SUMTRA: A Differentiable Pipeline for Few-Shot Cross-Lingual Summarization

Anonymous ACL submission

Abstract

Cross-lingual summarization (XLS) addresses summary generation in a language different from that of the input document (e.g., English to Spanish). In the present day, the predominant approach to this task is to take a performing, pretrained multilingual language model (LM) and fine-tune it for XLS on the language pairs of interest. However, the scarcity of fine-tuning resources makes this approach non-viable in some cases. For this reason, in this paper we propose revisiting the *summarize-and-translate* pipeline, where the summarization and translation tasks are performed in a sequence. This approach allows reusing the many, publicly-available resources for monolingual summarization and translation, obtaining a very competitive zero-shot performance. In addition, the proposed pipeline is completely differentiable end-to-end, allowing it to take advantage of few-shot fine-tuning, where available. Experiments over two contemporary and widely adopted XLS datasets (CrossSum and WikiLingua) have shown the remarkable zero-shot performance of the proposed approach, and also its strong few-shot performance compared to an equivalent multilingual LM baseline, where the proposed approach has been able to outperform the baseline in many languages with only 0.1X shots.

1 Introduction

Cross-lingual summarization (XLS) aims to take a document in a source language and generate a summary in a different language, providing humans with the ability to efficiently understand documents in foreign languages. However, XLS is a challenging task due to the limited training data available. Unlike for monolingual summarization, naturally-occurring cross-lingual document-summary pairs are rare, and dedicated XLS human annotation is demanding since it requires uncommon skills of the annotators (Wang et al., 2022). This has often led to the scraping of existing multilingual data to be later aligned for cross-lingual use (Ladhak et al., 2020; Bhattacharjee et al., 2022).

Given the constraints in dedicated training resources, most recent approaches have focused on exploiting available multilingual LMs (Liu et al., 2020; Tang et al., 2021; Xue et al., 2021) pretrained in the typical unsupervised manner over large, monolingual corpora in multiple languages, and fine-tuning them with the limited XLS resources available in the targeted language pairs (Perez-Beltrachini and Lapata, 2021; Ma et al., 2021). However, these multilingual models suffer from well-known limitations. On the one hand, the uneven pretraining of multilingual LMs often results in poor knowledge transfer to low-resource languages (Joshi et al., 2020; Bhattacharjee et al., 2022). On the other hand, the superposition of too many languages in a single model can result in a degradation of cross-lingual performance in the downstream task (a.k.a. language interference) (Pfeiffer et al., 2022). In addition, it is not trivial to reuse the abundant, existing monolingual summarization data, since fine-tuning a multilingual LM with monolingual data often compromises its ability to generate text in a language different from the input's (Vu et al., 2022; Bhattacharjee et al., 2022)—a problem known as "catastrophic forgetting" (van de Ven and Tolias, 2019). The above issues compound in the impossibility of achieving a satisfactory zero-shot XLS performance out of conventional multilingual LMs.

For this reason, this work revisits the *summarize-and-translate* approach to XLS (Wan et al., 2010), with the main aim of fully leveraging the existing monolingual summarization resources (i.e., training data, pretrained models) to obtain a performing zero-shot XLS pipeline. Specifically, we propose combining 1) a monolingual summarizer trained with abundant resources in the source language with 2) a pretrained machine translation model that translates into the target language. If the quality of both models is high, such a pipeline should be able to achieve a significant zero-shot performance. Yet, it can also suffer from model misalignment and error propagation. Therefore, we modify the summarizer to output "soft" predictions, ensuring that the pipeline remains fully differentiable end-to-end (Jauregi Unanue et al., 2023). This allows fine-tuning it to improve the coupling of the models, alleviate error propagation, and obtain summaries that are closer to the ideal, joint summarization/translation of the XLS task. For immediacy, we refer to the proposed pipeline as SumTra.

In particular, in this paper we focus on the less explored *English-to-many* XLS task (most work to date has focused on many-to-English (Zhu et al., 2019; Ladhak et al., 2020; Ma et al., 2021; Chi et al., 2021) or specific language pairs such as English-to-Chinese (Ayana et al., 2018; Zhu et al., 2019; Bai et al., 2021; Liang et al., 2022)). We believe that this is a valuable contribution as it provides access to summaries of the multitude of existing English documents for speakers of other languages around the world. To this aim, we have carried out experiments over two widely used XLS datasets (CrossSum (Bhattacharjee et al., 2022) and WikiLingua (Ladhak et al., 2020)), with a range of language pairs spanning high-, medium-, and low-resource languages. The results show a strong quantitative performance for the zero-shot pipeline, and a competitive edge over a comparable multilingual language model baseline with up to 100-shot fine-tuning[1].

Overall, our paper makes the following contributions:

- A *summarize-and-translate* pipeline that leverages contemporary state-of-the-art language models (and their resources) for the summarization and translation steps.
- A fully differentiable approach through the use of "soft" summaries, making the pipeline fine-tunable end-to-end.
- A novel objective function that incorporates a back-translation loss over the summarization module to ground the generation of the intermediate summaries to the target language reference.
- A comparative experimental evaluation of the proposed approach over two popular cross-lingual summarization datasets spanning two diverse domains, including an extensive qualitative, ablation, and sensitivity analysis.

2 Related Work

Cross-lingual summarization (XLS) has been an active research topic for a long time (Leuski et al., 2003; Wan et al., 2010). Pre-neural methods have often combined monolingual summarization and machine translation (MT) modules into pipeline approaches that *summarize-and-translate* (Orăsan and Chiorean, 2008; Wan et al., 2010), or *translate-and-summarize* (Leuski et al., 2003; Wan, 2011; Boudin et al., 2011). While conceptually defensible, these approaches inevitably suffer from error propagation between the modules, and, obviously, the architectural limitations of the models of the day (Zhu et al., 2019; Ouyang et al., 2019) .

With the recent development of multilingual pre-trained language models such as mBART (Lewis et al., 2020) and mT5 (Xue et al., 2021), there has been a surge in XLS research that has focused on fine-tuning these models with XLS datasets, and as a consequence has relegated pipeline methods to be regarded as mere baselines for comparison (Ladhak et al., 2020; Perez-Beltrachini and Lapata, 2021). However, the current approaches are not exempt from performance limitations at their turn, in particular when applied to low-resource languages[2]. To address them, Bhattacharjee et al. (2022) has attempted to transfer knowledge from high- to low-resource languages by a multi-stage sampling algorithm that aptly up-samples the low-resource languages. Other works have explored using language-specific adapter modules in various cross-lingual tasks (Rebuffi et al., 2017; Houlsby et al., 2019) to increase the linguistic capacity of the model at a parity of trainable parameters and alleviate language interference (Pfeiffer et al., 2022). Bai et al. (2021) have proposed using a combination of monolingual and cross-lingual summarization in an attempt to improve performance on low-resource languages. However, none of these approaches has specifically focused on the zero- and few-shot scenario that we canvass in our paper.

[1] Our anonymized code is publicly accessible at: https://anonymous.4open.science/r/sumtra-6490/

[2] We note that in the XLS task there are many dimensions in which a language can be "low-resource", namely: the monolingual data for model pretraining; the parallel corpora for translation pretraining; and the annotated XLS document-summary pairs.

3 SumTra

The proposed SUMTRA model consists of the cascade of two language models: a monolingual summarization language model, followed by a machine translation language model, which we refer to as SUM and TRA for *summarize* and *translate*, respectively.

Let us denote the token sequence of the input document as $x = \{x_1, \dots x_n\}$, and the token predicted by the SUM module at slot j as s_j. We can then express the sequence of probability vectors output by the SUM module over the vocabulary as $\{\mathbf{p}_1, ...\mathbf{p}_j..., \mathbf{p}_m\}$, with:

$$\mathbf{p}_j = \textsc{Sum}(s_{j-1}, x, \theta) \tag{1}$$

where s_{j-1} is the previous predicted token and θ are the module's parameters. For simplicity and efficiency we use greedy search for token prediction, but in principle any decoding approach can be used.

The probability vectors $\{\mathbf{p}_1, ...\mathbf{p}_j..., \mathbf{p}_m\}$ are then individually mixed with the embedding layer $\mathbf{E}$ of the TRA module of size $D \times V$ (embedding $\times$ vocabulary) to obtain a sequence of expected embeddings, $\mathbf{e} = \{\mathbf{e}_1...\mathbf{e}_j...\mathbf{e}_m\}$, with:

$$\mathbf{e}_j = \mathbb{E}[\mathbf{E}]_{\mathbf{p}_j} = \mathbf{E}\ \mathbf{p}_j \tag{2}$$

which are equivalent to "soft" predictions from the SUM module. These expected embeddings, which represent the intermediate summary, are then provided as input to the TRA module bypassing its embedding layer. Eventually, the TRA module predicts the translation in the target language:

$$\bar{y} = \textsc{Tra}(\mathbf{e}, \sigma) \tag{3}$$

where $\bar{y}$ denotes the translation and σ the module's parameters. Since the soft predictions from the SUM module do not interrupt backpropagation, the whole network can be trained end-to-end.

For fine-tuning the entire SUMTRA model, we use the standard negative log-likelihood:

$$\text{NLL} = -\sum_{t=1}^{T} \log p(y_t | y_1, \dots y_{t-1}, \mathbf{e}, \theta, \sigma) \tag{4}$$

where with $\{y_1, \dots y_T\}$ we denote the sequence of ground-truth tokens in the target language, and with $p(y)$ the probabilities output by the translator. However, fine-tuning the SUM module with only backpropagation from this training objective, combined with the inherently large generation space of summarization, tends to lead to summaries that abstract from the target language ground-truth. For this reason, we add an auxiliary training objective that encourages the predicted summary to adhere to the target more closely. To this aim, we first back-translate the ground-truth sequence, y, into the language of the summarizer (i.e., English) using a reverse TRA module, and then use it as auxiliary training objective for the summarizer:

$$\text{NLL}_{\text{SUM}} = -\sum_{t=1}^{T} \log p(\hat{y}_t | \hat{y}_1, \dots \hat{y}_{t-1}, x, \theta) \tag{5}$$

where $\hat{y}$ denotes the back-translated sequence, and $p(\hat{y})$ the probabilities output by the summarizer. It is interesting to note that our use of a separate summarization module would allow us to also use other typical summarization training objectives such as sentence-level coherence (Li et al., 2019), coverage of the input document (Parnell et al., 2022) and so forth, but we leave this to future work.

The training objectives in Equations 4 and 5, are eventually combined in a simple convex combination:

$$L = \alpha \text{NLL}_{\text{SUM}} + (1 - \alpha)\text{NLL} \tag{6}$$

using a scaling coefficient α that acts as a hyperparameter in the loss. A sensitivity analysis is presented in Section 5.4.

4 Experimental Setup

4.1 Datasets, Baselines, and Evaluation Metrics

We have carried out extensive zero and few-shot experiments over twelve English-to-many language pairs from the CrossSum (Bhattacharjee et al., 2022) and WikiLingua (Ladhak et al., 2020) datasets. We have selected six languages from each dataset, and labelled them as high, medium, and low-resource based on the number of sentences used for the pretraining of the respective language in mBART-50 (Tang et al., 2021) that we have used as our main baseline. Languages with >1M pretraining sentences have been labelled as high-resource, between 100k and 1M as medium-resource, and <100K as low-resource[3].

As strong multilingual baselines, we employ the mT5-m2m model of Bhattacharjee et al. (2022),

[3] As per Table 6 of Tang et al. (2021).

Model	High		Medium		Low		**Average**
	en-es†	en-fr†	en-ar†	en-uk	en-az	en-bn†	
mBART-50 (0-shot)	1.18 / 26.46	0.26 / 21.14	0.85 / 33.62	0.00 / 28.96	0.11 / 19.79	0.00 / 25.83	0.40 / 25.97
mBART-50 (50-shot)	1.18 / 26.54	0.26 / 21.06	1.27 / 36.14	0.00 / 28.96	0.17 / 20.56	0.00 / 25.00	0.48 / 26.38
mBART-50 (100-shot)	1.18 / 26.50	14.53 / 48.42	1.28 / 36.20	4.46 / 54.69	0.17 / 20.57	0.81 / 39.70	3.74 / 37.68
SUMTRA (0-shot)	20.19 / 55.41	20.87 / 53.98	15.80 / 60.33	8.74 / 59.80	13.28 / 54.09	4.04 / 54.32	13.82 / 56.32
SUMTRA (50-shot)	21.32 / **56.66**	20.03 / 53.46	15.84 / 60.62	8.76 / 59.88	**14.68 / 54.54**	3.90 / 54.85	14.09 / 56.67
SUMTRA (100-shot)	**21.47** / 56.41	**21.24 / 54.06**	**16.08 / 60.67**	**9.47 / 59.98**	13.97 / 54.10	**4.67 / 56.28**	**14.47 / 56.92**
mBART-50 (1000-shot)	18.29 / 53.99	17.57 / 50.76	14.36 / 60.06	7.41 / 58.01	14.32 / 54.74	*7.17 / 60.53*	13.19 / 56.35
mT5-m2m (Bhattacharjee et al., 2022)	*22.23 / 56.86*	19.27 / 52.48	*16.56* / 60.49	8.63 / 59.65	*18.48 / 57.27*	*11.49 / 66.31*	*16.11 / 58.84*

Table 1: Results for the CrossSum dataset, grouped into high, medium, and low-resource languages. We report ROUGE (or mROUGE as denoted with †) and BERTScore. The best scores between mBART-50 and SUMTRA are bolded. The results for mT5-m2m that surpass SUMTRA are italicized.

Model	High		Medium		Low		**Average**
	en-ru†	en-zh†	en-ar†	en-tr†	en-th†	en-id	
mBART-50 (0-shot)	0.57 / 29.54	0.00 / 36.75	0.78 / 33.29	0.91 / 23.08	1.78 / 31.11	0.94 / 26.44	0.83 / 30.04
mBART-50 (50-shot)	0.71 / 30.69	0.00 / 36.75	0.78 / 34.19	1.02 / 23.56	1.71 / 31.04	1.25 / 27.54	0.91 / 30.63
mBART-50 (100-shot)	6.77 / 52.70	0.00 / 36.75	0.79 / 34.09	6.70 / 47.84	0.63 / 31.77	1.25 / 27.32	2.69 / 38.41
SUMTRA (0-shot)	10.35 / 56.12	**21.13** / 57.24	11.61 / 61.48	10.96 / 53.96	14.66 / 51.39	12.83 / 54.84	13.59 / 55.84
SUMTRA (50-shot)	11.73 / 58.33	19.70 / 60.16	**11.74 / 61.79**	11.44 / 54.78	15.83 / 53.04	12.79 / 55.06	13.87 / 57.19
SUMTRA (100-shot)	**12.01 / 58.85**	19.70 / **61.08**	11.58 / 61.66	**12.50 / 55.69**	**16.15 / 54.16**	**13.12 / 55.68**	**14.18 / 57.85**
mBART-50 (1000-shot)	9.43 / 56.49	20.35 / *62.06*	11.11 / 61.74	*15.08 / 56.74*	*19.65 / 61.71*	10.95 / 53.01	*14.43 / 58.63*

Table 2: Results for the WikiLingua dataset, grouped into high, medium, and low-resource languages. We report ROUGE (or mROUGE as denoted with †) and BERTScore. The best scores between mBART-50 and SUMTRA are bolded. The results for mBART-50 (1000-shot) that surpass SUMTRA are italicised.

fine-tuned on all languages and full training splits of the CrossSum dataset, and a pretrained mBART-50 (Tang et al., 2021). For the proposed approach, we have used the mBART-50 one-to-many variant for the TRA module, and the many-to-one variant for both the SUM module and the one-off generation of the back-translations. To evaluate the predictions, we have used ROUGE (Lin, 2004) and its multilingual adaptation[4], mROUGE, which leverages language-specific tokenizers and stemmers (Conneau and Lample, 2019) to pre-process non-English text prior to a standard ROUGE calculation. As common in summarization, we have computed the ROUGE score as average of ROUGE-1, ROUGE-2 and ROUGE-L F1. Similarly to Koto et al. (2021), we also report BERTScore (Zhang et al., 2020) for its greater ability to assess the semantic alignment of the predictions and the references.

4.2 Model Training

Prior to running the XLS experiments, we have trained the SUM module for monolingual summarization in English. To this aim, we have leveraged the English-English training split of either CrossSum or WikiLingua, and chosen the best performing checkpoint based on a validation criterion. We have then performed a set of cross-lingual summarization experiments in zero-shot and few-shot fine-tuning configurations. For the latter, we have chosen to fine-tune the entire SUMTRA model; however, it is also possible to freeze either the summarization or the translation module, and we present an ablation in Section 5.1. Further details of the experimental setup are provided in Appendix A.

5 Results and Analysis

Table 1 presents the main results over the test sets of the chosen language pairs from the CrossSum dataset. In the table, we compare the proposed SUMTRA model with mBART-50 at zero shots and with variable amounts of few-shot XLS fine-tuning (50 and 100 samples). For reference, we also report the performance of mBART-50 with 1000 shots and that of the mT5-m2m model. However, we note that the latter has been fine-tuned over all the language pairs in the CrossSum dataset (1,500+), and with the entire available XLS training set (∼900-1,500 samples per pair) (Bhattacharjee et al., 2022), and should therefore be regarded as a hard-to-near upper bound. The results show that SUMTRA has amply outperformed mBART-50 at a parity of fine-tuning examples in all cases. While this was to be expected to a large extent since mBART-50 is

[4] For brevity, we will refer to "ROUGE" as "mROUGE" throughout, to accommodate all languages.

not designed for zero-shot XLS performance, the zero-shot SumTra model has still surpassed the 100-shot mBART-50 model by more than +10.00 mROUGE pp and almost +20.00 BERTScore pp on average across the six languages. The 50-shot fine-tuning has improved SumTra's performance in all the tested languages, and the 100-shot fine-tuning has improved for five languages out of six, proving the effectiveness of the proposed training objective. In addition, the proposed model has performed very well also vis-à-vis the 1000-shot mBART-50 and mT5-m2m. In particular, our zero-shot model has outperformed the 1000-shot mBART-50 in four languages out of six in mROUGE score, and our 50-shot model has outperformed it in five out of six. While SumTra has not reached the same average scores as the mT5-m2m model even in the 100-shot configuration, it has surpassed it in two languages (French and Ukrainian). These results show that the proposed approach is capable of very strong zero-shot performance, and with a few-shot fine-tuning can reach or near state-of-the-art performance. This can prove particularly useful for languages with a scarcity ($\leq$ 100) of annotated XLS samples.

In turn, Table 2 presents the main results for the WikiLingua dataset. Also for this dataset, the trend for the proposed model and mBART-50 has been similar: the proposed zero-shot model has surpassed the 100-shot mBART-50 in all cases, and by +10.90 mROUGE pp and +19.44 BERTScore pp on average. Very notably, our zero-shot model has also outperformed the 1000-shot mBART-50 in mROUGE score in four of the six languages, showing again that the proposed approach is capable of a strong performance even in the complete absence of annotated XLS data for fine-tuning.

5.1 Module Fine-Tuning

The proposed SumTra model has approximately double the number of parameters of a single mBART-50-large language model. However, this is a rather small model by contemporary standards (611M parameters), and SumTra can comfortably fit in the memory of any standard machine for inference. Conversely, the memory footprint may become an issue for some machines in the case of fine-tuning. For this reason, we have tested the SumTra's performance by fine-tuning only either the summarizer or the translator. This is also to show that a significant performance can still be achieved if memory constraints force the fine-tuning to be carried out at a parity of trainable parameters with mBART-50. To this aim, Figure 1 plots the BERTScore score of the various configurations for Spanish and Bengali at increasing amount of fine-tuning. For both languages, updating only the parameters of the summarizer has led to the smallest improvements over the zero-shot performance. It could be argued that the summarizer has already been well-trained by the monolingual data, and as such its relative margin for improvement is smaller. Conversely, fine-tuning only the translator with 10 shots has achieved a comparable performance to fine-tuning the entire model, and has surpassed it in the case of Bengali with 50 shots. The trend has been the opposite for Spanish, where fine-tuning the translator alone has underperformed the fine-tuning of the entire model. This shows that the translation component can be more sensitive to the specificities of the target language.

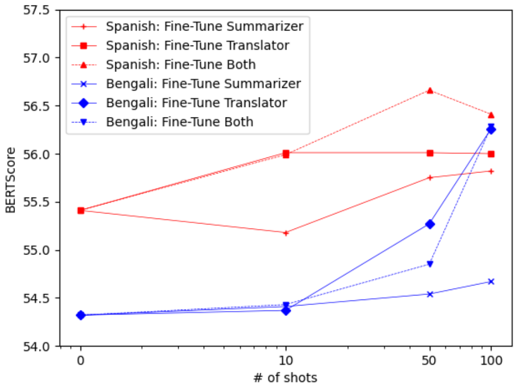

Figure 1: BERTScore scores for the CrossSum Spanish and Bengali test sets with different fine-tuning configurations (summarizer only, translator only, and both).

If memory constraints force the fine-tuning to be carried out at a parity with a single mBART-50 model, several other strategies could be easily put in place, such as alternating between updating the summarizer and the translator in turn, or fine-tuning only selected layers of the modules' encoders and decoders. However, we believe that this is not specially critical and have not explored it further.

5.2 The Catastrophic Forgetting Problem

In the context of multilingual models, the catastrophic forgetting problem refers to the drop in multilingual performance for models that have been trained with monolingual task data (Pfeiffer et al., 2022). Bhattacharjee et al. (2022) have explored

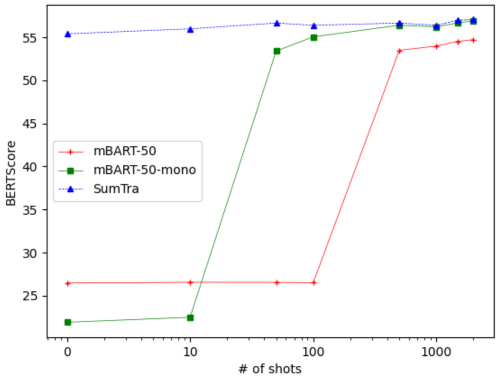

Figure 2: Exploring the catastrophic forgetting problem with mBART-50 and SUMTRA on the CrossSum Spanish test set. In the figure, mBART-50-mono refers to an mBART-50 model first trained with the same monolingual summarization data as SUMTRA.

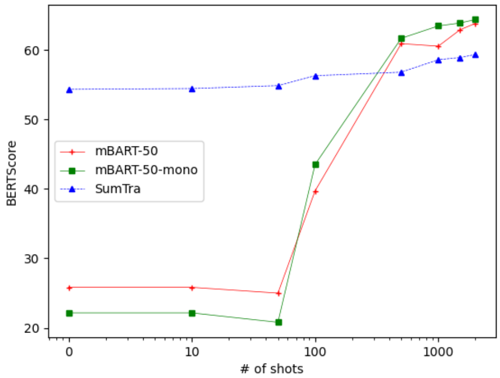

Figure 3: Exploring the catastrophic forgetting problem with mBART-50 and SUMTRA on the CrossSum Bengali test set. In the figure, mBART-50-mono refers to an mBART-50 model first trained with the same monolingual summarization data as SUMTRA.

this within their mT5-m2m model and shown that its zero-shot cross-lingual performance is very poor despite its extensive multilingual pretraining with a multitude of language pairs. Therefore, in this section we set to explore how catastrophic forgetting behaves in the XLS case within a zero-shot, few-shot and unlimited shot scenarios.

To this aim, Figures 2 and 3 compare the performance of mBART-50 with an mBART-50 model (nicknamed mBART-50-mono) trained over the same En-En monolingual summarization data that we used for SUMTRA before fine-tuning. The two plots report the BERTScore score at an increasing number of fine-tuning samples for Spanish and Bengali, respectively. However, for this experiment we have used all the 1241 available fine-tuning samples for Bengali, and 2000 for Spanish. For both languages, it is manifest that SUMTRA is the only model that is capable of a significant zero-shot performance, with a difference of approximately +30 pp compared to both mBART-50 models. At zero-shot and 10-shot, the performance of mBART-50-mono has been even lower than that of the original mBART-50, confirming the catastrophic forgetting. However, from around 100-shots, mBART-50-mono has stably overtaken mBART-50, showing that its "forgotten" multilingual capabilities can be restored with a sufficient amount of fine-tuning. In the case of Spanish, mBART-50-mono has caught up with SUMTRA at 500-shots, and then progressed with a virtually identical performance. Conversely, for Bengali, both mBART-50 models have surpassed SUMTRA at 500-shots and maintained a comparable performance from there. These trends seem very interesting as they show that, while training a cross-lingual model with monolingual data undoubtedly causes a "catastrophic forgetting" of its multilingual capabilities at zero- and few-shots, such capabilities can be restored with a sufficient amount of fine-tuning, and even outperform an equivalent model that has not undergone monolingual training. In the case of Bengali, it also shows that a single language model can outperform our pipeline of two, most likely because it addresses the summarization and translation task in a genuinely "joint" manner. However, it is worth noting that our pipeline can more easily and more directly take advantage of existing summarization and translation resources, as they can be independently used to train its two modules. For instance, in this case we could leverage any other En-Bn parallel corpora to boost the translator's performance. In all cases, we do not target a scenario with unlimited number of fine-tuning data; rather, a zero/few-shot one demanding minimal effort of the annotators.

5.3 Qualitative Analysis

To allow a qualitative appreciation of the generated summaries, Table 3 shows an example for Spanish, comparing an mBART-50-mono model fine-tuned with 1000 shots with SUMTRA fine-tuned with 1/10 of the shots (100). For the latter, we also show the summary generated by the same SUMTRA model fine-tuned without the back-translation (BT) loss of Equation 5. In the table, the sum-

Model	Summary	BERTScore
Reference	Las autoridades estadounidenses amenazaron a la compañía tecnológica Yahoo con ponerle una multa de US$250.000 diarios si el gigante informático no le entregaba datos de usuarios. **Back-Translation:** The US authorities threatened the technology company Yahoo with a daily fine of US$250,000 if the computer giant did not provide it with user data.	
mBART-50-mono (1000-shot)	**Prediction:** El gobierno de Estados Unidos publicó información sobre un caso que ha sacudido a la empresa de informática Yahoo.	55.61
SumTra (100-shot)	**Intermediate Summary:** The US government threatened to impose fines of up to $250,000 (£250,000) if it refused to comply with a court order against Yahoo, according to newly released documents. **Prediction:** El gobierno estadounidense **amenazaba** con imponer multas de hasta 250.000 dólares **(£250,000)** **si se niega a cumplir un** **decreto judicial** **contra Yahoo**, según documentos publicados recientemente.	61.47
SumTra (100-shot) (no BT loss)	**Intermediate Summary:** Yahoo has been fined $250,000 (£250,000) for breaching a US government order to monitor its online services. **Prediction:** Yahoo **ha sido sancionado** con 250.000 dólares **(250.000 libras esterlinas)** por **violar un decreto del gobierno estadounidense** para controlar sus servicios en línea.	54.78

Table 3: Qualitative example for Spanish (CrossSum). **(Red)** denotes incorrect translations or factual inconsistencies, **(Blue)** denotes information from the source document, and **(Green)** refers to matching information in the reference summary.

mary generated by the mBART-50-mono model does contain some information relevant to the reference, such as the relationship between the US authorities and Yahoo. However, it is overall generic and vague. For instance, the specific mention of a "fine of $250,000" in the reference is not conveyed in the prediction. Conversely, the predictions from the SumTra models have both been able to pick up this fact. At its turn, the prediction from the model without the BT loss has incorrectly stated that Yahoo has already been sanctioned (*ha sido sancionado*), while the prediction from the full model has been in general the most informative and accurate. For example, it has been able to include the entity *decreto judicial* (*court order*) that is not present in the reference, but is an important piece of information in the input document (NB: not shown for reasons of space), and also the key term *amenazaba* (*threatened*). The intermediate summary in English shows that this is owed to an effective summarization, which has been carried over faithfully into the Spanish translation. However, it is also clear that the summary generated by the full SumTra model is still imperfect, having predicted £250,000 instead of $250,000. Another example is provided and commented upon in Appendix A.7.

5.4 Sensitivity to the Alpha Hyperparameter

The fine-tuning objective in Equation 6 combines an XLS loss and a back-translation loss with a positive coefficient, α. The back-translation loss only influences the summarizer, while the XLS loss influences the translator directly, and the summarizer via backpropagation through the soft predictions. To explore the sensitivity of the performance to the value of the α coefficient, Table 4 reports the mROUGE and BERTScore scores of the 100-shot SumTra over Spanish and Bengali for increasing α values (i.e., increasing relative influence of the back-translation loss). The results show that in the case of Spanish the best α value has been rather high (0.95), likely because the pretrained translator is already good enough for this language, and the emphasis has been on keeping the summarization aligned with the target. Conversely, in the case of Bengali the relative weight of the XLS loss for the best performance has been much higher (0.50), showing that for this lower-resource language the updates to the translator have proved more important.

For our experiments, we have faced the decision whether to grid-search a best value of α for every language—which would have made our model perform even better—or just use a trade-off value for all languages, which is more practical and con-

α	**Spanish**	**Bengali**
0.00	21.04 / 56.44	4.20 / 55.54
0.50	20.76 / 56.20	**5.21 / 56.38**
0.90	21.30 / 56.46	4.58 / 56.02
0.95	**21.43 / 56.56**	4.25 / 55.65
0.99	21.37 / 56.41	4.67 / 56.28
1.00	19.96 / 55.33	3.81 / 54.61

Table 4: mROUGE and BERTScore scores for different α values in the objective function (CrossSum).

venient for prospective users. In the interest of usability, we have chosen to not over-validate α, selecting a somehow arbitrary fixed value of 0.99 to emphasize the back-translation loss in all cases.

It is perhaps worth remarking once more the respective impact of these two losses on the summarizer: the back-translation loss keeps the predicted summaries more closely aligned with the references, while the XLS loss only influences the summarizer in a "looser" and indirect way via backpropagation of the translator's gradients. Therefore, removing the back-translation loss altogether, or likewise keeping α very low, leads to summaries that still seem qualitatively very effective, but are less faithful to the target. This tends to penalize the scores, especially mROUGE, but did not seem undesirable to us from a qualitative perspective. We leave further exploration and evaluation of this trade-off to future work.

5.5 Inference Time

Given that the proposed model uses two language models in pipeline, we also compare its inference times to those of mBART-50. To this aim, Table 5 reports the inference times per sample[5] of the two models over the test sets of Spanish and Bengali. As to be expected, the proposed model has proved slower on average to generate a prediction; however, less than twice as slow: in the case of Bengali, the inference time per sample has been 1.87x that of mBART-50, and for Spanish only 1.15x. To justify the differences between the two languages, we first note that the inference times for mBART-50 have been nearly identical. In the case of SumTra, the modest overhead with Spanish has been simply due to the addition of the explicit translation stage. In the case of Bengali, the more substantial overhead has been due to the impact of an average lengthening of the predicted intermediate summaries, which has increased both the summarization and the translation times. We ascribe this to the fact that the back-translated summaries used to fine-tune the summarization module have been on average slightly longer than the references, with a corresponding impact on the length of the predicted intermediate summaries and processing times. However, the overall speed seems to have remained acceptable.

Model	**Spanish**	**Bengali**
	Per Sample (s)	Per Sample (s)
mBART-50	**0.146**	**0.145**
SumTra	0.168	0.271

Table 5: Average inference times per sample for mBART-50 and SumTra over the CrossSum Spanish and Bengali test sets (zero-shot fine-tuning configuration).

6 Conclusion

In this paper, we have proposed SumTra, an XLS model that revisits the traditional summarize-and-translate approach into a more contemporary end-to-end differentiable pipeline. Given that genuine XLS annotation is demanding, the main aim of the proposed model is to provide a competitive zero- and few-shot performance. In the paper, we have tested the proposed approach over two mainstream XLS datasets, comparing it with a strong multilingual baseline (mBART-50) and a state-of-the-art model (mT5-m2m). The model's zero-shot performance has been very strong, and also its 100-shot performance has been higher than that of the 1000-shot baseline for the majority of the languages. Through various sensitivity, ablation, and qualitative analyses we have shown that the proposed model benefits from the possibility to separately train its component modules, and that its memory and inference time overheads compared to the baseline are both manageable. In the future, we aim to test model configurations with different base language models for the summarization and translation modules, and explore alternative fine-tuning strategies such as adversarial training and reinforcement learning.

Limitations

The proposed approach has several limitations. The most immediate is that we have limited our experimental validation to the English-to-many case.

[5] We have measured the inference time as the time taken to traverse the model's `generate` function, which occurs twice per sample in SumTra and once in mBART-50. All other overheads are negligible.

However, this was done only for the simplicity of carrying out a one-to-many set of experiments rather than a many-to-many. Instead, the intrinsic limitation of the proposed approach is that it relies on a separate, strong performance from both its summarization and translation modules. In turn, this assumes the availability of an adequate monolingual summarization training set for the source language, and an adequate parallel training corpus for the language pair—or equivalent pretrained models. However, both these requirements are much more easily met than requiring the availability of large XLS annotated resources.

The memory footprint of the proposed model, that has 1.2B total parameters, is also more imposing than that of a conventional multilingual model. In particular, the memory required during fine-tuning has been approximately 34 GB (with the selected hyperparameters). However, in Section 5.1 we have shown that it is possible to fine-tune only one of the two modules in turn (either the summarizer or the translator) and still retain a remarkable performance, bringing back the memory requirements to those of a standard multilingual model.

Finally, the computation of the expected embeddings in Equation 2 requires the product of token embeddings from the translator with the probabilities assigned to those same tokens by the summarizer. This implies that the summarizer and the translator have to share the same vocabulary, and for this reason we have built them both out of the same base model (mBART-50-large). However, it should be easy to organize a redistribution of the summarizer's probabilities over a different vocabulary, allowing mixing different base models.

References

Ayana, Shi-qi Shen, Yun Chen, Cheng Yang, Zhiyuan Liu, and Mao-song Sun. 2018. Zero-shot cross-lingual neural headline generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(12):2319–2327.

Yu Bai, Yang Gao, and Heyan Huang. 2021. Cross-lingual abstractive summarization with limited parallel resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6910–6924, Online. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. 2022. Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs.

Florian Boudin, Stéphane Huet, and Juan-Manuel Torres-Moreno. 2011. A graph-based approach to cross-language multi-document summarization. *Polibits*, 43:113–118.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Saksham Singhal, Xian-Ling Mao, Heyan Huang, Xia Song, and Furu Wei. 2021. mT6: Multilingual pretrained text-to-text transformer with translation pairs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1671–1683, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Inigo Jauregi Unanue, Gholamreza Haffari, and Massimo Piccardi. 2023. T3l: Translate-and-test transfer learning for cross-lingual text classification.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Evaluating the efficacy of summarization evaluation across languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 801–812, Online. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Anton Leuski, Chin-Yew Lin, Liang Zhou, Ulrich Germann, Franz Josef Och, and Eduard H. Hovy. 2003. Cross-lingual c*st*rd: English access to hindi information. *ACM Trans. Asian Lang. Inf. Process.*, 2:245–269.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6038–6044, Hong Kong, China. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Chulun Zhou, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2022. A variational hierarchical model for neural cross-lingual summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2088–2099, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Constantin Orăsan and Oana Andreea Chiorean. 2008. Evaluation of a cross-lingual Romanian-English multi-document summariser. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Jessica Ouyang, Boya Song, and Kathy McKeown. 2019. A robust abstractive system for cross-lingual summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2025–2031, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Parnell, Inigo Jauregi Unanue, and Massimo Piccardi. 2022. A multi-document coverage reward for RELAXed multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5112–5128, Dublin, Ireland. Association for Computational Linguistics.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Models and datasets for cross-lingual summarisation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9408–9423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Gido M. van de Ven and Andreas S. Tolias. 2019. Three scenarios for continual learning.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9300, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaojun Wan. 2011. Using bilingual information for cross-language document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1546–1555, Portland, Oregon, USA. Association for Computational Linguistics.

Xiaojun Wan, Huiying Li, and Jianguo Xiao. 2010. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022. A Survey on Cross-Lingual Summarization. *Transactions of the Association for Computational Linguistics*, 10:1304–1323.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3054–3064, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Experimental Setup

For the evaluation of our approach, we have adopted ROUGE and BERTScore to assess both the surface and semantic matching between the predictions and the reference summaries. Given the number of ROUGE variants, we have chosen to report the average of ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. In specific, mROUGE[6] has been used for those languages where the underlying package (NLTK) had support for the language via special stemmers and/or language-specific tokenizers. We note that the adoption of mROUGE in the XLS literature is not widespread, probably because its reliance on dedicated stemmers and tokenizers is somehow limiting. Given this, and a recent advocacy for BERTScore in XLS (Koto et al., 2021), we have chosen to report BERTScore extensively. To ensure that we could compute it consistently for all the languages in our evaluation, we have populated it with the weights of the encoder of the pretrained multilingual LM used for the TRA module of SUMTRA (`mBART-large-50-one-to-many-mmt`).

In both the training of the monolingual summarizer and the few-shot fine-tuning of SUMTRA, we have selected the best checkpoints based on a) either meeting a validation criterion b) or reaching the maximum set number of training iterations.

A.2 Model Hyperparameters

Our baseline model is the pretrained `mBART-large-50` (Tang et al., 2021) in its various variants (one-to-many[7], many-to-many[8], and many-to-one[9]). All the models have been fine-tuned and run using PyTorch Lightning on a single NVIDIA A40 GPU with 48 GB of memory. Fine-tuning the entire SUMTRA with the chosen hyperparameters uses up approximately 70% of the total memory. Increasing the batch size and/or the input/output sequence length correspondingly increases the memory footprint. Table 6 reports the full list of the hyperparameters used for training, fine-tuning and inference.

[6] https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring

[7] https://huggingface.co/facebook/mbart-large-50-one-to-many-mmt

[8] https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

[9] https://huggingface.co/facebook/mbart-large-50-many-to-one-mmt

Hyperparameter	Value
Training SUM	
Warmup	500 steps
Input Length	512 tokens
Output Length	128 tokens
Fine-Tuning SUMTRA	
Warmup	0 steps
Input Length	512 tokens
Output Length	$84^{\dagger}/64^{\ddagger}$ tokens
Freeze Strategy	Train All
$\alpha/1-\alpha$ (Eq. 6)	0.99 / 0.01
Shared Hyperparameters	
Training LR	3×10^{-5}
Training Epochs	10
Early Stopping Criterion	2 epochs
Training Batch Size	1
Inference Batch Size	8
Gradient Accumulation	8
Optimizer	AdamW

Table 6: Hyperparameters used for training and evaluation of each module. (†) and (‡) superscripts correspond to the CrossSum and WikiLingua datasets respectively.

A.3 Dataset Links and Statistics

We refer the reader to the original papers (Ladhak et al., 2020; Bhattacharjee et al., 2022) for detailed statistics of the CrossSum[10] and WikiLingua[11] datasets, as well as access to the original data we have made use of in this work. For quick reference, Table 7 provides the total size of the training, validation, and test splits of the English-to-many versions of both datasets for the languages covered in our experiments. For the XSum dataset, we have downloaded the En-En data from Hugging Face[12].

Dataset	Train	Val	Test
CrossSum	22.3K	2.8K	2.8K
WikiLingua	117.4K	16.8K	33.5K
XSum	204K	11.3K	11.3K

Table 7: Total size of the training, validation and test splits for the languages covered in our experiments. For XSum, we have only used the En-En data.

A.4 Impact of Additional Monolingual Training

A key advantage of the proposed SUMTRA model is its ability to leverage the existing wealth of publicly-available monolingual summarization resources, allowing it to obtain a remarkable zero-shot performance. To probe the impact of additional monolingual summarization training, we post-train our CrossSum En-En summarizer with the XSum En-En summarization training set (Narayan et al., 2018), and repeat our zero- and few-shot tests over a high-resource language (Spanish) and a low-resource one (Bengali). The results plotted in Figure 4 show that the zero-shot performance in both languages has improved with the additional training. The improvement is more evident in the case of Spanish, possibly because of its greater linguistic similarity with English compared to Bengali. With an increasing number of fine-tuning samples, the performance of the two models (with and without additional training) over Spanish seems to have approximately converged. In the case of Bengali, the performance trend has been similar, but the impact of fine-tuning has been proportionally greater. This may be because Bengali is a very distant language from English, and fine-tuning in the target language can prove even more beneficial. We conclude that additional monolingual summarization training can be useful to boost zero-shot performance, but its impact dilutes as fine-tuning takes on.

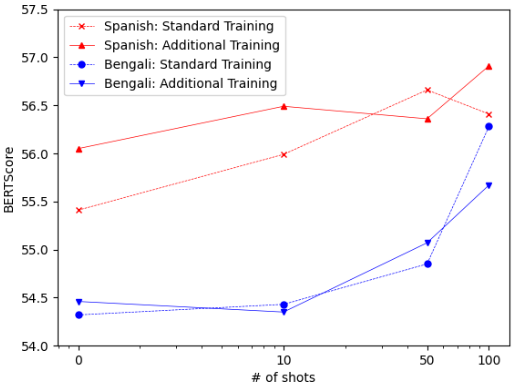

Figure 4: Impact of additional monolingual summarization training on the zero- and few-shot BERTScore performance over the CrossSum Spanish and Bengali test sets.

A.5 Cross-Domain Analysis

In this section, we explore the cross-domain robustness of SUMTRA by training the summarizer on the En-En data of one dataset and testing the model on a different dataset (i.e., En-En CrossSum-trained summarizer tested on WikiLingua, and vice versa).

[10] https://github.com/csebuetnlp/CrossSum (License: CC BY-NC-SA 4.0)

[11] https://github.com/esdurmus/Wikilingua (License: CC BY-NC-SA 3.0)

[12] https://huggingface.co/datasets/xsum (License: Unknown)

Table 8 presents the results for SumTra and an equivalent mBART-50 model, both fine-tuned with 100-shots in Spanish and Arabic from one dataset, and tested in the same language from the other dataset. We also report the results for mBART-50 fine-tuned with 1000 shots to show the competitiveness of our approach with 1/10 of the fine-tuning samples.

Model	**Spanish**	**Arabic**
	CrossSum-tuned + WikiLingua-tested	
mBART-50 (100-shot)	1.04 / 25.83	0.66 / 34.47
mBART-50 (1000-shot)	*10.72 / 47.24*	0.92 / 47.83
SumTra (100-shot)	**9.89 / 46.87**	**5.44 / 54.07**
	WikiLingua-tuned + CrossSum-tested	
mBART-50 (100-shot)	1.41 / 26.28	1.30 / 35.41
mBART-50 (1000-shot)	*12.82 / 47.74*	5.37 / *53.42*
SumTra (100-shot)	**10.82 / 44.84**	**5.65 / 53.00**

Table 8: Cross-domain ROUGE/BERTScore scores for Spanish and Arabic. The top rows are for CrossSum-tuning and WikiLingua-testing, and the bottom rows for the vice versa. For mBART-50 (1000-shot), we have italicized the results that have surpassed SumTra (100-shot).

The result trends in Table 8 are significantly lower than those in Tables 1 and 2; however, the performance gap between SumTra (100-shot) and mBART-50 (100-shot) has remained wide. These results further highlight the benefits of the proposed pipeline-based approach, as they show that it generalizes quite well across domains (news in the case of CrossSum, and how-to articles for WikiLingua). In turn, mBART-50 (1000-shot) has outperformed SumTra in some cases, but only marginally, despite requiring 10X the number of fine-tuning samples.

A.6 Soft vs. Hard Predictions at Inference Time

In the proposed model, the use of soft predictions is strictly required during fine-tuning, but becomes an option at inference time. For this reason, in this section, we examine the impact of using either soft or hard predictions for inference. As hard predictions, we simply extract the argmaxed predictions from the summarizer and pass them to the translator, without converting them to embedding space and bypassing the embedding layer of the translator.

Table 9 shows the results over the CrossSum Spanish test set for a 100-shot configuration for both cases. It is clear that the hard predictions have led to noticeably better scores. While this is only for a single language, it is reasonable to assume that these results may generalize to other languages, given that using the argmax provides a more confident and tighter input to the translation module.

Type	mROUGE	BERTScore
Hard	**21.37**	**56.41**
Soft	20.47	55.77

Table 9: Soft vs. hard predictions at inference time over the CrossSum Spanish test set.

To complement these results, we present a short qualitative example in Table 10. For both types of predictions, we have fine-tuned the model using the soft predictions, but passed either hard or soft predictions to the translator module for inference. For clarity, the summarizer generates the same intermediate summary in both cases. As the BERTScore values show, there is little semantical difference between the two types of prediction. However, given that the argmax has obtained a mildly higher score (alongside a minor inference speedup), we have chosen to use the hard predictions throughout our experiments.

A.7 Additional Qualitative Analysis

To supplement Table 3, in Table 11 we show another qualitative example for Indonesian from WikiLingua. For this example, we have only compared SumTra with and without the use of the back-translation loss. Without the back-translation loss, the summary predicted by SumTra has made reference to angel birds (*burung-burung malaikat*) and painting (*cara untuk mengecatkan*) as a means of decorating a costume. The prediction has also included an incorrect capitalization of "you" (*Anda*). While we can roughly infer what the predicted summary means, the summary predicted by SumTra with the back-translation loss has made the conveyed meaning much clearer. Specifically, SumTra with the back-translation loss has referred to making wings (*buat sayap*) and a halo (*halo*), aligning more closely with the meaning of the reference summary (e.g., *buatlah sayap*). Like in the qualitative example in Table 3, even this summary is still imperfect, as we note a false generation of the phrase "*kain jambu*". However, as mentioned in the main paper, we expect that for low-resource languages such as Indonesian, dedicated training of the translator would be able to improve the translation quality and further boost BERTScores.

Model	**Summary**	**BERTScore**
Reference	Un hombre demasiado asustado para volar debido a la pandemia vivió sin ser detectado en un área segura del aeropuerto internacional de Chicago durante tres meses, según los fiscales de EE.UU.	
Intermediate Summary	A man arrested after allegedly stealing a badge from an airport in Chicago was "unauthorised, non-employee" according to the official prosecutor.	
Argmax	**Prediction:** Un hombre detenido después de haber supuesto **robo de un badge** en un **aeropuerto de Chicago** fue "**no autorizado**, **no asalariado**" según el fiscal oficial.	56.03
Soft	**Prediction:** Un hombre detenido por supuesto **robo de un cohete** de un **aeropuerto de Chicago** fue "**no autorizado**", **no trabajador**", según el fiscal oficial.	55.43

Table 10: Qualitative example to support the use of the hard vs. soft predictions at inference time (CrossSum Spanish). **(Red)** denotes incorrect translations or factual inconsistencies, **(Blue)** denotes information from the source document, and **(Green)** refers to matching information in the reference summary.

Model	**Summary**	**BERTScore**
Reference	Buatlah sayap. Buatlah lingkaran cahaya. Kombinasikan sayap dan lingkaran cahaya dengan kostum. **Back-Translation:** Make wings. Make circles of light. Combine wings and circles of light with costumes.	
SumTra (100-shot)	**Intermediate Summary:** Make or buy wings. Make or buy a halo. Make or buy a scarf. **Prediction:** **Buat** atau beli **sayap**. **Buat** atau beli **halo**. **Buat** atau beli **kain jambu.**	57.54
SumTra (100-shot) (no BT loss)	**Intermediate Summary:** Angel wings are a way of decorating your Halloween costume. **Prediction:** **Burung-burung** **malaikat** adalah **cara untuk mengecatkan** kostum **Halloween Anda**.	45.63

Table 11: Qualitative example for Indonesian (WikiLingua) for SumTra (100-shot) with and without the use of the back-translation (BT) loss. **(Red)** denotes incorrect translations or factual inconsistencies, **(Blue)** denotes information from the source document, and **(Green)** refers to matching information in the reference summary.