



Interactive Machine Translation with Large Language Models in Low Resources Languages

Sergio Gómez^{1,2}, Miguel Domingo^{1,2}, Francisco Casacuberta^{1,2}

¹PRHLT Research Center - Universitat Politècnica de València, Spain

²ValgrAI - Valencian Graduate School and Research Network for Artificial Intelligence, Spain

sgomgon@prhlt.upv.es, midobal@prhlt.upv.es, fcn@prhlt.upv.es

Abstract

Despite the advances in machine translation (MT) derived from the arrival of large language models (LLMs), computers are still not able to obtain high-quality translations for many tasks. However, the time spent to obtain those translations is significantly shorter than what a human would take. Interactive machine translation (IMT) emerges as an intermediate solution between both paradigms: it proposes an iterative, collaborative scheme in which the system proposes translation hypotheses that are partially validated and corrected by a human. Each time, the system reacts to this feedback providing a new hypothesis.

The rise of LLMs and the good performance they obtain in the MT discipline invites us to try to deploy them to the IMT paradigm. Thus, throughout this paper we discuss the experiments we have performed using some representative models for IMT for low-resourced languages. Moreover, we present the results we have obtained and how the models have performed for this type of tasks.

Index Terms: Language model, machine translation, interactive machine translation, segment-based interactive machine translation, restricted generation

1. Introduction

Nowadays the quality of translations obtained automatically is improving continuously with new advances in training and architecture of the models. However, it is far from being perfect, specially for low-resourced languages [1]. Thus, when high-quality translations are critical, a human translator needs to review and correct the machine translation (MT) hypothesis in a process known as post-editing.

Among the many proposals to reduce the human effort from the post-editing process, interactive machine translation (IMT) proposed a collaborative framework in which human and machine work together to construct the final translation: instead of manually correcting the complete translation hypothesis, the expert can provide the system with some feedback which is used to generate a new hypothesis. This process is repeated until the user is satisfied with the system's hypothesis.

Throughout the years, different protocols have been proposed in the literature, among which we find prefix-based [2] and segment-based [3,4] IMT. In the first paradigm, the human expert reviews the system's translation hypothesis and corrects the leftmost wrong word—which inherently means that all preceding words are correct. The system reacts to this feedback by generating a new suffix that completes the prefix to conform a new hypothesis. This process is repeated until the user is satisfied with the current hypothesis.

In the second paradigm, the process also starts with the user reviewing the system's hypothesis. Now, they can validate sequences of words which they consider to be correct, and make a word correction at any point in the sentence (breaking the left-to-right limitation from the previous protocol). The system reacts to this feedback by generating a new hypothesis and, thus, starting a new iteration of the process.

In this work, we aim to fulfill a comparative study of how current large language model (LLM) perform for the task of IMT using the two aforementioned protocols. To accomplish this goal, we selected four different representative models that have been distinguished from the rest. These models have been known for performing well for MT tasks and for covering a wide range of languages. We tested them for two low-resourced language pairs: Galician–English and Swahili–English. These were selected corresponding to the rarest languages in the train datasets of the models. In that way, we could explore whether the user effort increases when the model does not perfectly fit the language.

2. Related work

Since MT alone is not enough to obtain high quality translations, many paradigms have been proposed with the aim of reducing the user effort during the post-editing process. These kinds of systems have been gathered under the umbrella of post-editing and carried one step further with the proposal of IMT [5]. This first attempt was characterized by a first action of the user in which they selected a section of the source sentence and started typing its translation. At every keystroke of the user, the system displayed a set of possible full words that the user could choose.

Projects as TransType [6], CasMacat [7], and TranSmart [8] established a workbench with an array of features that were not included in other tools at that time, such as other ways of editing a translation and different visualizations of the information that effectively helped to reduce the user effort.

All of these projects kept the use of the prefix-based protocol, characterized by the user reviewing the translation hypothesis in reading order (i.e., from left to right) until they found the first error, which they would correct. Therefore, the user validated a longer prefix at each iteration while the system proposed a fitting suffix as completion of the translation. Later, the protocol was improved with advances in the suffix generation [9], new kinds of interaction [10] and visualization of the information with confidence measures [11, 12].

One of these improvements was the segment-based protocol introduced by Domingo et al. [3] and Peris et al. [4], which has also evolved over the years. Techniques such as reinforcement learning [13] and confidence measures [14], have been

deployed to this paradigm to obtain validated segments and improve segment predictions with text-infilling methods [15].

The latest works on this field are represented by the introduction of LLMs to IMT systems [16, 17]. These new integrations bring the strength of LLMs in MT to obtain better completions between validated segments and suffixes. In these previous works, LLMs are deployed for the first time to prefix-based [16] and segment-based [17] IMT through the use of mBART [18] and mT5 [19]. In this work, we experimented with a total of four different LLMs, and compared and studied the performance of both protocols under a low-resourced setting.

3. Interactive machine translation

IMT follows the same mathematical framework as MT. Thus, given a source sentence x_1^J of length J , its goal is to find the most likely translation $\hat{y}_1^{\hat{I}}$ of length \hat{I} [20]:

$$\hat{y}_1^{\hat{I}} = \arg \max_{y_1^{\hat{I}}} Pr(y_1^{\hat{I}} | x_1^J) \quad (1)$$

Depending on the protocol, the regular search in the translation space will be constrained by the user’s feedback.

3.1. Prefix-based IMT

The prefix-based paradigm starts with the system proposing an initial translation y_1^I of length I . The user, then, reviews the hypothesis and corrects the leftmost wrong word y_i . With this action, they are also validating all the words that precede this correction, forming a validated prefix \tilde{y}_1^i , that includes the corrected word \tilde{y}_i . The system immediately reacts to this user feedback ($f = \tilde{y}_1^i$), generating a suffix \hat{y}_{i+1}^I that completes \tilde{y}_1^i to obtain a new translation of x_1^J : $\hat{y}_1^I = \tilde{y}_1^i \hat{y}_{i+1}^I$. This process is repeated until the user accepts the system’s complete suggestion.

The neural equivalent of the suffix generation was formalized by Peris et al. [4] as follows:

$$p(\hat{y}_{i'} | \hat{y}_1^{i'-1}, x_1^J, f = \tilde{y}_1^i; \Theta) = \begin{cases} \delta(\hat{y}_{i'}, \tilde{y}_{i'}) & \text{if } i' \leq i \\ \bar{\mathbf{y}}_{i'}^\top \mathbf{p}_{i'} & \text{otherwise} \end{cases} \quad (2)$$

where x_1^J is the source sentence; \tilde{y}_1^i is the validated prefix together with the corrected word; Θ are the models parameters; $\bar{\mathbf{y}}_{i'}^\top$ is the one hot codification of the word i' ; $\mathbf{p}_{i'}$ contains the probability distribution produced by the model at time-step i' ; and $\delta(\cdot, \cdot)$ is the Kronecker delta.

This is equivalent to a forced decoding strategy, which is very similar to eq. (1): at each iteration, the process consists in a regular search in the translations space but constrained by the prefix \tilde{y}_1^i .

3.2. Segment-based IMT

The segment-based IMT translation process also starts with the system proposing an initial translation hypothesis y_1^I of length I , which is reviewed by the user. Now, they can validate those sequences of words which they consider to be correct ($\hat{\mathbf{f}}_1, \dots, \hat{\mathbf{f}}_N$; where N is the number of non-overlapping validated segments). Next, they are able to merge two consecutive segments $\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_{i+1}$ into a new one. Finally, they make a word correction—introducing a new one-word validated segment, $\hat{\mathbf{f}}_i$, which is inserted in $\hat{\mathbf{f}}_1^N$.

The system responds to this user feedback by generating a sequence of new translation segments $\hat{\mathbf{g}}_1^N = \hat{\mathbf{g}}_1, \dots, \hat{\mathbf{g}}_N$;

where each $\hat{\mathbf{g}}_n$ is a subsequence of words in the target language. This sequence complements the user’s feedback to conform the new hypothesis:

$$\hat{y}_1^I = \hat{\mathbf{f}}_1, \hat{\mathbf{g}}_1, \dots, \hat{\mathbf{f}}_N, \hat{\mathbf{g}}_N \quad (3)$$

Peris et al. [4] formalized the word probability expression for the words belonging to a validated segment $\hat{\mathbf{f}}_n$ as:

$$p(y_{i_n+i'} | y_1^{i_n+i'-1}, x_1^J, \hat{\mathbf{f}}_1^N; \Theta) = \mathbf{y}_{i_n+i'}^\top \mathbf{p}_{i_n+i'}, \quad (4) \\ 1 \leq i' \leq \hat{l}_n$$

where l_n is the size of the non-validated segment generated by the system, which is computed as follows:

$$\hat{l}_n = \arg \max_{0 \leq l_n \leq L} \frac{1}{l_n + 1} \sum_{i'=i_n+1}^{i_n+l_n+1} \log p(y_{i'} | y_1^{i'-1}, x_1^J; \Theta) \quad (5)$$

4. Experimental framework

This section presents the experimental setting for assessing the quality of our proposal. We start by introducing the evaluation metrics and corpora. Then, we present the LLMs studied and describe how we built our IMT systems and the user simulation.

4.1. Evaluation metrics

We made use of the following well-known metrics in order to assess our proposal:

Word Stroke Ratio (WSR) [21]: measures the number of characters typed by the user, normalized by the number of words in the final translation.

Mouse Action Ratio (MAR) [2]: measures the number of mouse actions made by the user, normalized by the number of characters in the final translation.

Additionally, we assessed the initial translation quality of each system using:

Bilingual evaluation understudy (BLEU) [22]: computes the geometric average of the modified n -gram precision, multiplied by a brevity factor that penalizes short sentences. In order to ensure consistent BLEU scores, we used *sacreBLEU* [23] for computing this metric.

Translation error rate (TER) [24]: computes the number of word edit operations (insertion, substitution, deletion and swapping), normalized by the number of words in the final translation. It can be seen as a simplification of the user effort of correcting a translation hypothesis on a classical post-editing scenario.

Finally, we applied approximate randomization testing (ART) [25]—with 10,000 repetitions and using a p -value of 0.05—to determine whether two systems presented statistically significance.

4.2. Corpora

In order to perform the evaluation we selected a parallel dataset that contained sentences in two low-medium resourced languages. Thus, we made use of the corpus of the High Performance Language Technologies (HPLT) [26] project, specifically the English–Galician and English–Swahili language pairs. Table 1 showcases the statistics of the data. We selected these

two languages pairs attending to their involving in the pre-training of the LLMs which we made use of (see section 4.3). Additionally, this dataset was created recently. Therefore, it could not have been used in the pre-training of the aforementioned models.

Table 1: *Corpora statistics. K denotes thousands and M millions. |S| stands for number of sentences, |T| for number of tokens and |V| for size of the vocabulary.*

		HPLT	
		En-Gl	En-Sw
Train	S	956.8K	1.7M
	T	13.1M / 12.5M	21.0M / 20.0M
	V	22.5K / 28.1K	25.6K / 28.0K
Validation	S	3.0K	3.0K
	T	40.6K / 38.9K	36.3K / 34.5K
	V	5.6K / 9.4K	5.2K / 6.9K
Test	S	3.0K	3.0K
	T	41.9K / 40.3K	36.8K / 34.9K
	V	5.5K / 9.4K	5.3K / 7.2K

4.3. Models

In our study, we made use of four different LLMs, which we chose due to their innovative approach and good performance at the time they were launched.

mBART [18] (611M params.): The family of models known as *mBART* have proved to offer a very good performance on the 50 languages covered by the largest models. It was one of the first approaches to explore the possibilities of multilinguality in MT. It uses several encoders and decoders associated with the source and target languages, and it is designed according to the technique of pivot language.

M2M [27] (418M params.): This model refuses to use pivot languages and condense the architecture into a single encoder-decoder. It codifies the source language at the source sentence and the target language at the representation obtained from the encoder. With this strategy, it duplicates the number of covered languages while maintaining the quality of the translations.

Flan-T5 [28] (250M params.): This LLM also uses the single encoder-decoder architecture but is trained differently. Chung et al. [28] described how the instruction-training known as Flan [29] improves the performance of the model, obtaining a smaller model than T5 [30] that worked better.

NLLB [31] (600M params.): Obtained from the *No Language Left Behind* project [31], nowadays this is the model that covers the widest range of languages. It also uses a shared encoder-decoder architecture, but with a difference from the classical Transformer [32]: a sparsely gated mixture of experts that expands the number of parameters without increasing the amount of operations required to obtain a translation.

4.4. IMT’s systems

To implement the generation of a hypothesis taking into account the user’s feedback, the system composes a tuple (x, F, s, t, M) , where x represents the input sentence to translate, F is the set of validated segments, s and t are the input and output languages respectively and M states for the translation model. Using that tuple, the system constrains the generation of the model to generate the next translation hypothesis.

Therefore, in the prefix-based approach, the beam search

is restricted to first generate it. Then, the beam search is expanded normally. In the segment-based approach, the functioning is slightly different: At the beginning, the search can be also forced to generate a first segment. Then, the beam search works as usual until the first token of the next segment is expanded in some branch. After that, this branch is force to generate the full segment. The search continues as described until the translation is completed. Thus, at each iteration of the translation the set of segments is updated with the new information granted by the user. The new segments are added, and the extended ones are substituted by the former ones. If any of them is fixed to the beginning it gets registered in the state of the system as well.

Our implementation of these IMT protocols is available at GitHub.¹

4.5. User simulation

Due to the high time and economic cost of conducting frequent human evaluations at the development stage, we conducted the evaluation using simulated users whose goal was to generate the translations from the reference.

Prefix-based protocol: In this protocol, the user corrects the leftmost wrong word from the hypothesis—inherently validating all preceding words (the prefix). We achieve this behavior by comparing hypothesis and reference at a word level, and correcting the first (from left to right) word that differs (using the equivalent word in the reference as the user’s correction). This process has a fixed cost of one word stroke (to account for the correction) and one mouse action (to place the mouse in the corresponding position), and is repeated until hypothesis and reference are the same, at which point the user validates the hypothesis (with a cost of 1 mouse action).

Segment-based protocol: The simulation starts with the system offering an initial hypothesis. Then, the user reviews it and validates word segments, which are obtained by computing the longest common subsequence [33] between hypothesis and reference. This has an associated cost of one mouse action for each one-word segment and two for each multi-word segment. After this, the user looks for pairs of consecutive validated segments which could be merged into a single larger segment (i.e., they appear consecutively in the reference but are separated by some words in the hypothesis). If there are, then they merge them, increasing mouse actions in one if there was a single word between the segments, or two otherwise. Finally, they correct the leftmost wrong word². This process is repeated until the hypothesis and the reference are the same.

5. Results

Table 2 presents the results of our evaluation. Overall, we can observe how the segment-based protocol yielded the greatest reduction of the typing effort (as indicated by WSR). However, this reduction came with a significant effort of the mouse effort (as indicated by MAR). This behavior was already known in the literature. However, unlike in previous works (e.g., [4]), this increment is significantly greater than the reduction of the typing effort compared with the prefix-based protocol—whose reduction of the typing effort is around 10/5 WSR points (for Gali-

¹https://github.com/sergiogg-ops/TFM_IMT/

²For the sake of simplicity and without loss of generality, we followed the segment-based protocol authors assumption that the user always corrects the leftmost wrong word [3,4].

Table 2: Comparison of the prefix-based and segment-based protocols under a low-resourced setting. The initial translation quality is meant to be a starting point comparison of each system. Best results are denoted in **bold**. All results are statistically different between all approaches except those denoted with †. The abbreviations En, Gl and Sw stand for English, Galician and Swahili, respectively.

Language Pair	Model	Translation Quality		Prefix-based		Segment-based	
		BLEU [†]	TER [↓]	WSR [↓]	MAR [↓]	WSR [↓]	MAR [↓]
En-Gl	Flan-T5	10.0	79.1	55.8	10.2	53.7	52.6
	M2M	33.2	37.2	79.1	6.4	32.8	29.8
	NLLB	23.3	59.5	41.6	6.3	38.0	45.2
	mBART	29.5	40.5	77.0	6.6	34.6	32.6
Gl-En	Flan-T5	20.7	60.4	39.9 [†]	6.8	38.7	43.3
	M2M	34.4	38.1 [†]	76.9	6.0[†]	29.3[†]	29.1 [†]
	NLLB	26.8	60.5	37.5 [†]	6.0[†]	33.1	40.6
	mBART	33.4	38.8 [†]	77.0	6.6	28.7[†]	28.6 [†]
En-Sw	Flan-T5	10.9	194.7	71.0	13.6	63.8	47.9
	M2M	52.0	39.3	54.0	6.1	25.0	21.4
	NLLB	47.5	48.3	28.5	5.3	28.5	23.7
	mBART	44.2	88.9	60.6	7.4	35.9	29.4
Sw-En	Flan-T5	7.3	759.4	52.9	8.8	63.8	58.9
	M2M	52.6	35.8	51.0	6.1	23.5	22.7
	NLLB	47.0	47.3	28.6	5.3	24.9 [†]	30.8
	mBART	50.5	53.6	53.5	7.3	25.6 [†]	25.1

cian and Swahili, respectively) worse than the segment-based one, but the increment of the use of the mouse is one order of magnitude smaller. We have observed that in some cases the simulated user needs to join segments to prevent other words to be introduced between them quite often. We believe that the excessive verbosity of the LLMs might be related to these interferences between valid segments. Therefore, the use of this kind of models may lead to the increase of the mouse effort when a segment-based IMT system is being used.

Regarding each model, despite having the lowest initial translation quality, *Flan-T5* had the second-best performance for all language pairs (except English–Swahili) for prefix-based IMT, but it had the worst performance (by a large margin) for the segment-based protocol. On the opposite way, despite having the highest initial translation quality, *M2M* and *mBART* had the worst performance for the prefix-based protocol, and the best ones for segment-based (*M2M* performing slightly better). Finally, *NLLB* had the best performance for prefix-based IMT in all cases, and an average performance for segment-based IMT.

Overall, the greatest reduction of the human effort are achieved by *NLLB* on a prefix-based setting: the typing reduction is consistently the second best—the best one always being achieved by *M2M* on a segment-based setting—and the increment of the mouse effort is minimal.

5.1. Qualitative analysis

Once we have studied the quantitative results, we proceed to analyze which are the cases in which the IMT systems performed well or failed to reduce the human effort.

The prefix-based approach achieved the overall best post-editing effort reduction. Corrections are now resolved in fewer iterations, requiring little effort by the human expert. e.g., the sentence “Agrasar proponnos tres receitas de Nadal para maridar cos nosos tres viños albariños: Pazo Baión, Gran a Gran e Vides de Fontán.” is translated in seven iterations to “Agrasar proposes three Christmas recipes to pair with our three Albariño wines: Pazo Baión, Gran a Gran and Vides de Fontán.”. This example comes from the *NLLB* model and took a 28.1 word stroke rate (WSR) and a 5.6 mouse action rate (MAR) of effort. This happens because a great part of the words of the final translation are not written but validated. In other words, the model has been able to suitably autocomplete the translation.

Nevertheless, some sentences such as “151 Ee Bwana, Wewe U karibu, Na maagizo yako yote ni kweli.” needs 14

iterations to be translated to “151 Thou art near, O LORD; and all thy commandments are truth.”. This example comes from the *Flan-T5* model and obtained a 86.7 WSR and a 9.52 MAR. In the process, the model won’t stop adding a token until the completion of the maximum length in many iterations. As a result, the human expert needs to type a large amount of words.

The segment-based protocol was able to further reduce the typing effort since the user can validate sequence of words which are already correct (and that could be changed in successive iterations of the prefix-based approach), preserving them for future iterations. e.g., the sentence “Entón, cando se elabora un diagrama similar para un cúmulo cuxas distancias non se coñecen, a posición da secuencia principal pode compararse coa do primeiro cúmulo e as distancias estimadas.” is translated to “Then, when similar diagram is plotted for a cluster whose distance is not known, the position of the main sequence can be compared to that of the first cluster and the distance estimated.” in just four iterations. The required human effort is defined by a 13.9 WSR and a 17 MAR. This example comes from the *M2M* model.

However, there are other sentences that need more effort to be translated. e.g., the sentence “Best programu ya kununua na kuuza karibu na wewe!” needed seven iterations to be translated into “The #1 Marketplace to Buy & Sell”. In almost each of them, the simulated user typed a correction and joined the correction of the previous iteration to the prefix. Due to that, the prefix-based approach would have been more suitable for this sentence. This example comes from the *Flan-T5* model and the user effort was a 100.0 WSR and a 90.9 MAR.

6. Conclusions

In this work, we have studied the performance of prefix-based and segment-based IMT under a low-resourced setting, using four different LLMs. Unlike previous works from the literature (e.g., [3]), the prefix-based protocol yielded the best overall effort reduction. We believe that the excessive verbosity of the LLMs might be related to these interferences between valid segments.

As a future work, we would like to try other languages that included a change in the alphabet or the script direction. Additionally, it could be interesting to perform a specific review of which are the best models in low resources languages, i.e. different neural architectures and LLMs.

7. Acknowledgements

This work received funding from *ValgrAI*; *Generalitat Valenciana*; *European NextGenerationEU/PRTR* (project FAKEnHATE-PdC; PDC2022-133118-I00); and *MCIN/AEI* and *ERDF/EU* (project LLEER; PID2021-124719OB-I00).

8. References

- [1] T. Kocmi, E. Avramidis, R. Bawden, O. Bojar, A. Dvorkovich, C. Federmann, M. Fishel, M. Freitag, T. Gowda, R. Grundkiewicz *et al.*, “Findings of the 2023 conference on machine translation (wmt23): LLMs are here but not quite there yet,” in *Proceedings of the Eighth Conference on Machine Translation*, 2023, pp. 1–42.
- [2] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J.-M. Vilar, “Statistical approaches to computer-assisted translation,” *Computational Linguistics*, vol. 35, pp. 3–28, 2009.
- [3] M. Domingo, A. Peris, and F. Casacuberta, “Segment-based interactive-predictive machine translation,” *Machine Translation*, vol. 31, pp. 163–185, 2017.
- [4] Á. Peris, M. Domingo, and F. Casacuberta, “Interactive neural machine translation,” *Computer Speech & Language*, vol. 45, pp. 201–220, 2017.
- [5] G. Foster, P. Isabelle, and P. Plamondon, “Target-text mediated interactive machine translation,” *Machine Translation*, vol. 12, pp. 175–194, 1997.
- [6] P. Langlais, G. Foster, and G. Lapalme, “TransType: a computer-aided translation typing system,” in *Proceeding of the ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*, 2000, pp. 46–51.
- [7] V. Alabau, R. Bonk, C. Buck, M. Carl, F. Casacuberta, M. García-Martínez, J. González-Rubio, P. Koehn, L. A. Leiva, B. Mesa-Lao, D. Ortiz-Martínez, H. Saint-Amand, G. Sanchis-Trilles, and C. Tsoukala, “CASMACAT: An open source workbench for advanced computer aided translation,” *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 101–112, 2013.
- [8] G. Huang, L. Liu, X. Wang, L. Wang, H. Li, Z. Tu, C. Huang, and S. Shi, “Transmart: A practical interactive machine translation system,” *arXiv preprint arXiv:2105.13072*, 2021.
- [9] P. Koehn, C. Tsoukala, and H. Saint-Amand, “Refinements to interactive translation prediction based on search graphs,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 574–578.
- [10] G. Sanchis-Trilles, D. Ortiz-Martínez, J. Civera, F. Casacuberta, E. Vidal, and H. Hoang, “Improving interactive machine translation via mouse actions,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 485–494.
- [11] J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta, “Balancing user effort and translation error in interactive machine translation via confidence measures,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 173–177.
- [12] Á. Navarro and F. Casacuberta, “Confidence measures for interactive neural machine translation,” in *Proceedings of the IberSPEECH conference*, 2021, pp. 195–199.
- [13] T. K. Lam, J. Kreutzer, and S. Riezler, “A reinforcement learning approach to interactive-predictive neural machine translation,” *arXiv preprint arXiv:1805.01553*, 2018.
- [14] T. Zhao, L. Liu, G. Huang, H. Li, Y. Liu, L. GuiQuan, and S. Shi, “Balancing quality and human involvement: An effective approach to interactive neural machine translation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 9660–9667.
- [15] Y. Xiao, L. Liu, G. Huang, Q. Cui, S. Huang, S. Shi, and J. Chen, “Bitiimt: a bilingual text-infilling method for interactive machine translation,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 1958–1969.
- [16] Á. Navarro and F. Casacuberta, “Exploring multilingual pre-trained machine translation models for interactive translation,” in *Proceedings of Machine Translation Summit*, 2023, pp. 132–142.
- [17] —, “Segment-based interactive machine translation for pre-trained models,” *arXiv preprint arXiv:2407.06990*, 2024.
- [18] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, “Multilingual translation with extensible multilingual pretraining and finetuning,” *arXiv preprint arXiv:2008.00401*, 2020.
- [19] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” *arXiv preprint arXiv:2010.11934*, 2021.
- [20] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, “The mathematics of statistical machine translation: Parameter estimation,” *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.
- [21] J. Tomás and F. Casacuberta, “Statistical phrase-based models for interactive computer-assisted translation,” in *Proceedings of the International Conference on Computational Linguistics/Association for Computational Linguistics*, 2006, pp. 835–841.
- [22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [23] M. Post, “A call for clarity in reporting bleu scores,” in *Proceedings of the Third Conference on Machine Translation*, 2018, pp. 186–191.
- [24] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A study of translation edit rate with targeted human annotation,” in *Proceedings of the Association for Machine Translation in the Americas*, 2006, pp. 223–231.
- [25] S. Riezler and J. T. Maxwell, “On some pitfalls in automatic evaluation and significance testing for mt,” in *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 57–64.
- [26] O. De Gibert, G. Nail, N. Arefyev, M. Bañón, J. Van Der Linde, S. Ji, J. Zaragoza-Bernabeu, M. Aulamo, G. Ramírez-Sánchez, A. Kutuzov *et al.*, “A new massive multilingual dataset for high-performance language technologies,” *arXiv preprint arXiv:2403.14009*, 2024.
- [27] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary *et al.*, “Beyond english-centric multilingual machine translation,” *Journal of Machine Learning Research*, vol. 22, no. 107, pp. 1–48, 2021.
- [28] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, “Scaling instruction-finetuned language models,” *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.
- [29] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, “Finetuned language models are zero-shot learners,” *arXiv preprint arXiv:2109.01652*, 2021.
- [30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.
- [31] M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heffernan, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard *et al.*, “No language left behind: Scaling human-centered machine translation,” *arXiv preprint arXiv:2207.04672*, 2022.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [33] A. Apostolico and C. Guerra, “The longest common subsequence problem revisited,” *Algorithmica*, vol. 2, pp. 315–336, 1987.