	0.80
	Fuzzy two-shot + glossary terms	**3.03**	**0.94**
EN-ES	Zero-shot	3.76	0.87
	Zero-shot + glossary terms	**3.93**	**0.96**
	Fuzzy two-shot	3.77	0.89
	Fuzzy two-shot + glossary terms	**3.84**	**0.97**
EN-FR	Zero-shot	3.55	0.89
	Zero-shot + glossary terms	**3.64**	**0.97**
	Fuzzy two-shot	3.50	0.91
	Fuzzy two-shot + glossary terms	**3.55**	**0.92**

Table 7: Human evaluation of terminology-constrained MT, for EN-AR, EN-ES, and EN-FR. The results cover zero-shot and two-shot translation without and with (maximum 5) glossary terms. The column "Human Eval." refers to the average evaluation score on a 1-4 scale. The column "Terms" refers to the average number of terms that the model has successfully transferred into the translation on a 0-1 scale.

According to the evaluators, for Arabic, French and Spanish, terminology-constrained MT successfully transferred the provided glossary terms into the target more often than zero-shot and few-shot translation without terminology incorporation. In several cases, forcing glossary terms to be used could help improve the overall translation quality; however, sometimes it was detrimental to grammatical accuracy. Although we provided the model with longer terms before shorter ones, contradictory terms can hurt translation quality.

Lang	System	spBLEU ↑	chrF++ ↑	TER ↓	COMET ↑
EN-AR	MT (OPUS)	43.11	60.79	57.24	63.64
	GPT-3 fuzzy 5-shot	41.33	58.64	59.95	62.65
	GPT-3 fuzzy 5-shot + 1-MT	**45.9**	**62.9**	**55.14**	**67.74**
EN-ES	MT (Google)	58.98	75.17	32.46	86.62
	GPT-3 fuzzy 2-shot	59.64	75.83	32.56	90.37
	GPT-3 fuzzy 2-shot + 1-MT	59.82	75.73	**32.16**	89.0
	GPT-3 fuzzy 2-shot + all-MT	**60.2**	**76.06**	32.32	**92.0**
	GPT-3 fuzzy 5-shot	**61.24**	**76.73**	**31.32**	91.51
	GPT-3 fuzzy 5-shot + 1-MT	60.49	76.16	31.49	89.55
	GPT-3 fuzzy 5-shot + all-MT	61.1	76.52	31.8	**92.07**
EN-FR	MT (OPUS)	46.05	65.08	49.8	56.29
	GPT-3 fuzzy 5-shot	**51.94**	**68.43**	**45.09**	**62.81**
	GPT-3 fuzzy 5-shot + 1-MT	47.95	66.72	48.34	59.69
EN-RW	MT #1 (Google)	20.63	48.37	73.54	N/A
	GPT-3 fuzzy 5-shot	14.96	39.84	100.11	N/A
	GPT-3 fuzzy 5-shot + 1-MT #1	22.51	**49.69**	**72.97**	N/A
	GPT-3 fuzzy 5-shot + all-MT #1	**25.01**	49.43	74.75	N/A
	MT #2 (NLLB 3.3B)	25.17	52.59	73.06	N/A
	GPT-3 fuzzy 5-shot + 1-MT #2	25.59	53.12	**72.73**	N/A
	GPT-3 fuzzy 5-shot + all-MT #2	**27.52**	**53.23**	73.79	N/A
EN-ZH	MT (Google)	48.58	52.02	70.87	73.62
	GPT-3 fuzzy 5-shot	47.94	50.28	**64.96**	**74.86**
	GPT-3 fuzzy 5-shot + 1-MT	**49.45**	**52.4**	67.81	74.61

Table 5: Combining fuzzy matches with high-quality MT from encoder-decoder systems can improve translation quality with GPT-3.5 few-shot in-context learning, especially for low-resource and medium-resource languages. 1-MT refers to appending fuzzy matches with the MT of the segment to be translated, while all-MT refers to additionally adding MT for each segment of the fuzzy matches along with its approved translation. For EN-AR and EN-RW improvements are clearer than for EN-ES, EN-FR and EN-ZH, potentially due to the limited support of EN-AR and EN-RW by GPT-3.5, which made them benefit more from incorporating MT from stronger encoder-decoder models.

Hence, it might be better to exclude shorter terms if they overlap with longer ones.[14] In production workflows, linguists can be provided with translation alternatives with and without fuzzy matches and/or terminology to be able to use the best translation. Alternatively, automatic quality estimation can be conducted to select the best translation.

Among interesting observations that human evaluation reveals is that in few-shot translation with fuzzy matches (even *without* terms), the number of successfully used terms is more than those in zero-shot translation. This can help enhance consistency with approved translations. Moreover, incorporating glossary terms in a zero-shot prompt can result in quality gains comparable to those of few-shot translation with fuzzy matches.

8 ChatGPT

At the time of writing this paper, OpenAI has released new conversational models, publicly referred to as ChatGPT. This range of models includes: GPT-3.5 Turbo and GPT-4. In this section, we briefly investigate the translation capabilities of these models compared to GPT-3.5 Davinci. Generally, we observe that both of the new models solve some tokenization issues, especially for non-Latin languages such as Arabic. While *gpt-3.5-turbo* is more efficient than *text-davinci-003*, it shows comparable quality for both zero-shot and few-shot translation (with fuzzy matches). The newest model *gpt-4* provides better zero-shot translation quality, while the quality of few-shot translation is relatively similar to that of the two other models. Table 8 demonstrates the results.

[14]For example, "New York Times" can be transferred without translation into the target, while "New York" might be translated. If the model is provided with both terms while it is actually supposed to use the former, this can cause confusion.

Lang	Model	Context	spBLEU ↑	chrF++ ↑	TER ↓	COMET ↑
EN-AR	GPT-3.5 Davinci	0-shot	27.6	48.36	70.6	41.28
	GPT-3.5 Turbo		38.06	56.35	61.34	62.68
	GPT-4		**40.29**	**57.86**	**59.55**	**64.25**
	GPT-3.5 Davinci	2-shot	38.41	56.57	62.31	57.36
	GPT-3.5 Turbo		46.04	62.18	55.03	73.35
	GPT-4		**47.52**	**63.28**	**53.04**	**73.7**
EN-ES	GPT-3.5 Davinci	0-shot	53.91	72.61	36.86	84.0
	GPT-3.5 Turbo		52.91	70.87	38.86	82.28
	GPT-4		**56.93**	**74.41**	**34.35**	**87.89**
	GPT-3.5 Davinci	2-shot	59.64	75.83	32.56	90.37
	GPT-3.5 Turbo		**60.35**	**76.51**	32.05	91.57
	GPT-4		60.16	**76.51**	**31.77**	**91.86**
EN-FR	GPT-3.5 Davinci	0-shot	44.87	65.29	50.34	58.67
	GPT-3.5 Turbo		46.85	66.75	48.31	61.34
	GPT-4		**47.39**	**67.14**	**48.03**	**61.93**
	GPT-3.5 Davinci	2-shot	49.79	67.41	46.79	61.38
	GPT-3.5 Turbo		**49.88**	68.33	46.27	63.62
	GPT-4		49.75	**68.38**	**45.97**	**64.04**
EN-RW	GPT-3.5 Davinci	0-shot	2.82	22.53	143.12	N/A
	GPT-3.5 Turbo		5.31	29.77	114.34	N/A
	GPT-4		**8.95**	**35.28**	**93.15**	N/A
	GPT-3.5 Davinci	2-shot	12.23	36.66	105.54	N/A
	GPT-3.5 Turbo		12.49	39.37	105.51	N/A
	GPT-4		**16.78**	**44.21**	**83.31**	N/A
EN-ZH	GPT-3.5 Davinci	0-shot	32.41	40.82	99.45	59.87
	GPT-3.5 Turbo		36.83	45.77	99.83	69.13
	GPT-4		**37.65**	**47.02**	**99.37**	**70.75**
	GPT-3.5 Davinci	2-shot	46.18	49.12	**69.0**	73.9
	GPT-3.5 Turbo		**45.95**	49.79	74.53	74.63
	GPT-4		45.37	**50.26**	79.29	**74.9**

Table 8: Comparing GPT-3.5 *text-davinci-003* to ChatGPT models *gpt-3.5-turbo* and *gpt-4* for zero-shot and few-shot translation with 2 fuzzy matches

9 BLOOM and BLOOMZ

In this section, we compare GPT-3.5 to open-source multilingual models, namely BLOOM (BigScience Workshop et al., 2022) and BLOOMZ (Muennighoff et al., 2022). While BLOOM is

Lang	GPT-3.5 Context	spBLEU ↑	chrF++ ↑	TER ↓	COMET ↑
EN-AR	zero-shot	27.6	48.36	70.6	41.28
	zero-shot + max 5 terms (glossary)	**35.38**	**54.53**	**65.36**	**54.91**
	fuzzy 2-shot	38.41	56.57	62.31	57.36
	fuzzy 2-shot + terms (fuzzy)	39.38	57.22	62.01	59.36
	fuzzy 2-shot + max 5 terms (glossary)	41.27	58.84	60.09	62.17
	fuzzy 2-shot + max 10 terms (glossary)	**41.95**	**59.34**	**59.45**	**62.48**
EN-ES	zero-shot	53.91	72.61	36.86	84.0
	zero-shot + max 5 terms (glossary)	**55.99**	**74.18**	**35.3**	**87.21**
	fuzzy 2-shot	59.64	75.83	32.56	90.37
	fuzzy 2-shot + terms (fuzzy)	59.66	75.91	32.53	90.04
	fuzzy 2-shot + max 5 terms (glossary)	60.5	76.55	**31.93**	**91.05**
	fuzzy 2-shot + max 10 terms (glossary)	**60.54**	**76.58**	32.02	**91.05**
EN-FR	zero-shot	44.87	65.29	50.34	58.67
	zero-shot + max 5 terms (glossary)	**45.94**	**66.01**	**49.22**	**59.78**
	fuzzy 2-shot	49.79	67.41	46.79	61.38
	fuzzy 2-shot + terms (fuzzy)	**50.58**	**67.93**	45.81	62.04
	fuzzy 2-shot + max 3 terms (glossary)	50.46	67.69	46.22	**68.94**
	fuzzy 2-shot + max 5 terms (glossary)	50.55	67.78	**46.19**	60.24
	fuzzy 2-shot + max 10 terms (glossary)	49.64	66.86	47.34	58.57
EN-RW	zero-shot	2.82	22.53	143.12	N/A
	zero-shot + max 5 terms (glossary)	**7.26**	**30.83**	**115.44**	N/A
	fuzzy 2-shot	12.23	36.66	105.54	N/A
	fuzzy 2-shot + terms (fuzzy)	12.43	36.48	102.22	N/A
	fuzzy 2-shot + max 5 terms (glossary)	15.34	39.96	96.09	N/A
	fuzzy 2-shot + max 10 terms (glossary)	**15.49**	**40.53**	**96.0**	N/A
EN-ZH	zero-shot	32.41	40.82	99.45	59.87
	zero-shot + max 5 terms (glossary)	36.31	44.72	96.45	68.6
	zero-shot + max 10 terms (glossary)	**36.64**	**45.06**	**96.24**	**68.94**
	fuzzy 2-shot	46.18	49.12	69.0	**73.9**
	fuzzy 2-shot + terms (fuzzy)	46.16	49.11	**68.79**	73.41
	fuzzy 2-shot + max 5 terms (glossary)	**46.6**	**49.51**	69.46	73.88
	fuzzy 2-shot + max 10 terms (glossary)	46.31	49.25	69.39	73.57

Table 6: Terminology-constrained MT with GPT 3.5 outperforms both zero-shot and 2-shot translation with fuzzy matches, although gains are much higher for zero-shot translation. For zero-shot translation, we experimented with adding terms from a glossary. For 2-shot translation with fuzzy matches, we compared adding terms from these 2 fuzzy matches to adding terms from a glossary. The latter revealed better results.

a general-purpose LLM, BLOOMZ belongs to a family of models capable of following human instructions in a zero-shot manner.

We use BLOOM and BLOOMZ via the Hugging Face's Inference API.[15] As mentioned in Section 2, recommended (sampling) parameters for translation with GPT-3.5 are top-p 1 and temperature up to 0.3. For BLOOM, the same parameters are not good for translation.[16] We found that "greedy search" achieves better results for BLOOM, which are reported in Table 9. We use a batch size of 1, and set the max_new_tokens parameter to be double the number of words of the source sentence if it is less than 250, the maximum number of new tokens allowed by BLOOM's API; otherwise, we set it to 250 tokens. For comparison purposes, we use the same values for BLOOMZ.[17]

When providing each system with two fuzzy matches, generally GPT-3.5 outperforms both BLOOM and BLOOMZ for most language pairs, except English-to-Arabic translation. The English-to-French translation quality of BLOOM and GPT-3.5 is comparable.

[15] https://huggingface.co/inference-api

[16] Using lower sampling values of top-p and temperature such as 0.9 and 0.1, respectively, can generate good outputs. However, greedy search shows better translation performance.

[17] BLOOMZ is trained to generate the required output only; however, using BLOOM, we had to truncate over-generated text outputs, excluding anything generated in a new line.

Lang	System	spBLEU ↑	chrF++ ↑	TER ↓	COMET ↑
EN-AR	BLOOM fuzzy 2-shot	**43.19**	**59.48**	**57.58**	**67.36**
	BLOOMZ fuzzy 2-shot	36.29	53.33	66.86	58.4
	GPT-3 fuzzy 2-shot	38.41	56.57	62.31	57.36
EN-ES	BLOOM fuzzy 2-shot	57.67	74.25	34.86	86.48
	BLOOMZ fuzzy 2-shot	53.07	70.44	40.45	81.38
	GPT-3 fuzzy 2-shot	**59.64**	**75.83**	**32.56**	**90.37**
EN-FR	BLOOM fuzzy 2-shot	**50.52**	66.81	**46.45**	55.74
	BLOOMZ fuzzy 2-shot	45.1	62.73	51.69	47.49
	GPT-3 fuzzy 2-shot	49.79	**67.41**	46.79	**61.38**
EN-RW	BLOOM fuzzy 2-shot	10.95	31.87	91.07	N/A
	BLOOMZ fuzzy 2-shot	**12.26**	35.44	**88.36**	N/A
	GPT-3 fuzzy 2-shot	12.23	**36.66**	105.54	N/A
EN-ZH	BLOOM fuzzy 2-shot	40.62	40.62	75.24	66.23
	BLOOMZ fuzzy 2-shot	34.82	38.23	80.03	59.92
	GPT-3 fuzzy 2-shot	**46.18**	**49.12**	**69.0**	**73.9**

Table 9: Comparing GPT-3.5 to BLOOM and BLOOMZ for few-shot translation with 2 fuzzy matches

10 Conclusion

In this work, we conducted several experiments to assess the performance of GPT-3.5 across multiple translation tasks, namely adaptive MT using fuzzy matches (cf. Section 3), MT post-editing (cf. Section 5), terminology extraction (cf. Section 6), and terminology-constrained MT (cf. Section 7). Moreover, we compared its translation quality with strong encoder-decoder MT systems. Generally speaking, results obtained from these experiments are very promising. While some high-resource languages such as English-to-French, English-to-Spanish and even English-to-Chinese show excellent results, other languages have lower support

either because they are low-resource languages such as English-to-Kinyarwanda or because of issues in the GPT-3.5 tokenizer such as English-to-Arabic. Nevertheless, when we used GPT-3.5 for MT post-editing of the English-to-Arabic translation obtained from OPUS, the quality significantly surpassed that obtained from both OPUS and Google Translation API. This means that different pipelines can be adopted in production for different language pairs, based on the level of support of these languages by an LLM.

Furthermore, we briefly compared GPT-3.5 translation quality with open-source LLMs such as BLOOM and BLOOMZ. In the future, we would like to expand our experiments with open-source LLMs to cover more aspects.

For adaptive MT with fuzzy matches, it would be interesting to investigate *dynamic* few-shot example selection. For instance, instead of selecting 5 fuzzy matches for all sentences, only high-quality fuzzy matches up to a certain similarity score are used. Similarly, when incorporating glossary terms or MT outputs from other systems, only those with certain quality characteristics are utilized. This can potentially enhance performance gains.

For terminology extraction, we would like to try "phrases" instead of "terms". This would generate longer strings. We would like to see the effect of using such longer phrases, especially for low-resource languages.

This work mainly aims at understanding the quality and level of support that LLMs can achieve (out of the box) for a range of translation tasks across diverse language pairs. In the future, we might consider starting with fine-tuning the model, and then conducting similar experiments. This can be especially beneficial for low-resource languages and rare domains, and can help enhance quality and efficiency.

Acknowledgements

This work is supported by the Science Foundation Ireland (SFI) Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224, the ADAPT Centre for Digital Content Technology under SFI's Grant No. 13/RC/2106_P2, and Microsoft Research.

We would like to extend our sincere thanks to Julie Locquet, Senior Linguist; Philippe Locquet, Senior Linguist and Academic Program Manager at Wordfast; and Dr Muhammed Yaman Muhaisen, Ophthalmologist and Linguist, for conducting the evaluation of our translation tasks.

References

Agrawal, Sweta, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022. In-context Examples Selection for Machine Translation. *arXiv [cs.CL]*, December.

Anastasopoulos, Antonios, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, et al. 2020. TICO-19: the Translation Initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online, December.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv [cs.CL]*, November.

Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 1877–1901.

Bulte, Bram and Arda Tezcan. 2019. Neural Fuzzy Repair: Integrating Fuzzy Matches into Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy, July.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, et al. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv [cs.CL]*, April.

Coughlin, Deborah. 2003. Correlating automated and human assessments of machine translation quality. In *Proceedings of Machine Translation Summit IX: Papers*, New Orleans, USA.

Dinu, Georgiana, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training Neural Machine Translation to Apply Terminology Constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy, July.

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2022. A Survey on In-context Learning. *arXiv [cs.CL]*, December.

Etchegoyhen, Thierry, David Ponce, Harritxu Gete, and Victor Ruiz. 2021. Online Learning over Time in Adaptive Neural Machine Translation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 411–420, Held Online, September.

Farajian, M Amin, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-Domain Neural Machine Translation through Unsupervised Adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark, September.

Goyal, Naman, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguist.*, 10:522–538, May.

Haque, Rejwanul, Yasmin Moslem, and Andy Way. 2020. Terminology-Aware Sentence Mining for NMT Domain Adaptation: ADAPT's Submission to the Adap-MT 2020 English-to-Hindi AI Translation Shared Task. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON): Adap-MT 2020 Shared Task*, pages 17–23, Patna, India, December.

Hokamp, Chris and Qun Liu. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada, July.

Hosseini, Kasra, Federico Nanni, and Mariona Coll Ardanuy. 2020. DeezyMatch: A Flexible Deep Learning Approach to Fuzzy String Matching. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 62–69, Online, October.

Hu, Junjie, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain Adaptation of Neural Machine Translation by Lexicon Induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy, July.

Klein, Guillaume, Dakun Zhang, Clément Chouteau, Josep Crego, and Jean Senellart. 2020. Efficient and high-quality neural machine translation with OpenNMT. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 211–217, Stroudsburg, PA, USA, July.

Knowles, Rebecca, John Ortega, and Philipp Koehn. 2018. A Comparison of Machine Translation Paradigms for Use in Black-Box Fuzzy-Match Repair. In *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 249–255, Boston, MA, March.

Kudo, Taku 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, Brussels, Belgium, November.

Michon, Elise, Josep Crego, and Jean Senellart. 2020. Integrating Domain Terminology into Neural Machine Translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3925–3937, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Moslem, Yasmin, Rejwanul Haque, John Kelleher, and Andy Way. 2022. Domain-Specific Text Generation for Machine Translation. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 14–30, Orlando, USA, September.

Muennighoff, Niklas, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, et al. 2022. Crosslingual Generalization through Multitask Finetuning. *arXiv [cs.CL]*, November.

NLLB Team, Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, et al. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. *arXiv [cs.CL]*, July.

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, et al. 2022. Training language models to follow instructions with human feedback. *arXiv [cs.CL]*, March.

Peris, Álvaro and Francisco Casacuberta. 2019. Online learning for effort reduction in interactive neural machine translation. *Comput. Speech Lang.*, 58:98–126, November.

Pham, Minh Quang, Jitao Xu, Josep Crego, François Yvon, and Jean Senellart. 2020. Priming Neural Machine Translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 516–527, Online, November.

Post, Matt and David Vilar. 2018. Fast Lexically Constrained Decoding with Dynamic Beam Allocation for 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 1314–1324, New Orleans, Louisiana, June.

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November.

Tiedemann, Jörg. 2020. The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online, November.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv [cs.CL]*, February.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, volume 30.

Vilar, David, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting PaLM for Translation: Assessing Strategies and Performance. *arXiv [cs.CL]*, November.

Wang, Shuo, Zhaopeng Tu, Zhixing Tan, Wenxuan Wang, Maosong Sun, and Yang Liu. 2021. Language Models are Good Translators. *ArXiv*.

Wuebker, Joern, Patrick Simianer, and John DeNero. 2018. Compact Personalized Models for Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 881–886, Brussels, Belgium.

Xu, Jitao, Josep Crego, and Jean Senellart. 2020. Boosting Neural Machine Translation with Similar Translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online, July.

Zhang, Biao, Barry Haddow, and Alexandra Birch. 2023. Prompting Large Language Model for Machine Translation: A Case Study. *arXiv [cs.CL]*, January.

A Prompts

This appendix provides examples of the prompts we used for our experiments.

A.1 Zero-shot Translation

Prompt: EN-AR zero-shot translation

English: <source_segment>
Arabic:

A.2 Adaptive MT with Fuzzy Matches

Prompt: EN-AR two-shot translation

English: <source_fuzzy_match$_2$>
Arabic: <target_fuzzy_match$_2$>
English: <source_fuzzy_match$_1$>
Arabic: <target_fuzzy_match$_1$>
English: <source_segment>
Arabic:

A.3 MT Post-editing

Prompt: EN-ZH two-shot + 1-MT

English: <source_fuzzy_match$_2$>
Chinese: <target_fuzzy_match$_2$>
English: <source_fuzzy_match$_1$>
Chinese: <target_fuzzy_match$_1$>
English: <source_segment>
MT: <mt_segment>
Chinese:

Prompt: EN-ZH two-shot + all-MT

English: <source_fuzzy_match$_2$>
MT: <mt_fuzzy_match$_2$>
Chinese: <target_fuzzy_match$_2$>
English: <source_fuzzy_match$_1$>
MT: <mt_fuzzy_match$_1$>
Chinese: <target_fuzzy_match$_1$>
English: <source_segment>
MT: <mt_segment>
Chinese:

A.4 Terminology Extraction

Prompt: terminology extraction

<source_lang>: <source_sentence>
<target_lang>: <target_sentence>

Extract <number> terms from the above sentence pair. Type each <source_lang> term and its <target_lang> equivalent in one line, separated by '<separator>'.

1.

A.5 Terminology-constrained MT

Prompt: EN-ES zero-shot + glossary terms

Terms: <src_term$_1$> = <tgt_term$_1$> - <src_term$_2$> = <tgt_term$_2$> ... <src_term$_5$> = <tgt_term$_5$>
English: <source_segment>
Spanish:

Prompt: EN-ES two-shot + fuzzy terms

Terms: <terms_fuzzy_match$_2$>
English: <source_fuzzy_match$_2$>
Spanish: <target_fuzzy_match$_2$>
Terms: <terms_fuzzy_match$_1$>
English: <source_fuzzy_match$_1$>
Spanish: <target_fuzzy_match$_1$>
Terms: <terms_from_fuzzy_matches$_{1+2}$>
English: <source_segment>
Spanish:

Prompt: EN-ES two-shot + glossary terms

Terms: <terms_fuzzy_match$_2$>
English: <source_fuzzy_match$_2$>
Spanish: <target_fuzzy_match$_2$>
Terms: <terms_fuzzy_match$_1$>
English: <source_fuzzy_match$_1$>
Spanish: <target_fuzzy_match$_1$>
Terms: <terms_from_glossary>
English: <source_segment>
Spanish: