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

Anonymous ACL submission

Abstract

Cross-lingual summarization (XLS) generates summaries in a language different from that of the input documents (e.g., English to Spanish), allowing speakers of the target language to gain a concise view of their content. 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 samples makes this approach challenging 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, that the proposed approach has been able to outperform in many languages with only 10% of the fine-tuning samples.

1 Introduction

Cross-lingual summarization (XLS) aims to take a document written in a given source language and generate a summary in a chosen target language, providing the speakers of the latter with the ability to concisely understand the content of documents written in foreign languages. However, XLS is a challenging task due to the limited training data which are typically available. Unlike in 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., 2022b). This has often led to the reuse of existing multilingual data with post-hoc alignments 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 employing existing multilingual LMs (Liu et al., 2020; Tang et al., 2021; Xue et al., 2021), pretrained in the typical unsupervised manner over large corpora, and fine-tuning them with the limited XLS resources available for the chosen 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 across languages 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 (i.e., 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 and few-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 comparable multilingual language model baselines with up to 1000-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.

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

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 justifiable, these approaches inevitably suffered 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 pretrained 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; Dou 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

[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 for fine-tuning.

languages. More recently, Wang et al. (2023b) has proposed leveraging various large (>100B parameters) language models for zero-shot cross-lingual summarization. By contrast, in this paper we intentionally focus on the utilization of much smaller, modular, and trainable models in the zero- and few-shot scenario.

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:

$$\mathbf{p}_j = \text{SUM}(s_{j-1}, x, \theta) \quad (1)$$

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:

$$\mathbf{e}_j = \mathbb{E}[\mathbf{E}]_{\mathbf{p}_j} = \mathbf{E}\,\mathbf{p}_j \quad (2)$$

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:

$$\bar{y} = \text{TRA}(\mathbf{e}, \sigma) \quad (3)$$

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:

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

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 the standard negative log-likelihood of the ground-truth summary in the target language allows for too many degrees of freedom in the generation of the intermediate English summary, and can lead to inaccurate summaries with respect to the source document. 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:

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

where $\hat{y}$ denotes the back-translated sequence, and $p(\hat{y})$ the probabilities output by the summarizer. We note that our use of a separate summarization module would also allow using 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 have decided to leave this exploration to future work.

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

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

using a scaling coefficient, α, that acts as a hyper-parameter in the loss. We have set α to 0.99 for all experiments, and report a sensitivity analysis in Appendix A.5.

4 Experimental Setup

4.1 Datasets, Baselines, 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 categorized them as high-, medium- and low-resource based on the number of sentences used for the pretraining of the respective language in our main baseline, mBART-50 (Tang et al., 2021).

To implement the proposed approach, we have used the mBART-50 one-to-many[3] variant for the TRA module, and the many-to-one[4] variant for both the SUM module and the generation of the back-translations used for fine-tuning (Equation 5). The back-translations have been generated once and for all offline, and added to the dataset.

As baselines, we have employed various, strong multilingual models that include: 1) the mT5-m2m model of Bhattacharjee et al. (2022), fine-tuned on all languages and full training splits of the CrossSum dataset; 2) a pretrained mBART-50 (Tang et al., 2021), both with and without an initial training with a monolingual English dataset (respectively, mBART-50-mono and mBART-50 in the following); 3) two 175B-parameter language models (ChatGPT and davinci-003), leveraging a "direct" and "summarize-then-translate" prompt, respectively, as defined in Wang et al. (2023a), and 4) the PISCES model of Wang et al. (2023b) – a modified mBART-50 model that leverages extra cross-lingual and task-specific pretraining over huge resources (20.6M samples from the OPUS parallel corpora and 3.1 from mC4, respectively).

To evaluate the predictions, we have used ROUGE (Lin, 2004) and its multilingual adaptation[5], mROUGE (Conneau and Lample, 2019), which leverages language-specific tokenizers and stemmers to pre-process non-English text prior to a standard ROUGE calculation. We have computed the ROUGE scores as an 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 ability to better 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 respective English-English training split of CrossSum or WikiLingua[6], and chosen the best performing checkpoint based on a validation criterion. For the experiments in the few-shot fine-tuning configuration, 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 A.4. Further details of the experimental setup are provided in Appendixes A.1 and A.2.

5 Results and Analysis

Tables 1 and 2 present the results of the proposed approach and comparative baselines over the chosen language pairs, grouped into high-, medium-, and low-resource languages, for the CrossSum and WikiLingua datasets, respectively.

SUMTRA vs. mBART-50. In both tables, we compare the proposed SUMTRA model with both mBART-50 and mBART-50-mono (the version with an initial English summarization training), and in both zero- and few-shot configurations (50-1000 examples). The results show that the English training can be beneficial for improving the average zero- and few-shot performance of mBART-50 (Wang et al., 2022a); however, the results are not consistent across languages, even for those that are linguistically similar (e.g., Spanish, French). SUMTRA comparatively displays much stronger average zero- and few-shot performance up to and including 1000 shots, showing the usefulness of the proposed approach. For instance, SUMTRA (0-shot) outperforms both mBART-50 variants with 1000 shots on average over the CrossSum languages. In a similar fashion, at a parity of fine-tuning samples (1000-shots), the most performant SUMTRA model outperforms mBART-50 by +1.28 BERTScore pp on average over the WikiLingua languages.

SUMTRA vs. PISCES. We also compare SUMTRA against PISCES, but for brevity, limit the experiments to the zero-shot configuration downloaded from https://huggingface.co/Krystalan/PISCES. The results show a comparatively rather modest performance from PISCES, with the exception of two staggering results for the Chinese and Thai languages of WikiLingua. Since these scores are much higher than those reported in Wang et al. (2023b) for a fully fine-tuned PISCES

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

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

[5] For brevity, we will refer to "ROUGE" as "mROUGE" throughout, to accommodate all languages. Details on mROUGE are provided in Appendix A.1.

[6] Appendix 5.1 explores other options for the monolingual summarization training.

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
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
mBART-50-mono (0-shot)	5.39 / 29.98	4.97 / 31.58	0.20 / 21.74	1.75 / 23.47	2.00 / 21.84	0.00 / 16.31	2.39 / 24.15
mBART-50-mono (50-shot)	5.42 / 30.11	4.98 / 31.60	0.20 / 21.74	1.78 / 23.48	1.99 / 22.01	0.00 / 16.33	2.40 / 24.21
mBART-50-mono (100-shot)	5.66 / 30.76	4.88 / 31.64	0.20 / 21.73	1.73 / 23.56	2.18 / 21.59	0.00 / 16.42	2.44 / 24.28
mBART-50-mono (1000-shot)	18.65 / 54.06	16.69 / 50.91	12.52 / 58.38	7.52 / 56.67	13.56 / 51.68	7.78 / 62.62	12.79 / 55.72
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
SUMTRA (1000-shot)	21.29 / 56.41	20.30 / 53.94	**17.57 / 61.73**	**10.17 / 60.48**	15.74 / 55.94	6.11 / 58.58	15.20 / 57.85
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**
davinci-003 (ST) (Wang et al., 2023a)	13.71 / 50.74	6.58 / 24.46	8.74 / 55.60	5.52 / 54.96	9.17 / 49.27	4.82 / 61.66	8.09 / 49.45
ChatGPT (Direct) (Wang et al., 2023a)	16.20 / 52.02	13.75 / 47.41	10.24 / 56.36	4.03 / 54.78	11.14 / 47.85	3.99 / 60.69	9.89 / 53.19
PISCES (Wang et al., 2023b)	3.02 / 31.92	9.93 / 42.73	0.08 / 44.65	0.73 / 39.56	3.04 / 35.77	0.00 / 53.63	2.80 / 41.38

Table 1: Results for the CrossSum dataset, grouped into high, medium, and low-resource languages. We report the average of ROUGE-1, ROUGE-2, and ROUGE-L F1 (or the mROUGE equivalent where applicable as denoted with †) and BERTScore. The best scores are boldfaced.

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
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
mBART-50-mono (0-shot)	0.58 / 31.85	9.01 / 36.00	0.28 / 26.03	2.24 / 28.79	12.79 / 29.02	2.06 / 32.35	4.49 / 30.67
mBART-50-mono (50-shot)	0.57 / 31.86	8.98 / 36.00	0.28 / 26.03	2.24 / 28.78	12.79 / 29.02	2.05 / 32.34	4.48 / 30.68
mBART-50-mono (100-shot)	0.58 / 31.85	8.98 / 36.02	0.28 / 26.03	2.24 / 28.78	12.79 / 29.03	2.05 / 32.34	4.49 / 30.68
mBART-50-mono (1000-shot)	11.16 / 58.41	20.36 / 62.37	10.09 / 60.41	13.69 / 54.74	22.25 / 67.32	11.57 / 53.36	14.85 / 59.44
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
SUMTRA (1000-shot)	**13.38 / 59.85**	21.13 / 63.12	**13.04 / 62.61**	**16.23 / 57.94**	18.93 / 58.87	**14.67 / 57.09**	**16.23 / 59.91**
davinci-003 (ST) (Wang et al., 2023a)	10.37 / 53.19	10.80 / 38.48	8.78 / 56.23	9.55 / 52.25	12.84 / 58.84	10.37 / 50.45	10.45 / 51.57
ChatGPT (Direct) (Wang et al., 2023a)	8.52 / 52.55	15.33 / 53.19	7.34 / 55.18	9.24 / 53.17	10.45 / 58.07	10.75 / 51.30	10.27 / 53.91
PISCES (Wang et al., 2023b)	0.59 / 34.25	***42.65 / 73.66***	0.34 / 41.99	4.32 / 38.73	***47.13 / 78.60***	1.83 / 43.21	16.14 / 51.74

Table 2: Results for the WikiLingua dataset, grouped into high, medium, and low-resource languages. We report the average of ROUGE-1, ROUGE-2, and ROUGE-L F1 (or the mROUGE equivalent where applicable as denoted with †) and BERTScore. The best scores are boldfaced. The italicized results are commented upon in Section 5.

model, we speculate that there may exist some overlap between some of their training data and our test sets. An alternative explanation is that Chinese and Thai were part of PISCES' pre-training languages, and the alignment with their WikiLingua's test sets may have proved extraordinarily effective. For all other languages, SUMTRA has displayed a much stronger zero-shot performance compared to PISCES, confirming the validity of our pipeline approach.

SUMTRA vs. mT5/ChatGPT/davinci-003. Lastly, we compare SUMTRA to the remaining baselines: the mT5 many-to-many model, ChatGPT, and davinci-003. We note that the mT5 model 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 language pair) (Bhattacharjee et al., 2022), and should therefore be regarded in Table 1 as a hard-to-near upper bound. With that said, SUMTRA has obtained higher scores for 3 of the 6 languages, and competitive scores for the other three. Lastly, ChatGPT and davinci-003 have obtained some of the lowest average mROUGE and BERTScore scores compared to the other models, showing that they lack the task-specific capability that even a few-shot mBART-50 or SUMTRA model displays.

Overall, these results show that the proposed SUMTRA model is capable of a 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.

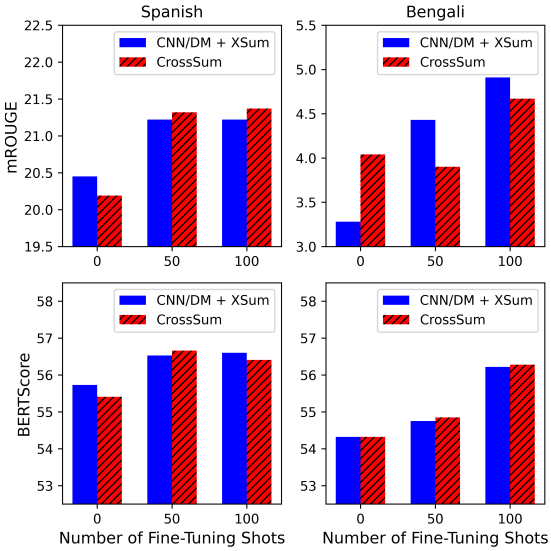

Figure 1: Performance comparison between SUMTRA models trained with CNN/DM and XSum, and with the CrossSum English training split.

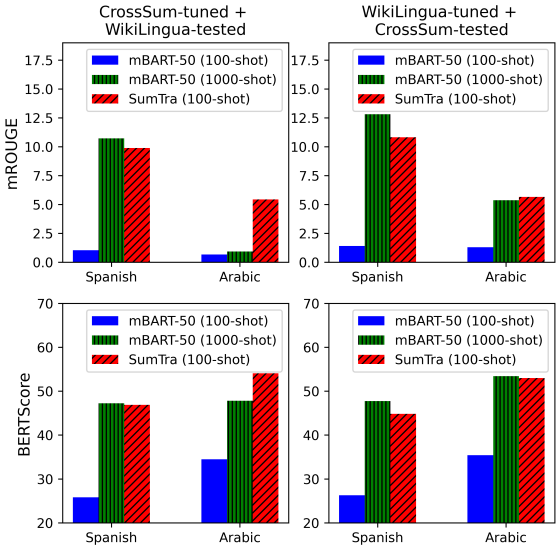

Figure 2: Cross-domain mROUGE/BERTScore scores for Spanish and Arabic. Left: CrossSum-tuned and WikiLingua-tested; Right: vice versa. We have also included mBART-50 (1000-shot) to highlight SUMTRA's few-shot capability.

5.1 Alternative Monolingual Training

Given the vast amounts of available English summarization datasets, we have also explored training the SUM module with two widespread datasets, CNN/DailyMail (See et al., 2017) and XSum (Narayan et al., 2018) in alternative to the English training splits of the XLS datasets. For simplicity, we have first trained the summarizer on CNN/DM, and then continued training on XSum. We have then performed zero-, 50-, and 100-shot fine-tuning of SUMTRA, and compared the performance with the model trained on the CrossSum English split. The results over the Spanish and Bengali test sets are displayed in Figure 1, showing that the performance has been approximately on par and always close. We can then argue that re-training the summarizer for every specific XLS dataset may be unnecessary, and that the zero-shot performance of the proposed approach trained with generic English summarization resources is likely to remain competitive over a variety of domains.

5.2 Cross-Domain Analysis

In addition, we have explored the cross-domain robustness of SUMTRA by training and fine-tuning the model on one dataset and testing it on the other (i.e., training with CrossSum and testing on WikiLingua, and vice versa). Figure 2 shows 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 on the other. We also report the results for mBART-50 fine-tuned with 1000 shots to show the competitiveness of our approach with just 10% of the fine-tuning samples.

Overall, the result trends shown in Figure 2 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 reasonably well across domains (news for CrossSum and how-to articles for WikiLingua), particularly in a few-shot setting. mBART-50 (1000-shot) has been able to outperform SUMTRA (100-shot) in some cases, but only marginally.

5.3 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 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 full fine-tuning scenarios.

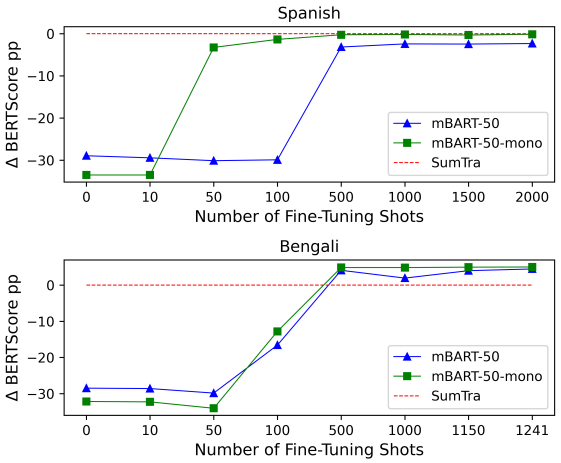

Figure 3: Exploring the catastrophic forgetting problem with mBART-50, mBART-50-mono and SUMTRA on the CrossSum Spanish and Bengali test sets.

To this aim, Figure 3 plots the relative changes in BERTScore for mBART and mBART-mono over Spanish and Bengali at an increasing number of fine-tuning samples. For this experiment we have used all the 1241 available fine-tuning samples for Bengali, and 2000 fine-tuning samples for Spanish.

For both languages, it is manifest that SUMTRA is the only model 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. At the same time, 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.

5.4 Qualitative Analysis

To qualitatively show that SUMTRA achieves better performance than mBART with fewer shots, in Table 3 we report an example for Spanish, comparing an mBART-50-mono model fine-tuned with 1000 shots with a SUMTRA model fine-tuned with 1/10 of the shots (100). For further comparison, we also show the summary generated by SUMTRA fine-tuned without the back-translation (BT) loss of Equation 5. The summary generated by the mBART-50-mono model undoubtedly contains 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, both predictions from the SUMTRA models have 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: Table 11 in Appendix A.8), 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. Additional, commented examples are provided in Appendix A.8.

5.5 Inference Time

Given that the proposed model uses two language models in pipeline, it is important to compare its in-

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.

ference times to those of the baseline. To this aim, Table 4 reports the inference times per sample[7] 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. For Bengali, the larger overhead has mainly been due to an average lengthening of the predicted intermediate summaries, which has increased both the summarization and the translation times. In turn, the lengthening of the intermediate summaries has likely been induced by the back-translated summaries, which have been on average slightly longer than the references. 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 4: Average inference times per sample for mBART-50 and SUMTRA over the CrossSum Spanish and Bengali test sets.

[7]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.

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 evaluated the proposed approach over two mainstream XLS datasets and against a set of performing baselines, giving evidence to the competitive performance of the proposed approach. In particular, SUMTRA's zero-shot performance has proved very strong, and its few-shot performance has been remarkable for a 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 base model are both manageable. In the future, we aim to test model configurations with different base language models (e.g., PISCES) for the summarization and translation modules, and explore alternative fine-tuning strategies such as adversarial training and reinforcement learning.

Limitations

The proposed approach has various limitations. The most immediate is that we have limited our experimental validation to the English-to-many case. 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, an actual, intrinsic limitation of the proposed approach is that it relies on a 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 single, equivalent multilingual model. In particular, the memory required during fine-tuning (with the selected hyperparameters) has been approximately 34 GB. However, in Appendix A.4 we show 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 model. At its turn, the training time of the proposed model has only been approximately 1.6x times that of a single model, and should not hinder its use.

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. As a final clarification, the generation of the back-translations used for fine-tuning is conducted offline and one-off, and their auxiliary fine-tuning objective carries no measurable computational overhead.

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.

Zi-Yi Dou, Sachin Kumar, and Yulia Tsvetkov. 2020. A deep reinforced model for zero-shot cross-lingual summarization with bilingual semantic similarity rewards. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 60–68, Online. Association for Computational Linguistics.

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. *Transactions of the Association for Computational Linguistics*, 11:1147–1161.

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 pre-training 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.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023a. Zero-shot cross-lingual summarization via large language models.

Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022a. ClidSum: A benchmark dataset for cross-lingual dialogue summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7716–7729, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022b. A survey on cross-lingual summarization. *Transactions of the Association for Computational Linguistics*, 10:1304–1323.

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023b. Towards unifying multi-lingual and cross-lingual summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15127–15143, Toronto, Canada. Association for Computational Linguistics.

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*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

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

We have selected six languages from both CrossSum and WikiLingua, and self-categorized them into high, medium, and low-resource based on the number of pretraining sentences used in Tang et al. (2021). The groupings are selected as follows: languages with >1M pretraining sentences have been labelled as high-resource, between 100k and 1M as medium-resource, and <100K as low-resource. We refer the reader to Table 6 of Tang et al. (2021) for language-specific breakdowns.

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. As mentioned in the main body, we have chosen to report the average of ROUGE-1, ROUGE-2, and ROUGE-L F1 scores, in line with previous summarisation literature. More specifically, mROUGE[8] has been used in our experiments for languages where existing language-specific stemmers and/or tokenizers are made available by the underlying package (NLTK). 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`).

A.2 Model Hyperparameters

Our baseline model is the pretrained `mBART-large-50` (Tang et al., 2021), with its variants (one-to-many[9], many-to-many[10], and many-to-one[11]) utilized throughout the paper. 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, as expected. Table 5 reports the full list of the hyperparameters used for training, fine-tuning and inference.

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
α (Eq. 6)	0.99
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 5: Hyperparameters used for training and evaluation of each module. The (†) and (‡) superscripts are for the CrossSum and WikiLingua datasets, respectively.

For model training, when training the monolingual summarizer, or conducting few-shot fine-tuning of SUMTRA and the mBART-50 variants, we have selected the best checkpoints based on either a) meeting a criterion based on validation performance, or b) reaching the maximum set number of training iterations/epochs. For mBART-50-mono, we have used the same hyperparameters as for our mBART-50 baseline model, with the exception that the former has first been trained on an English-English split of either CrossSum or WikiLingua, depending on the downstream fine-tuning dataset. This is the equivalent of the training of the SUM module used in SUMTRA. Lastly, for ChatGPT and davinci-003 we have used the OpenAI platform between the 18th and 28th of October 2023 (ChatGPT), and between the 12th and 28th of November 2023 (davinci-003).

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 and WikiLingua datasets, as well as access to the original data we have made use of in this work.

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

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

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

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

For quick reference, Table 6 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. Table 7 provides the actual links and license types.

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 6: 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.

GitHub	License
https://github.com/csebuetnlp/CrossSum	CC BY-NC-SA 4.0
https://github.com/esdurmus/Wikilingua	CC BY-NC-SA 3.0
https://huggingface.co/datasets/xsum	Unknown

Table 7: GitHub repositories and license details for the CrossSum, WikiLingua, and XSum datasets.

A.4 Fine-Tuning Ablation

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 SUMTRA's performance by fine-tuning only either the summarizer or the translator, and comparing it to fine-tuning both jointly. This is to show that 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 4 plots the BERTScore of the three configurations for Spanish and Bengali, with an increasing amount of fine-tuning samples. 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, in the case of Bengali in particular, fine-tuning only the translator with 50 shots has achieved performance that has surpassed the tuning of both the

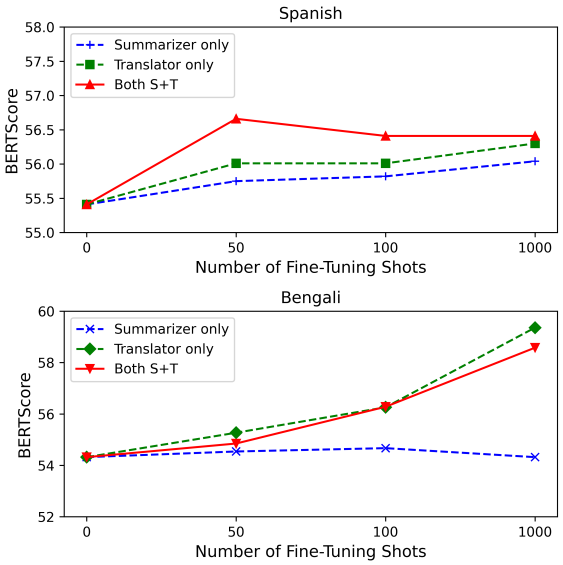

Figure 4: BERTScore scores for the CrossSum Spanish and Bengali test sets with different fine-tuning configurations (summarizer only, translator only, and both).

summarizer and translator together. 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 behavior of the translation component can be very language-dependent.

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.

A.5 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 8 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 could have grid-searched an optimal 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 convenient 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.

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

A.6 Sensitivity to Different Embedding-based Metrics

As a further sensitivity analysis, we explore the sensitivity of the results to the BERTScore evaluation metric by comparing it with MoverScore (Zhao et al., 2019). These two metrics are rather similar, as they are both variants of optimal transport. However, their main difference is that BERTScore performs a one-to-one alignment between the tokens of the prediction and the reference, while MoverScore performs a one-to-many, allowing a token to receive a good matching score from the accumulation of multiple, partial matches[12].

Figure 5 shows the BERTScore and MoverScore values for mBART-50 and SumTra for Spanish and Bengali in zero- and few-shot configurations. In addition, the values for the fully-trained mT5(m2m) are displayed for reference as an informal upper-bound. For Spanish, the qualitative trends for BERTScore and MoverScore are similar, with the only notable difference that the MoverScore values are more compressed in range. For Bengali, the

[12]For computing MoverScore, we have used BERT-base-multilingual-uncased (https://huggingface.co/bert-base-multilingual-uncased).

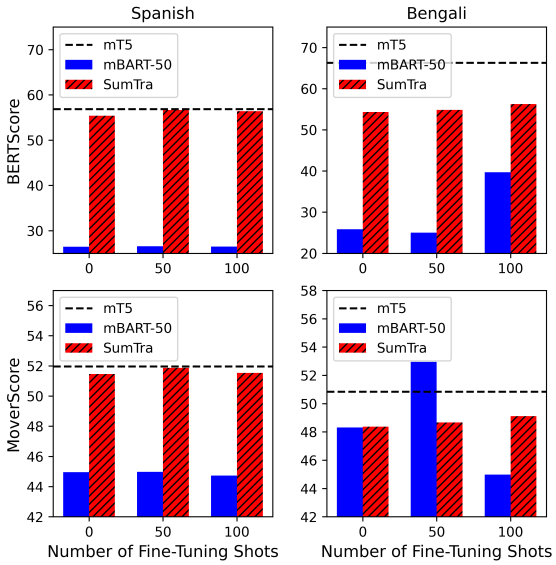

Figure 5: BERTScore and MoverScore comparison over the Spanish and Bengali test sets (CrossSum)).

trends have instead differed significantly, with the MoverScore values for mBART-50 and SumTra being roughly on par on average. However, the MoverScore results for mBART-50 show a very marked drop for 100-shot fine-tuning, which seems to contradict the qualitative evaluation and the expected impact from fine-tuning. For this reason, we have chosen to report BERTScore in the main paper.

A.7 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.

To showcase the differences, Table 9 presents a short qualitative example. 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. While these results are only for a single language, it is reasonable to assume that they may generalize to other languages, given that using the argmax provides a more confident and tighter input to the translation module.

A.8 Additional Qualitative Analysis

To supplement Table 3, in Table 10 we show another qualitative example from WikiLingua for Indonesian. 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, a dedicated training of the translator should be able to improve the translation quality and further boost BERTScores.

Additionally, to qualitative assess the performance of ChatGPT and davinci-003, Table 11 shows their predictions for the same example displayed in Table 3 in the main paper. In the case of davinci-003, the summarize-then-translate prompt has not worked very well in terms of length reduction, since the generated output has still come out relatively long. However, details of the input document have been relayed well in the generated summary. In contrast, the direct prompt used with ChatGPT has been effective at generating a shorter summary. However, the summary is truncated and has modest semantic correlation with the reference, as reflected by its low BERTScore. In contrast, the 100-shot SUMTRA model has retained a higher alignment with the reference summary (+10 pp BERTScore). As stated in Section 5, these two LLMs have not been able to match the task-specific capability of the dedicated, smaller models (mBART-50, SUMTRA, PISCES).

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 9: 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 10: 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.

Model	Summary	BERTScore
Input Document	According to court documents, the National Security Agency (NSA) had demanded that Yahoo comply with new surveillance rules, something the company said was unconstitutional. Yahoo failed in a court challenge on the constitutionality of the order. But the details emerged on Thursday when a federal judge ordered the unsealing of some material about the case. Yahoo's general counsel Ron Bell said publication of the material was "an important win for transparency". Yahoo said that the government amended a law to demand user information from online services, prompting a court challenge. Former NSA contractor Edward Snowden disclosed the programme last year. But the court documents reveal that the battle over surveillance between technology firms and the US government stretched back years before the Snowden revelations. The new material about the case, first reported by the Washington Post, underscores "how we had to fight every step of the way to challenge the US government's surveillance efforts',' Mr Bell added. "At one point, the US government threatened the imposition of $250,000 in fines per day if we refused to comply," he said. About 1,500 pages of previously classified documents were unsealed by a federal court.	
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.	
SUMTRA (100-shot)	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
davinci-003 (ST)	Según los documentos judiciales, la Agencia de Seguridad Nacional (NSA) exigió que Yahoo cumpliera con nuevas reglas de vigilancia, algo que la empresa dijo que era inconstitucional. Yahoo fracasó en un desafío judicial sobre la constitucionalidad de la orden. Pero los detalles surgieron el jueves cuando un juez federal ordenó desclasificar algunos materiales sobre el caso. El abogado general de Yahoo, Ron Bell, dijo que la publicación de los materiales fue "una importante victoria para la transparencia". Yahoo dijo que el gobierno modificó una ley para exigir información de los usuarios de los servicios en línea, lo que desencadenó un desafío judicial. El contratista de la NSA Edward Snowden dio a conocer el programa el año pasado. Pero los documentos judiciales revelan que la batalla entre las empresas tecnológicas y el gobierno de Estados Unidos se extendió durante años antes de las revelaciones de Snowden. Los nuevos materiales sobre el caso, informados por primera vez por el Washington Post, subrayan "cómo tuvimos que luchar en cada paso para desafiar los esfuerzos de vigilancia del gobierno de EE. UU.", Agregó el Sr. Bell. "En un momento, el gobierno de EE. amenazó con la imposición de multas de $250,000 por día si nos negábamos a cumplir", dijo. Unos 1.500 páginas de documentos previamente clasificados fueron desclasificados por un tribunal federal.	48.15
ChatGPT (Direct)	De acuerdo con documentos judiciales, la Agencia de Seguridad Nacional (NSA) exigió que Yahoo cumpliera con nuevas reglas de vigilancia, algo que la compañía consideró inconstitucional. Yahoo perdió en un desafío legal sobre la constitucionalidad de la orden. Sin embargo, los detalles salieron a la luz el jueves cuando un j	47.96

Table 11: Further qualitative example for Spanish (CrossSum) comparing the proposed approach with large language models, namely ChatGPT and davinci-003. We have included the input document here for reference.