Synslator: An Interactive Machine Translation Tool with Online Learning

Jiayi Wang*
University College London
London, UK
ucabj45@ucl.ac.uk

Ke Wang*
Alibaba Group
Beijing, China
wk258730@alibaba-inc.com

Fengming Zhou
Enjoy Global, Inc.
Wilmington, DE, USA
zfm@enjoy.ai

Chengyu Wang
Alibaba Cloud
Hangzhou, China
chengyu.wcy@alibaba-inc.com

Zhiyong Fu
Alibaba Group
Hangzhou, China
zhiyong.fzy@alibaba-inc.com

Zeyu Feng
Alibaba Group
Hangzhou, China
zeyu.fz@alibaba-inc.com

Yu Zhao†
Alibaba Group
Hangzhou, China
kongyu@alibaba-inc.com

Yuqi Zhang†
The Institute Of Service-Oriented Manufacturing
Hangzhou, China
zhangyuqi@isom.org.cn

ABSTRACT

Interactive Machine Translation (IMT) advances the computer-aided translation (CAT) paradigm, enabling collaboration between machine translation systems and human translators for high-quality outputs. This paper presents Synslator, a CAT tool designed for IMT and proficient in online learning with real-time translation memories. Synslator accommodates different CAT service deployments by integrating two neural translation models for online learning and a language model to boost translation fluency interactively. Our evaluations demonstrate the system's online learning effectiveness, showing a 13% increase in post-editing efficiency with Synslator's interactive features. A tutorial video is provided at: https://youtu.be/K0vRsb2lTt8.

CCS CONCEPTS

• **Computing methodologies** → **Machine translation**; *Natural language generation.*

KEYWORDS

computer-aided translation, interactive machine translation

ACM Reference Format:
Jiayi Wang, Ke Wang, Fengming Zhou, Chengyu Wang, Zhiyong Fu, Zeyu Feng, Yu Zhao, and Yuqi Zhang. 2024. Synslator: An Interactive Machine Translation Tool with Online Learning. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651240

*Both authors contribute equally to this work.
†Corresponding Authors.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05
https://doi.org/10.1145/3589335.3651240

1 INTRODUCTION AND RELATED WORKS

We have witnessed advancements made in the field of machine translation [1, 14, 16], which progressively improve translation quality. These advancements have prompted a transformation in the industry, with a shift from exclusive reliance on human translations to the integration of computer-aided translation (CAT) methods [3, 7, 12]. Instead of translating from scratch, humans engage in post-editing tasks, refining machine translations to yield the final approved results, and thus considerably improving the translation efficiency. Post-editing used to be generally static, wherein machines ceased to respond to human modifications as soon as human post-editing began [7]. Recent studies have explored interactive procedures [8, 10, 13, 18], enabling a more collaborative process between humans and machines, where machines can dynamically adjust translations in line with the edits made by humans.

Translation Memory (TM) is a key component that can be leveraged in CAT [6]. As human undertake post-editing with CAT tools, online incremental TMs can be invariably accumulated. Hence, the capability to use TMs for online learning emerges as a critical attribute for CAT. In fact, there are different environment settings for the deployment of CAT services. In environments where the deployment of CAT allows for authorized usage of TMs, it is feasible to utilize translation memories for model fine-tuning [2, 19]. While in different settings such as public cloud solutions for CAT services, where translation memories are usually introduced online by users and not authorized to use for model training, it is more beneficial to have a translation model capable of handling online TMs during inference. For instance, Khandelwal et al. [9] predicts target words with a k-nearest-neighbor (kNN) classifier over a datastore of cached TM examples during inference.

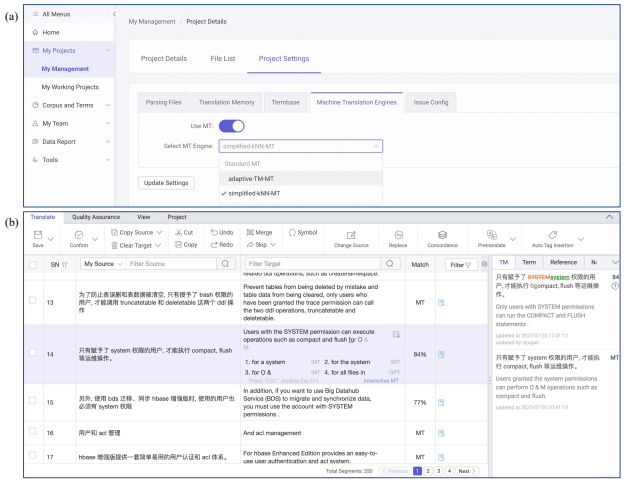

Figure 1: The user interfaces of Synslator: (a) the project setting interface, (b) the post-editing interface.

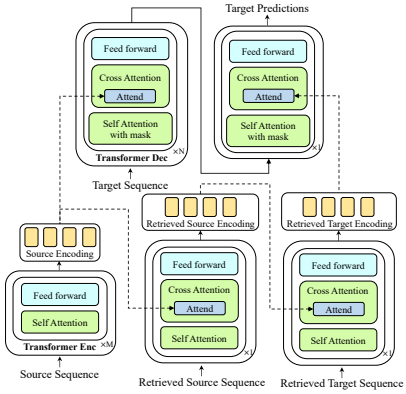

Figure 2: The *adaptive*-TM-MT framework.

In this paper, we introduce Synslator, a CAT tool that enhances IMT by offering real-time automated suggestions. It supports subword level inputs, ensuring flexibility in editing machine translations. Specifically, Synslator integrates two models for translation memory and online learning: an adaptive neural machine translation model (*adaptive*-TM-MT) and a simplified nearest-neighbor retrieval model (*simplified*-kNN-MT). Additionally, it leverages a GPT-based language model (LM) to provide suggestions aimed at improving monolingual fluency and style.

2 SYNSLATOR: THE PROPOSED SYSTEM

Synslator allows users to create translation projects, configure its respective settings, and perform post-editing on machine translation results in an interactive mode.[1] As depicted in the screenshots in Figure 1, there are two user interfaces that humans utilize to finish a translation task. The interface (a) allows adjustments for project settings, while the interface (b) supports human post-editing.

[1]A tutorial video, available at this link: https://youtu.be/K0vRsb2lTt8, demonstrates how Synslator works. The example project shown in the video focuses on legal translations from Chinese to English. We presume in this case that the CAT tool is set up as a public cloud service, and the user has already uploaded a suitable TM dataset.

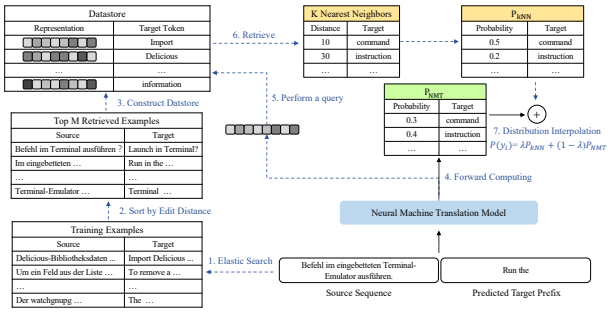

Figure 3: The *simplified*-*k*NN-MT framework with sequentially numbered workflow steps.

2.1 Project Setting Interface

Users are presented with choices for file parsing, selections of translation memory, selections of termbase, and options to choose different machine translation engines. The file parsing functionality is employed to segment sentences if a document is uploaded for translation. We will focus on the functions of translation memory, termbase and machine translation engines in this section.

2.1.1 ***Translation Memory and Termbase.*** After creating a translation project, users can upload related TMs and bilingual termbase. For each source sentence, Synslator will present the most relevant TM including its source and target translation as reference in the post-editing interface. The searching process is initially carried out by an open-sourced distributed search engine, ElasticSearch,[2] which retrieves as most as 64 bilingual sentence pairs that exhibit the highest relevance scores based on the source. Subsequently, from these bilingual sentence pairs, Synslator selects the one demonstrating the most similarity based on the edit distance on the source side. The minimum threshold for the edit distance is denoted as the Minimum Match Rate, and its value can be set in the Project Settings interface. When it comes to translating terms, Synslator utilizes an exact match strategy to locate their respective translations from the bilingual termbase. If multiple matches are found, all of them will be displayed.

2.1.2 ***Machine Translation Engines.*** Post-editing with a CAT tool results in incremental online TMs. Regarding the varying deployment environments associated with CAT services, Synslator utilizes two distinct models for online learning.

adaptive-TM-MT. When deploying CAT services with permission to use training TMs, leveraging TMs for fine-tuning would enhance domain-specific translation performance. Moreover, incorporating related TMs as an extra input would further improve the model. Bapna and Firat [2] retrieves neighbors from TMs and incorporates them into the model through Conditional Source Target Memory. Inspired by their work, we propose the *adaptive*-TM-MT, as illustrated in Figure 2. Given a pre-trained Transformer model, we fine-tune it with TMs as domain-specific training data. For each parallel sentence pair in TMs, we first use the pre-trained encoder and decoder to encode the TM's source and target sequences; afterwards, we execute a retrieval process to locate the nearest neighbor from the remaining TMs in the same way as in Section 2.1.1. The

[2]https://github.com/elastic/elasticsearch

retrieved source is encoded with one Transformer encoder layer, and is integrated with the encoder representation of the source sequence via a cross-attention. Similarly, the retrieved target is encoded, attending to the encoded retrieved source memory. Finally, we add one Transformer decoder layer upon the original decoder module, attending the encoded retrieved target memory. The *adaptive*-TM-MT is trained offline with historical TMs, and used to handle real-time incremental TMs online.

simplified-kNN-MT. For scenarios where using TMs in training is not feasible, such as in public cloud CAT services, we develop a model, *simplified-k*NN-MT, capable to handle plug-in TMs at inference, inspired by *k*NN-MT [9]. It adopts the retrieval approach detailed in Section 2.1.1 to obtain a smaller amount of relevant TMs per source sentence, and then gathers up to 16 TMs via edit distance with a minimum threshold of 0.4 for datastore construction. This procedure alleviates computational complexity and storage requirements in *k*NN-MT, thereby enhancing its applicability in practical scenarios. Finally, each sentence is linked to a tailored datastore, facilitating target generation by interpolating the distribution of *k*NN predictions as *k*NN-MT does. While a related technique is employed in recent research [4], our method simplifies the process by eliminating the need for their adaptive *k*NN retrieval.

2.2 Post-Editing Interface

Upon configuring project settings, users access the post-editing interface, shown in Figure 1 (b), which initially displaying machine translation results. Human post-editing triggers Synslator to adjust translations based on edits, enabling ongoing improvements. This interaction continues until translations reach the desired quality.

2.2.1 ***Workflow of Post-Editing.*** Translation memories and termbases, detailed in Section 2.1.1, are accessible on the right side of the post-editing interface, allowing translators to incorporate them into translations with a double-click and perform edits as needed. Incremental TMs from real-time editing are merged with existing memories, enabling the *adaptive*-TM-MT and *simplified*-kNN-MT models' online learning. For *adaptive*-TM-MT, Synslator generates translations utilizing the source and top-ranked TM. For *simplified*-kNN-MT, Synslator uses relevant TMs to create a datastore for kNN retrievals.

Translators can edit at the subword level, and the model, informed by subword inputs, completes the current target word and generates subsequent words. This is enabled by our subword-prefix decoding algorithm, as detailed in Section 2.2.2. Predictions with high confidence (e.g., translation probability above 0.6) are highlighted, and translators can easily confirm with the TAB key. Translators can also lock in accurate translations by clicking on them.

Beneath each translation, a suggestion box is featured. It furnishes the next 3-best translations generated by the translation model, excluding the top-ranked one which is already displayed. The box also includes a suggestion derived from a GPT-based LM. This acts as a supplementary reference, offering insights into monolingual fluency and stylistic nuances. However, the precision of a GPT model's next-word prediction depends on the preceding context [5, 11]. Therefore, our LM only provides a suggestion when the target prefix composes of more than ten translated words.

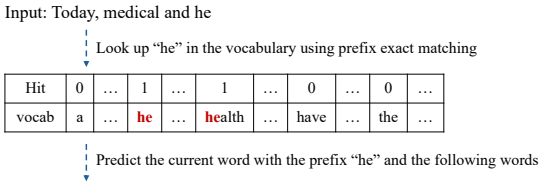

Figure 4: An example of Hit Vector.

Model	**BLEU**	Acc_{1-gram}	Acc_{2-gram}	Acc_{3-gram}
	In-house **IT** domain test set			
In-house NMT	25.47	46.38	30.47	21.13
+Fine-tuning	28.28	51.63	35.64	25.59
adaptive-TM-MT (Ours)	**29.05**	**52.25**	**36.31**	**26.30**
	Open-source **Law** domain test set			
NMT trained with CCMT	37.65	57.26	42.28	32.68
+*simplified*-kNN-MT (Ours)	**42.57**	**59.74**	**46.53**	**37.95**
In-house NMT	33.73	53.66	38.11	28.06
+*simplified*-kNN-MT (Ours)	**37.21**	**56.45**	**42.43**	**33.17**
	Open-source **Subtitles** domain test set			
NMT trained with CCMT	10.43	33.78	18.26	10.47
+*simplified*-kNN-MT (Ours)	**12.12**	**35.84**	**20.93**	**13.07**
In-house NMT	18.81	38.90	21.95	12.97
+*simplified*-kNN-MT (Ours)	**20.04**	**40.61**	**24.30**	**15.34**

Table 1: Translation evaluation results.

2.2.2 ***Subword-Prefix Decoding.*** In post-editing, when the last input from the translator is a space, it indicates the presence of a fully-formed word preceding the space character. In this case, the translation model and the GPT-based LM anticipate the ensuing words through the application of a forced decoding mode. Otherwise, we build a binary vector over the target vocabulary, called **Hit Vector**, to look up the subword prefix in the vocabulary using exact matching. In Hit Vector, any index with a value of 1 represents a match with the subword prefix, denoting a "hit" by the subword prefix. An example is illustrated in Figure 4. Among the words that the subword prefix hits, our model selects the one with the highest generation probability as the current prediction conditioned on all preceding words, and completes the following predictions in the forced decoding mode. This subword-prefix decoding algorithm can be use for both of the translation model and the GPT-based LM.

3 EVALUATION

3.1 Evaluation of Translation Engines

Building on the demonstration of *adaptive*-TM-MT's effectiveness on public datasets [2], we evaluate this approach using an in-house pre-trained Chinese-English neural machine translation (NMT) model based on the Transformer base architecture [17]. This model is fine-tuned with an in-house IT-domain dataset comprising 2.3 million parallel sentences, and either of the validation and test sets contains 2000 pairs. We evaluate online learning on the test set, where the training data serve as TMs for retrievals. In evaluating the *simplified*-kNN-MT, we utilize two NMT models: the in-house

		MT-PE		Synslator		
Week	#Word	Time (h)	Avg.	Time (h)	Avg.	Δ Avg.
1	90,344	147.85	611.05	136.67	661.04	+8%
2	78,882	148.51	531.16	124.97	631.21	+19%
3	44,750	80.05	559.03	71.36	627.10	+12%
Total	213,976	376.41	568.47	333.00	642.57	**+13%**

Table 2: The efficiency of real-time post-editing. #Word and Avg. represent the total # of source words in the projects and the averaged # of words completed per hour.

model and a Transformer base model trained on the CCMT 2022 Chinese-English Corpus.[3] We then employ Chinese-English Law and Subtitles training sets [15] as TMs to create datastores and perform kNN retrievals, and tune hyper-parameters ($K = 4$, $\lambda = 0.4$, $T = 5$) on the validation sets.[4]

We assess translations with the BLEU score using "multi-bleu.perl" of Moses,[5] and propose a novel metric, called N-gram Accuracy, to evaluate prediction accuracy given target prefix inputs. In details, we enumerate target prefix sequences from the golden references, which are assumed as inputs from human translators and enable the model to produce predictions for the subsequent N words. N-gram Accuracy is computed by determining the proportion of correct N-gram predictions relative to the total count of N-gram references, i.e.,

$$Acc_{N-gram} = \frac{Count(Pred_{N-gram} = Ref_{N-gram})}{Count(Ref_{N-gram})}, \quad (1)$$

where $Pred_{N-gram}$ and Ref_{N-gram} are the N-gram prediction and the reference given the target prefix input. A higher value of the N-gram Accuracy indicates better performance.

As shown in Table 1, our experimental findings demonstrate that both the *adaptive*-TM-MT and *simplified-k*NN-MT models are capable of online learning. Comparing the *adaptive*-TM-MT against the pre-trained and the fine-tuned NMT models, we observe that the *adaptive*-TM-MT exhibits superior performance. Moreover, the *simplified-k*NN-MT significantly surpasses both of the public and in-house NMT models in the Law and Subtitle domains.

3.2 Evaluation of Interactive Functionalities

We also evaluate the real-time efficiency of interactive features, including the sub-word prefix decoding and the Suggestion Box, through real-time post-editing experiments. Ten translators with eight years of experience in Chinese-English IT translations are divided into two groups for a three-week IT-domain translation project, using *adaptive*-TM-MT for online learning. Projects are randomly assigned. One group, the MT-PE group, used static post-editing with TMs and termbases, while the other used Synslator's interactive functionalities. The results, summarized in Table 2, show a **13%** improvement in post-editing efficiency for 213,976 words translated with Synslator.

[3]https://www.statmt.org/wmt22/translation-task.html
[4]Sizes of the validation and test sets are 2000, 456 for Law, and 2000, 597 for Subtitles.
[5]http://statmt.org/moses

4 CONCLUSION

We present Synslator, a user-friendly IMT tool. In different deployment environments, it utilizes distinct translation models for online learning with real-time translation memories, and provides multiple translation suggestions through a subword-prefix decoding algorithm. In practical applications, Synslator assists human translators to perform efficient post-editing interactively, enhancing the overall translation workflow.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. *arXiv preprint arXiv:1903.00058* (2019).
[3] Lynne Bowker. 2014. Computer-aided translation: Translator training. In *Routledge encyclopedia of translation technology*. Routledge, 126–142.
[4] Yuhan Dai, Zhirui Zhang, Qiuzhi Liu, Qu Cui, Weihua Li, Yichao Du, and Tong Xu. 2023. Simple and Scalable Nearest Neighbor Machine Translation. *arXiv preprint arXiv:2302.12188* (2023).
[5] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30 (2020), 681–694.
[6] Spence Green, Jason Chuang, Jeffrey Heer, and Christopher D Manning. 2014. Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 177–187.
[7] Spence Green, Jeffrey Heer, and Christopher D Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 439–448.
[8] Spence Green, Jeffrey Heer, and Christopher D Manning. 2015. Natural language translation at the intersection of AI and HCI. *Commun. ACM* 58, 9 (2015), 46–53.
[9] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710* (2020).
[10] Rebecca Knowles and Philipp Koehn. 2016. Neural Interactive Translation Prediction. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*. The Association for Machine Translation in the Americas, Austin, TX, USA, 107–120. https://aclanthology.org/2016.amta-researchers.9
[11] Klemens Lagler, Michael Schindelegger, Johannes Böhm, Hana Krásná, and Tobias Nilsson. 2013. GPT2: Empirical slant delay model for radio space geodetic techniques. *Geophysical research letters* 40, 6 (2013), 1069–1073.
[12] Samuel Läubli, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, Martin Volk, Sharon O'Brien, Michel Simard, and Lucia Specia. 2013. Assessing post-editing efficiency in a realistic translation environment. (2013).
[13] Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: Interactive neural machine translation prediction. 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): System Demonstrations*. 103–108.
[14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.
[15] Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. 2014. UM-Corpus: A Large English-Chinese Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 1837–1842. http://www.lrec-conf.org/proceedings/lrec2014/pdf/774_Paper.pdf
[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.
[18] Ke Wang, Xin Ge, Yuqi Zhang, Yu Zhao, and Jiayi Wang. 2022. Easy Guided Decoding in Providing Suggestions for Interactive Machine Translation. *arXiv preprint arXiv:2211.07093* (2022).
[19] 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*. Association for Computational Linguistics, 1570–1579.