Exploring the Potential of Large Language Models in Adaptive Machine Translation for Generic Text and Subtitles

Abdelhadi Soudi[1], **Mohamed Hannani**[2], **Kristof Van Laerhoven**[2], **Eleftherios Avramidis**[3]
[1] Ecole Nationale Supérieure des Mines de Rabat, Morocco
asoudi@enim.ac.ma
[2]University of Siegen, Germany
mohamed_hannani@yahoo.com, kvl@eti.uni-siegen.de
[3] German Research Center for Artificial Intelligence, Germany
eleftherios.avramidis@dfki.de

Abstract

This paper investigates the potential of contextual learning for adaptive real-time machine translation (MT) using Large Language Models (LLMs) in the context of subtitles and generic text with fuzzy matches. By using a strategy based on prompt composition and dynamic retrieval of fuzzy matches, we achieved improvements in the translation quality compared to previous work. Unlike static selection, which may not adequately meet all request sentences, our enhanced methodology allows for dynamic adaptation based on user input. It was also shown that LLMs and Encoder-Decoder models achieve better results with generic texts than with subtitles for the language pairs English-to-Arabic (En→Ar) and English-to-French (En→Fr). Experiments on datasets with different sizes for En→Ar subtitles indicate that the bigger is not really the better. Our experiments on subtitles support results from previous work on generic text that LLMs are capable of adapting to In-Context learning with few-shot, outperforming Encoder-Decoder MT models and that the combination of LLMs and Encoder-Decoder models improves the quality of the translation.

Keywords: Large Language Models, Adaptive MT, Prompt Composition, LangChain, Generic Text, Subtitles.

1. Introduction

While Large Language Models (LLMs), such as GPT, Llama 2, and Falcon (Penedo et al., 2023) have made progress in tackling a variety of language-related tasks, MT is not a simple sequence-to-sequence task. It involves the complicated task of preserving the subtleties, idiomatic expressions, and distinctive stylistic features that characterize human languages.

LLMs, including but not limited to GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), Falcon (Penedo et al., 2023), and LLaMA (Touvron et al., 2023), have been designed to predict the subsequent word in a sequence based on the context. Brown et al. (2020) and Ouyang et al. (2022) introduced the concept of "In-Context learning" to describe a scenario where a pre-trained language model, during inference, assimilates specific input-output text generation patterns without the need for further fine-tuning. Their research highlighted that autoregressive LLMs, such as GPT-3, exhibit strong performance across diverse tasks, including zero-shot, one-shot, and few-shot In-Context learning without requiring updates to their weights. Instead of directly instructing the model to perform a particular task, input data can be enriched with relevant examples to facilitate the model's adaptation. The core principle of In-Context learning revolves around learning from analogies embedded within demonstrations (Dong et al., 2022).

A key advantage of adaptive MT, a paradigm aimed at enhancing translation by tailoring it to specific domains, genres, or styles, is its ability to achieve domain-specific translation goals without the resource-intensive processes of model training and fine-tuning.

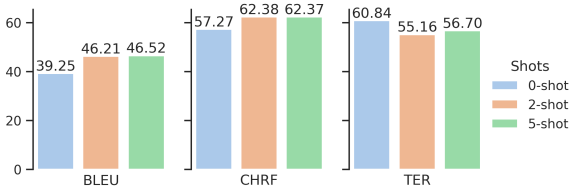

Figure 1: Evaluation results of ChatGPT 3.5 Turbo on TICO 19 for En→Ar language pair, with zero-shot, 2-shot and 5-shot fuzzy matches.

The results in Figure 1 show the performance of GPT-3.5 Turbo with zero-shot, 2-shot, and 5-shot fuzzy matches translation. When employing fuzzy matches, translation quality metrics such as BLEU and TER show substantial improvements, underlining the effectiveness of this technique in enhancing translation accuracy and fluency.

In this work, our particular emphasis lies in harnessing the capabilities of GPT-3.5 Turbo by OpenAI with In-Context examples. We examine the subtleties of adapting machine translation to domain-

BUCC 2024: The 17th Workshop on Building and Using Comparable Corpora, pages 51–58

specific requirements, using the TICO-19 dataset (Generic Text) and TED Talks 2013 dataset (Subtitles). By using our strategy based on prompt composition and dynamic retrieval of fuzzy matches, we report on experimental results for the language pairs English-to-Arabic (En→Ar) and English-to-French (En→Fr). To evaluate the effect of the dataset size on the translation quality of En→Ar generic text and En→Ar subtitles, we conduct experiments on different sizes for the same dataset type. An evaluation of the performance of LLMs and DeepL (Enoder-Decoder model) is also provided.

In the following sections, we provide an overview of the related work (section 2), the methodology (section 3), the experimental setup (section 4), and the results (section 5).

2. Related Work

Prior studies have focused on the application of neural language models in MT, encompassing zero-shot (Wang et al., 2021) and few-shot (Vilar et al., 2022) In-Context learning. Other researchers have proposed leveraging LLMs to generate synthetic domain-specific data to facilitate MT domain adaptation (Moslem et al., 2022). Recent research by Agrawal et al. (2022) and Zhang et al. (2023) have demonstrated the critical role of In-Context example selection in enhancing the quality of MT when employing LLMs.

One way to improve MT quality is the incorporation of fuzzy matches (Knowles and Koehn (2018), Bulte and Tezcan (2019b) and Xu et al. (2020)). Fuzzy matches comprise similar segments of previously approved translations stored within parallel datasets collected with computer-assisted translation tools, commonly referred to as translation memories (TMs). Knowles et al. (2018) showed that the utilization of fuzzy matches could enhance the quality of neural MT (NMT) systems by up to 2 BLEU points. Likewise, Bulte and Tezcan (2019b) demonstrated that fuzzy matches could enhance the consistency of MT systems, even in cases where these matches were not entirely precise (Bulte and Tezcan, 2019a). In the same vein, Moslem et al. (2022) focused on the prospect of compelling the translation of new sentence pairs to conform to the fuzzy matches found within the context dataset. They demonstrated that this approach yielded improvements in MT quality, particularly for challenging sentences.

To select fuzzy matches, Moslem et al. (2022) employed an embedding similarity-based retrieval method. This technique is initiated by generating embeddings for each sentence within the TM. These embeddings represent sentences in dense numerical forms, encapsulating their semantic essence. Subsequently, the system retrieves fuzzy matches for a new sentence by identifying TM sentences with the most analogous embeddings. Previous research has established the superiority of embedding similarity-based retrieval over alternative methods, such as Edit Distance (Hosseini et al. (2020)).

Within the few-shot setting, the MT system is provided with a limited number of translated examples (e.g., 2 or 5 fuzzy matches) to assist in generating a translation for a new sentence. This stands in contrast to the zero-shot where the MT system is solely equipped with the source sentence. Moslem et al. (2022) pointed out that incorporating fuzzy matches through few-shot translation prompts could further improve the MT quality. This is attributed to fuzzy matches equipping the MT system with additional insights into the desired translation's style and tone. In the same context, Wang et al. (2021) proposed an embedding similarity-based retrieval algorithm that improved the selection of fuzzy matches, hence the quality of the translation. Knowles and Littell (2022) investigated the role of fuzzy matches in improving low-resource language translation. Their findings underscored the potential for leveraging fuzzy matches to significantly enhance the translation of low-resource language pairs.

3. Methodology

Before the inference phase, we leverage the Sentence-Transformer model to compute embeddings for the segments of the source language (English) streamlining the retrieval of similar sentences using the Facebook AI Similarity Search (FAISS) index system (Douze et al., 2024). This technique enables us to construct contextually rich prompts, allowing the GPT-3.5 Turbo model to follow the style present in domain-specific examples. The performance of LLMs is compared with that of DeepL for the En→Fr language pair. We also evaluate the combination of both LLMs and Encoder-Decoder systems on the En→Fr language pair for the translation of subtitles.

Our particular areas of interest revolve around assessing the efficiency of LLMs in performing the following tasks without requiring additional training:

1. Adapting newly generated translations to seamlessly match the terminology and style in the context,
2. Using translations generated by Encoder-Decoder MT systems as fuzzy matches to further enhance the performance of LLMs,
3. Emphasizing the significance of prompt engineering in improving the capabilities of LLMs by using relevant translation examples for the given sentence request.

3.1. Retrieval of Fuzzy Matches

To efficiently retrieve fuzzy matches for a given input sentence, we use the FAISS system. The latter provides a variety of data structures and algorithms for efficient similarity search, and we have chosen to use the IndexFlatL2 index, which performs an exhaustive search of the index to find the nearest neighbors.

To generate the FAISS index, we first use the Sentence-Transformer model to generate embeddings for each sentence in our preprocessed dataset. Sentence embeddings are dense numerical representations of sentences that capture their semantic meaning and contextual nuances. Once the sentence embeddings are generated for all of the sentences in our dataset, the FAISS index can be created. This process involves the following steps:

1. Loading the sentence embeddings into FAISS,
2. Configuring the FAISS index with the desired parameters, such as the choice of index type and the dimensionality of the embeddings,
3. Building the FAISS index for the whole corpus.

Once the FAISS index is built, it can be used to retrieve fuzzy matches for a given input sentence. To do so, we simply compute the cosine similarity between the input sentence embedding and all of the embeddings in the index. The sentences with the highest cosine similarities are the fuzzy matches for the input sentence. The fuzzy matches are then used to compose context-aware prompts for the GPT-3.5 Turbo model. These prompts provide GPT-3.5 Turbo with additional information about the desired translation, which can help it generate more accurate translations.

3.2. Prompt Composition

For each translation request, our approach leveraged the FAISS index to retrieve the top-k closest sentence embeddings from the domain-specific dataset. The retrieved sentences serve as the foundation for constructing contextually rich prompts for the LLM model.

To facilitate prompt composition and enhance translation quality, we integrated LangChain [1] into our system. LangChain serves as a framework designed for the development of applications leveraging large language models. Its primary objective is to empower developers with the seamless integration of diverse data sources and the facilitation of interactions with other applications. To achieve this goal, LangChain offers modular components, serving as abstractions, and customizable

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

```
Prompt: EN-AR zero-shot translation

    <SystemMessage>
    English: HumanMessage<source_segment>
    Arabic: → AIMessage<predicted_segment>
```

Figure 2: Zero-shot translation prompt

```
Prompt: EN-AR 2-shot translation

    <SystemMessage>
    English: HumanMessage<source_fuzzy_match_1>
    Arabic: AIMessage<g_truth_fuzzy_match_1>

    English: HumanMessage<source_fuzzy_match_1>
    Arabic: AIMessage<g_truth_fuzzy_match_1>

    English: HumanMessage<source_segment>
    Arabic: → AIMessage<predicted_segment>
```

Figure 3: 2-shot translation prompt

use case-specific pipelines, referred to as chains. We utilized the following Langchain's components settings:

- **SystemMessage**: A Message for priming AI behavior, usually passed in as the first of a sequence of input messages. This component plays a pivotal role in guiding the LLM model to follow the desired style and context for subtitle translation tasks. It acts as a foundational prompt template, providing a structured starting point for generating high-quality translations. We set the component to: "Act like a good translator from English to <target_language>. Translate the following English sentence into <target_language>".

- **HumanMessage** and **AIMessage** are Built upon the SystemMessage. We employed a combination of stacked HumanMessage and AIMessage. These messages were carefully crafted to maintain a conversational flow and ensure that the GPT model understands the user's request.

- The last **HumanMessage** in the sequence is the user's sentence request, serving as the input for the translation task.

Figures 2 and 3 show the distinction between zero-shot and few-shot translation prompts. In the zero-shot scenario, only the source sentence and language specifications are provided, prompting the model to autonomously generate the translation guided by the SystemMessage only. Conversely, the few-shot prompt incorporates translation examples, guiding the style of the generated output.

In the evaluation phase of the translation system, we leveraged the above chat message format to interact with the GPT-3.5 Turbo model effectively. Each translation request is encapsulated within a chat message, providing a structured way to communicate with the model. The chat message typically consists of a series of messages, including

a SystemMessage, AIMessages, and a final UserMessage. The SystemMessage sets the context and instructs the model to perform as a skilled translator. AIMessages provide additional guidance, context, or clarifications as needed. The UserMessage encapsulates the user's specific translation request, serving as the input for the model. By crafting messages in this manner, we ensure that the GPT model receives a clear context.

4. Experimental Setup

In the course of our experimentation, we employed the GPT-3.5 Turbo model through its official OpenAI API [2], setting parameters to top-p 1 with a temperature of 0.3 for our translation tasks (Table 1). The choice of these parameters was made deliberately to optimize model performance on the translation task.

Parameters	temperature	top_p
Values	0.3	1

Table 1: GPT-3.5 Turbo parameters with OpenAI API

To simulate a document-level scenario emulating real-world generic text translation tasks, we leveraged the TICO-19 dataset (Anastasopoulos et al., 2020), which contains 3,070 distinct segments for the language pairs under study. English is used as the source language, while Arabic and French as target languages.

With respect to the subtitle translation task, our dataset is taken from TED Talks 2013, commonly known as the Web Inventory (Cettolo et al., 2012), is composed of roughly 150,000 distinct segments for each language pair. The translations are available in more than 109 languages. For the purposes of our study, we chose portions that are relatively in the same TICO-19 domain. We strategically selected three portion sizes (3,200, 6,200, and 9,200 segments) for our experiments to be able to compare the performance with regard to the type of text being translated (generic text or subtitles) as well as to the dataset sizes.

In the following section, we evaluate our method on generic text and subtitles datasets in different portion sizes and compare our results with related work.

5. Experiments and Results

5.1. Generic Text

Previous work by Moslem et al. (2023) has shown the importance of LLMs in adaptive machine translation for In-Context learning using the TICO-19 dataset. In their work, they ran extensive experiments on various language pairs and different types of models (LLMs and Encoder-Decoder models). Table 2 shows the results they obtained for English to Arabic language pair with GPT-3.5 Turbo.

Context	spBLEU↑	CHRF↑	TER↓
Our Results on 1500 Segments			
Zero-shot	37.42	55.48	62.8
Fuzzy 2-shot	45.52	61.7	56.26
Fuzzy 5-shot	46.43	62.41	55.98
Our Results on Full dataset			
Zero-shot	**39.25**	**57.27**	**60.84**
Fuzzy 2-shot	**46.21**	**62.38**	**55.16**
Fuzzy 5-shot	**46.52**	**62.37**	**56.7**
Moslem et al. (2023)'s results on Full dataset			
Zero-shot	38.06	56.35	61.34
Fuzzy 2-shot	46.04	62.18	55.03

Table 2: Our GPT-3.5 Turbo model evaluation results on TICO-19 English-to-Arabic dataset compared to those of Moslem et al. (2023).

With the same settings and parameters for the model and dataset (size and language pair), but with improvement in the prompt composition and selection of the fuzzy match (as explained in sections 3.1 and 3.2), we achieved a significant improvement in the BLEU score as is shown in Table 2 above. For instance, an improvement of 1.19 for zero-shot and 0.17 for 2-shot.

Even in the case of zero-shot translation, notable improvement in BLEU score values is achieved, which is attributed to the effective utilization of prompt composition techniques, using LangChain which helps improve the results.

With the incorporation of fuzzy matches as context for the translation task (with 2 or 5 shots), we can also see an improvement, thanks to the fuzzy matches selection as explained in the previous sections. This technique selects the most contextually relevant and representative shots to the user request on the fly instead of using static fuzzy matches for all sentences as is the case in the work of Moslem et al. (2023). In their work, when composing the prompt, the fuzzy matches were retrieved out of 10 fuzzy matches which were statistically stored as the 10-closest sentences for the overall dataset[3].

To further illustrate our strategy based on prompt composition and dynamic retrieval of fuzzy matches, we conducted experiments on English-to-French language pair. As can be seen in Table 3 below, the resulting translation performance was

[2] https://openai.com/

[3] https://github.com/ymoslem/Adaptive-MT-LLM/blob/main/MT/ChatGPT-BatchTranslation.ipynb

shown to improve in all shot settings for this language pair. We find an improvement of 0.9 for 0-shot setting over Moslem et al. (2023)'s results.

Context	spBLEU↑	CHRF↑	TER↓
Our Results			
Zero-shot	**47.75**	**67.41**	**47.86**
Fuzzy 2-shot	**50.59**	**69.28**	**45.41**
Fuzzy 5-shot	**53.68**	**71.3**	**42.56**
Moslem et al. (2023)'s results			
Zero-shot	46.85	66.75	48.31
Fuzzy 2-shot	49.88	68.33	46.27

Table 3: Our GPT-3.5 Turbo model evaluation results on TICO-19 English-to-French dataset compared to those of Moslem et al. (2023).

It is worth noting that in both Moslem et al. (2023)'s work and ours, the results for the language pair English-Arabic are lower than those of the language pair English-French (Tables 2 and 3).

Our results show the effectiveness of both prompt composition and fuzzy match selection techniques as well as the FAISS index for efficient and fast translation quality.

5.2. Subtitles

Subtitles are short text lines usually at the bottom of the screen that allows the viewer of a film or TV program to follow the dialogue(s) without understanding the audio. We distinguish between same-language subtitles and cross-language subtitles. Same-language subtitles are usually targeted at hearing-impaired viewers or added for educational purposes, while cross-language subtitles enable viewers to enjoy a film in a language different from the audio. Same-language subtitles for hearing-impaired viewers need to include a written or a graphical representation of sounds (e.g. approaching footsteps) which hearing viewers do not need even if they do not understand the original language. Subtitles are typically limited to two rows of text with up to 37 characters on each row. They are displayed on the screen between 3 and 7 seconds. More details about the characteristics of subtitles can be found in Jorge and Remael (2007).

In this section, we report on experiments conducted on the TED Talks 2013 dataset. These experiments encompass various dataset sizes and are run on English-to-Arabic and English-to-French language pairs.

We conducted experiments on 3200 segments of the English-to-Arabic language pair. We found significant improvements in the BLEU score across three experiments as shown in Table 4. The experiments' settings are zero, 2, and 5 fuzzy matches. We notice that the translation performance was shown to improve appreciably with the 5 fuzzy matches setting.

Context	spBLEU↑	CHRF↑	TER↓
3200 Segments			
Zero-shot	21.31	44.03	78.3
Fuzzy 2-shot	22.75	45.21	76.69
Fuzzy 5-shot	24.26	46.23	75.89
6200 Segments			
Zero-shot	22.74	44.85	76.99
Fuzzy 2-shot	22.85	44.9	76.79
Fuzzy 5-shot	24.87	44.93	76.79
9200 Segments			
Zero-shot	22.97	45.14	76.4
Fuzzy 2-shot	22.97	45.14	76.35
Fuzzy 5-shot	24.98	45.12	76.27

Table 4: GPT-3.5 Turbo model evaluation results on English-to-Arabic Ted Talks 2013 dataset with 3200, 6200 and 9200 segments.

Interestingly, we noticed that there is a significant difference in the experimental results for the English-to-Arabic (generic text) and English-to-Arabic (subtitles). For the zero-shot setting and with approximately the same dataset sizes, the BLEU score of the TED Talks 2013 dataset on English-to-Arabic translation is 21.31 (Table 4), whereas the TICO-19 on English-to-Arabic translation has a BLEU score of 39.25 (Table 2). The difference in the results can be attributed to the dataset translation quality and type.

With the same previous experimental settings, we conducted experiments on 6200 subtitle segments. The results show a very slight improvement with increased data size (Table 4). For example, with the 3200 dataset, the BLEU score for the two-shot setting is 22.75, whereas with the 6200 dataset, it is 22.85. This means that the bigger the size is is not necessarily the better.

With the same settings, we tripled our dataset to 9200 segments and noticed a very minor improvement again as shown in Table 4 above. The small increase in the BLEU score even when doubling or tripling the dataset size may be due to the quality difference between the three dataset portions based on manual checks of samples of the dataset. We noticed that the translation quality of the first 3200 segments are better than the additional portions, which explains the slight improvement.

In order to verify the effect of the dataset size on performance, we also conducted experiments on generic text. Results on different size datasets for generic text show a significant improvement when doubling the dataset. Experiments on the full 3071 sentence pairs of the TICO-19 dataset presented in Table 2 show a significantly higher BLEU score than those obtained with roughly half the TICO-19 dataset. By way of example, we noticed an

additional gain of 1.83 in the BLEU score in the zero-shot setting (37.42 on the 1500 sub-dataset and 39.25 on the 3071 full dataset). This means that performance increases with more data in the case of generic text.

We also conducted experiments on English-to-French subtitles and compared the results obtained with those of the English-to-Arabic pair. Table 5 below presents the results of 3000 TED Talks subtitle segments. It can be seen that there is an improvement when adding more fuzzy matches.

Context	spBLEU↑	CHRF↑	TER↓
Zero-shot	44.26	64.72	51.82
Fuzzy 2-shot	44.68	64.99	51.12
Fuzzy 5-shot	45.15	65.34	50.29

Table 5: Evaluation results on TED Talks 2013 dataset composed of 3000 sentence pairs on the English-to-French language pair with GPT-3.5 Turbo.

As we have seen in the case of English-Arabic generic text and subtitle translation, we notice that the evaluation scores of the English-French subtitles are lower than those of the English-French generic text.

In order to compare the results obtained with GPT-3.5 Turbo for the translation of subtitles of the English-to-French language pair, we conducted experiments using the DeepL Encoder-Decoder model, used as API from their official website[4]. Table 6 below shows the results of experiments run on 3000 sentence pairs of the TED Talks 2013 dataset.

spBLEU↑	CHRF↑	TER↓
44.33	64.12	49.91

Table 6: Evaluation results on TED Talks 2013 dataset composed of 3000 sentence pairs on English-to-French language pair with DeepL model.

When used with the zero-shot setting, the Encoder-Decoder slightly outperforms LLMs as can be seen in Tables 6 and 7. However, the results of our experiments demonstrate LLMs' capability to adapt to In-Context learning with few-shot, outperforming Encoder-Decoder MT models. By way of illustration, with a 5-shot setting, GPT-3.5 Turbo achieves an increased BLEU score of 0.82 as shown in Tables 5 and 6.

In the previous experiments, we used the fuzzy matches from the ground-truth translations. In order to see the performance of the combination of LLMs and the Encoder-Decoder model (DeepL), with fuzzy matches constructed using the predicted sentences from DeepL, we conducted experiments on the subtitles dataset composed of 3000 segments. The experimental results are shown in Table 7 below.

Context	spBLEU↑	CHRF↑	TER↓
Zero-shot	44.26	64.72	51.82
Fuzzy 2-shot	45.85	65.67	48
Fuzzy 10-shot	46.01	65.74	47.75

Table 7: Evaluation results on TED Talks 2013 dataset composed of 3000 sentence pairs on English-to-French language pairs with GPT-3.5 Turbo + DeepL model.

We can see that constructing the fuzzy matches from the DeepL Encoder-Decoder model's predictions as a context to the GPT-3.5 Turbo model can improve the quality of the translation of the source segments. By way of illustration, an improvement of 1.17 and 0.86 in the BLEU score for 2-shot and 5-shot, respectively (cf. Tables 5 and 7). This can be explained by the use of the predicted sentences(from DeepL model) to compose the prompt for the GPT-3.5 Turbo model, which supports our previous hypothesis based on manual checks of the quality of the translation in the TED Talks 2013 dataset.

6. Conclusion

This work explored GPT-3.5 Turbo's efficiency in adaptive MT with fuzzy matches. Experimental results were provided showing the effectiveness of our technique with respect to the prompt composition and the selection of the fuzzy matches. The results of our experiments indicate LLMs' capability to adapt to context, outperforming Encoder-Decoder MT models. Our work on subtitles corroborated results from previous work on generic text that the combination of LLMs and Encoder-Decoder models improves the quality of the translation. It was also shown that LLMs and Encoder-Decoder models achieve better results with generic texts than with subtitles for the language pairs En → Ar and En→Fr. Experiments using GPT-3.5 Turbo on different data sizes of English-to-Arabic subtitles indicated that the bigger is not really the better. Further research is required to validate these results and also explore the use of other LLMs in MT, especially for low-resource languages.

7. Bibliographical References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022.

[4] https://www.deepl.com/pro-api/

In-context examples selection for machine translation. *arXiv preprint arXiv:2212.02437*.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federman, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, et al. 2020. Tico-19: the translation initiative for covid-19. *arXiv preprint arXiv:2007.01788*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Boris Bulte and Ayşe Aişe Tezcan. 2019a. Fuzzy matches for improving the consistency of neural machine translation. *arXiv preprint arXiv:1903.11534*.

Bram Bulte and Arda Tezcan. 2019b. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *57th Annual Meeting of the Association-for-Computational-Linguistics (ACL)*, pages 1800–1809.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Proceedings of the Conference of European Association for Machine Translation (EAMT)*, pages 261–268.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Kasra Hosseini, 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.

Daz Cintas Jorge and Aline Remael. 2007. *Audiovisual translation: subtitling*. Routledge.

Rebecca Knowles and Philipp Koehn. 2018. Fuzzy match incorporation for neural machine translation. *arXiv preprint arXiv:1806.08117*.

Rebecca Knowles and Patrick Littell. 2022. Translation memories as baselines for low-resource machine translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6759–6767.

Rebecca Knowles, 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.

Yasmin Moslem, Rejwanul Haque, John D Kelleher, and Andy Way. 2022. Domain-specific text generation for machine translation. *arXiv preprint arXiv:2208.05909*.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2023. Adaptive machine translation with large language models. *arXiv preprint arXiv:2301.13294*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

Shuo Wang, Zhaopeng Tu, Zhixing Tan, Wenxuan Wang, Maosong Sun, and Yang Liu. 2021. Language models are good translators. *arXiv preprint arXiv:2106.13627*.

Jitao Xu, Josep-Maria Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Annual Meeting of the Association for Computational Linguistics*, pages 1570–1579. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. *arXiv preprint arXiv:2301.07069*.