Unbabel-IST at the WMT Chat Translation Shared Task

João Alves[*1], **Pedro Henrique Martins**[*1,3,4],
José G. C. de Souza[1], **M. Amin Farajian**[1], **André F. T. Martins**[1,3,4]
[1]Unbabel, Lisbon, Portugal, [2]INESC-ID, Lisbon, Portugal
[3]Instituto de Telecomunicações, Lisbon, Portugal
[4]Instituto Superior Técnico, University of Lisbon, Portugal

Abstract

We present the joint contribution of IST and Unbabel to the WMT 2022 Chat Translation Shared Task. We participated in all six language directions (English ↔ German, English ↔ French, English ↔ Brazilian Portuguese). We addressed the lack of domain-specific data with a lightweight adaptation approach, using mBART50, a large pretrained language model trained on millions of sentence-pairs, as our base model. We fine-tune it using a two-step fine-tuning process. In the first step, we fine-tune the model on publicly available data. In the second step, we use the validation set. After having a domain-specific model, we explore the use of kNN-MT as a way of incorporating domain-specific data at decoding time.[1]

1 Introduction

In recent years, neural machine translation (NMT) has seen remarkable advances due to the increasingly powerful models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). The translation of conversational text is an important and challenging application for machine translation, specially in the customer support domain, since international companies have an increasing need to offer customer support in various languages. However, this domain has not been substantially explored in machine translation research.

In the Chat Translation shared task, the goal is to understand the context's impact in conversational text translation, and to study the feasibility of multilingual systems for customer support translation. This year, the focus was on the case in which we have a centralizing costumer support with English speaking agents and a translation layer between agent and costumer, allowing the communication with customers which speak different languages.

In this paper we discuss our submission to this task. Our submitted system covers all 3 language pairs: English-German, English-Brazilian Portuguese, and English-French, in both directions: we translate the agent utterance from English to the other language and the customer utterances from the other language to English. As no training data is provided for this task, we recur to the use of the pre-trained multilingual machine translation model mBART50 (§2.1; Tang et al. (2020)) and perform domain adaptation through fine-tuning (§2.2) with domain-specific data and by retrieving similar examples from domain-specific datastores (§2.3). To increase the size of training examples that can be used to fine-tune the model and to create the domain-specific datastore we search for similar examples on publicly available datasets (§3.1) and perform back-translation of the provided monolingual data (§3.2).

2 Models

In this section, we describe the model that we used to tackle this shared task. We start by describing the base model. Then, we describe the techniques used to adapt the base model to customer support chat translation.

2.1 Base Model

As our base model, we use the mBART50 (Tang et al., 2020) "one-to-many" (English to 49 other languages) or "many-to-one" (49 languages to English), depending on the language direction. mBART50 can translate sentences between English and 49 different languages, which include the languages present in this shared task (German, French and Brazilian Portuguese). It consists of a pre-trained encoder-decoder transformer that is first pretrained on a auto-denoising task with monolingual data from 25 languages (mBART; Liu et al. (2020)) and then further pre-trained on an extended set of monolingual data that comprises

[*]Equal contribution.
[1]The code was based on: https://github.com/deep-spin/efficient_kNN_MT.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 943–948
Abu Dhabi, December 7–8, 2022.

50 languages. Then, to adapt the model to perform machine translation, Tang et al. (2020) performed multilingual fine-tuning on machine translation, using data from the 50 supported languages. For this, they used three different configurations: "one-to-many", "many-to-one", and "many-to-many". The first two are obtained by fine-tuning the model with the bilingual data, having English as the source or target language, respectively. The latter is obtained by fine-tuning the model with all the language pairs combinations (using English as the pivot language to obtain the bilingual data).

2.2 Fine-tuning

We performed a two-step fine-tuning process. First, we fine-tuned mBART50 on the domain-specific data that was obtained using data augmentation (§3.1). Then we performed a second step of fine-tuning using the validation sets provided by the shared task organization.

2.3 Nearest Neighbor Machine Translation

To further adapt mBART50, we use the nearest neighbor machine translation approach, kNN-MT, introduced by Khandelwal et al. (2021). kNN-MT consists of a semi-parametric model: besides having a parametric component (base model) that outputs a probability distribution over the vocabulary, $p_{\text{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$, it also has a nearest neighbor retrieval mechanism, which allows direct access to a datastore of examples.

More specifically, we build a datastore $\mathcal{D}$ which consists of a key-value memory, where each entry key is the decoder's output representation, $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t}) \in \mathbb{R}^d$, and the value is the corresponding target token y_t:

$$\mathcal{D} = \{(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t}), y_t) \; \forall \; t \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}\}, \quad (1)$$

where $\mathcal{S}$ denotes a set of parallel sentences.

Then, at inference time, the model searches the datastore to retrieve the set of k nearest neighbors $\mathcal{N}$. Using their distances $d(\cdot)$ to the current decoder's output representation, we can compute the retrieval distribution $p_{k\text{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ by applying the softmax function:

$$p_{k\text{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = \quad (2)$$
$$\frac{\sum_{(\boldsymbol{k}_j, v_j) \in \mathcal{N}} \mathbb{1}_{y_t = v_j} \exp\left(-d\left(\boldsymbol{k}_j, \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t})\right)/T\right)}{\sum_{(\boldsymbol{k}_j, v_j) \in \mathcal{N}} \exp\left(-d\left(\boldsymbol{k}_j, \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t})\right)/T\right)},$$

where T is the softmax temperature, k_j denotes the key of the j^{th} neighbor and v_j its value. Finally, the two probability distributions, $p_{\text{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ and $p_{k\text{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$, are combined to obtain the final distribution, which is used to generate the translation through beam search, by performing interpolation:

$$p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = (1 - \lambda)\, p_{\text{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \quad (3)$$
$$+ \lambda\, p_{k\text{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}),$$

where $\lambda \in [0, 1]$ is a hyper-parameter that controls the weights given to the two distributions.

2.3.1 Using Two Datastores

As we use data from multiple sources (described in Section 3), we adapt kNN-MT to perform retrieval from two datastores which are composed of examples from different datasets. To do this, we simply need to perform retrieval from the two datastores obtaining two retrieval distributions, $p_{k\text{NN}_1}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ and $p_{k\text{NN}_2}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$, computed using Eq. 2.

Then, we need to modify the distributions interpolation (Eq. 3) to account for three distributions:

$$p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = (1 - \lambda_1 - \lambda_2)\, p_{\text{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \quad (4)$$
$$+ \lambda_1\, p_{k\text{NN}_1}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$$
$$+ \lambda_2\, p_{k\text{NN}_2}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}),$$

where $\lambda_1 \in [0, 1]$ and $\lambda_2 \in [0, 1]$ are hyper-parameters that control the weights given to the three distributions.

3 Data

The data provided by the shared task organization is part of a corpus called MAIA corpus. It consists of parallel data of chats between an agent (English) and a customer (Brazilian Portuguese, German or French) across one domain: customer support conversation. Thus, there are a total of 6 translation directions. One of the main obstacles of this domain is the lack of parallel data publicly available. To make the task closer to a real case scenario, the shared task organization has only provided bilingual validation sets and monolingual data, for all languages.

As already mentioned, finding parallel data for this specific domain is challenging. The only exception is the dataset from WMT 2020 Shared Task on Chat Translation (Farajian et al., 2020). Unfortunately, it only contains two translation directions:

Language Direction	Original dev set	New training set	New dev set
en-de	1006	528	478
en-fr	1750	894	856
en-pt_br	1353	668	685
de-en	1103	519	584
fr-en	1003	466	537
pt_br-en	1006	469	537

Table 1: Statistics (number of sentences) of the development sets provided by the shared task organization, and of the new development and training sets after splitting it in two.

Language Direction	Number of Sentences
de-en	203,169,413
fr-en	471,885,306
pt_br-en	192,874,694

Table 2: Statistics (number of sentences) of the public available data in OPUS.

Language Direction	Number of Sentences
en-de	6494
en-fr	3311
en-pt_br	2010
de-en	6874
fr-en	1929
pt_br-en	1657

Table 3: Statistics (number of sentences) of the back-translated data.

English to German and German to English. Therefore, to circumvent the lack of domain-specific data available to fine-tune the model and to add to the datastores, we perform data augmentation (§3.1) and back-translate the monolingual data provided (§3.2).

3.1 Data Augmentation

As the domain-specific data available is limited to the provided bilingual development sets and monolingual sets, we perform data augmentation to create training sets. To do so, we use LaBSE (Language-Agnostic BERT Sentence Embedding) (Feng et al., 2020) multilingual sentence representations. In order to perform data augmentation we also use the k-nearest neighbours (kNN) implementation of the FAISS toolkit (Johnson et al., 2019).

We start by defining a seed corpus (which in our case is the validation set) and a pool corpus (generic data). Then, we use LaBSE to compute the sentence embeddings. After having the sentence embeddings, we built an in-house kNN implementation that relies on FAISS to compute the similarity among all sentences, obtaining a score between 0 (no similarity) and 1 (maximum similarity). Then, we keep the sentence-pairs with a score higher than 0.7.

3.1.1 Data Selection

Regarding data selection, we use all possible datasets publicly available in OPUS (Tiedemann, 2012) to create our pool of public data. Statistics can be find in Table 2.

3.1.2 Data Cleaning

After having downloaded all data, we perform data cleaning. To do so, we used a combination of heuristic filters and Bicleaner (Sánchez-Cartagena et al.; Ramírez-Sánchez et al., 2020). Bicleaner is a tool that detects noisy sentence-pair in a parallel corpus. It outputs the likelihood of two sentences being a mutual translation (in this case the value is near 1) or not (the value is near 0). We could have trained our own Bicleaner models but we decided to use the available ready-to-use language packages.

3.2 Back-translation

To increase the amount of domain-specific data, we also use the monolingual data provided by the shared task organizers. To do so, we performed back-translation of these datasets with best fine-tuned model using beam-search with 5 beams. The statistics are reported in Table 3. To perform back-translation we used ours models fine-tuned. We use the back-translated examples both for fine-tuning our models and as part of the datastores.

4 Experiments

In this section, we describe the experiments we made, to allow us to choose the best model to submit to the shared task.

4.1 Experimental Settings

The shared task organization provided two different baselines: one leveraging the conversation context

Model	Language Direction					
	en-de		en-fr		en-pt_br	
	SacreBLEU	COMET	SacreBLEU	COMET	SacreBLEU	COMET
Baseline (without context)	35.11	0.3989	54.23	0.8011	50.35	0.7897
Baseline (with context)	33.75	0.3755	53.95	0.8013	51.02	0.8721
kNN-MT	52.20	0.5873	61.20	0.9032	48.80	0.9398
Fine-tuned Model	62.50	0.7289	71.60	1.0485	67.80	1.1285
Fine-tuned Model + kNN-MT (1 datastore)	**62.70**	**0.7351**	71.60	1.0324	68.10	1.1330
Fine-tuned Model + kNN-MT (2 datastores)	61.30	0.7334	**72.00**	**1.0495**	**68.10**	**1.1356**

Table 4: Results obtained for the agent direction (en -> X).

Model	Language Direction					
	de-en		fr-en		pt_br-en	
	SacreBLEU	COMET	SacreBLEU	COMET	SacreBLEU	COMET
Baseline (without context)	45.75	0.5421	47.12	0.6413	44.52	0.5887
Baseline (with context)	47.13	0.6253	48.25	0.6855	47.29	0.6475
kNN-MT	57.70	0.8617	52.70	0.8390	50.90	0.7984
Fine-tuned Model	**59.40**	0.8811	**57.70**	**0.9250**	50.10	0.8117
Fine-tuned Model + kNN-MT (1 datastore)	59.20	0.8760	57.40	0.9277	50.90	0.7984
Fine-tuned Model + kNN-MT (2 datastores)	58.70	**0.8814**	57.20	0.9226	**51.80**	**0.8009**

Table 5: Results obtained for the customer direction (X -> en).

and another one that does not. Both of them use the M2M-100 (Fan et al., 2020) large pre-trained language model, which is originally a sentence-level model. Together with the baselines, the shared task organizers provided scripts to rerun the experiments using conversational context, which we did for our small test set.

As no training data was provided by the organization, we splitted the validation set into two. We used one of them as our validation set and the other was used to fine-tune the models and to perform kNN-MT. We report the data sets statistics in Table 1. We took into consideration the fact that we are dealing with conversations, and thus, we do not split conversations, i.e., we do not perform segment filtering that might break a conversation context.

We implemented all the models by the open-sourced toolkit *fairseq* (Ott et al., 2019).

Although mBART50 supports multilingual training, we trained each language direction separately. We started by fine-tuning mBART50 with the data obtained with the data augmentation process (§3.1), the data from WMT2020 Chat Translation shared task, and the back-translated monolingual data (§3.2). Then, we continued the fine-tuning step using the the training set of the shared task.

To perform retrieval we use 2 datastores having the first datastore the data from the validation sets and the second one the data obtained with the data augmentation process (§3.1) and the back-translated monolingual data (§3.2).

Hyper-Parameter	Value
Learning Rate	0.00003
Warmup updates	16000
Label Smoothing	0.2
Optimizer	Adam
β_1, β_2	0.9, 0.98
Weight Decay	0.1
Dropout	0.1
Clip Norm	5
Batch Size	256 (tokens)
Beam Size	5
kNN-MT k	8
kNN-MT temperature	10
kNN-MT λ_1	0.1
kNN-MT λ_2	0.1

Table 6: Fairseq Hyperparameters for our experiments. The first block gives the base settings used for fine-tuning mBART50 and the second block provides the details for the kNN-MT.

The selected values for hyperparameters are stated in Table 6. To evaluate the performance of our models we used SacreBLEU (Post, 2018) and COMET (Rei et al., 2020).

4.2 Results

We tested multiple configurations for kNN-MT: using only one datastore with the validation data or using two datastores with different values for the

parameters that control the weight given to each distribution (λ_1 and λ_2), changing the number of neighbours retrieved, and the softmax temperature.

The results reported in Tables 4 and 5 show that performing fine-tuning of mBART50 on domain-specific data leads to large gains for all language pairs, for the two metrics. We can also see that, despite leading to worse scores than fine-tuning, simply retrieving examples from domain-specific datastores, using kNN-MT, leads to considerable gains when comparing with the baselines. Moreover, using kNN-MT with the fine-tuned model as the base model, leads to small gains on most language pairs, for the agent direction (English→X). For the customer direction (X→English), the results are very similar to the ones obtained without retrieval. When comparing with using 1 datastore (only with the validation data), using 2 datastores leads to small improvements, which suggests that the gains led by performing retrieval are due to the data coming from the validation sets.

In terms of speed, kNN-MT model requires retrieval for every single token, leading to a low decoding speed, around 8 times slower than a model that does not perform retrieval steps according to (Martins et al., 2022). Although, it is important to take into consideration that the time the model takes to add examples to the datastores is much shorter than the time needed to fine-tune the model.

Due to the repetitive nature of dialogues in customer service conversational content, we can see that by using only a few thousand domain-specific bilingual sentence-pairs together with out-of-domain sentence-pairs (selected using the data augmentation process), we are able to improve the performance of the baselines by a large margin. By analysing these experiments' results, we selected the model that combines fine-tuning and kNN-MT (with 2 datastores) as our primary submission. For the submission, we performed fine-tuning again using the complete development sets, and also added the entire development sets to the kNN-MT datastores.

5 Conclusions

We presented the joint contribution of IST and Unbabel to the WMT 2022 Chat Translation shared task. First, we perfomed fine-tuning of a large pretrained model, mBART50. Then we perfomed kNN-MT using multiple datastores to incorporate domain-specific data at decoding time. Through experiments we show that the combination of the proposed methods is a good way of performing domain adaptation when we have few domain-specific data available.

As we are dealing with conversational content it would be interesting to incorporate context information. Unfortunately the few experiments that we have performed using context did not improve the performance of our models. As future work, one interesting line of research is how to incorporate the context information together with augmented retrieval approaches. These can be complementary to each other leading to translation quality improvements.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 65–75, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proc. ICLR*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022. Chunk-based nearest neighbor machine translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael

Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

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

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NeurIPS*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual Translation with Extensible Multilingual Pretraining and Finetuning.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

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 *Proc. NeurIPS*.