LiSTra
Automatic Speech Translation: English to Lingala case study

Anonymous ACL submission

Abstract

In recent years there have been great interests in addressing the low resourcefulness of African languages and provide baseline models for different Natural Language Processing tasks (Orife et al., 2020). Several initiative (Nekoto et al., 2020) on the continent uses the Bible as a data source to provide proof of concept for some NLP tasks. In this work, we present the Lingala Speech Translation (LiSTra) dataset, release a full pipeline for construction of such dataset in other languages and report baselines using both the traditional cascade approach (Automatic Speech Recognition -> Machine Translation), and a revolutionary transformer based End-2-End architecture (Liu et al., 2020) with a custom interactive attention that allows information sharing between the recognition decoder and the translation decoder.

1 Introduction

Automatic Speech Translation (AST) is the task of converting an utterance from a source language to transcription in a target language, such a task has several applications in real life. Success in this task will revolutionize online education, the majority of educational content available on e-learning platforms like Udacity, Edx, and Coursera among others are English-centric and this is a bottleneck to people with limited or no knowledge of English to have access to those contents. As a starting point in this direction, inspired by (Orife et al., 2020) we performed a proof of concept for Automatic Speech Translation from a higher resources language (English) to a lower one, Lingala in this case.

Lingala (Ngala) (Lingala: lingála) is a Bantu language spoken throughout the northwestern part of the Democratic Republic of the Congo (Wikipedia contributors, 2020) and a large part of the Republic of the Congo. It is spoken to a lesser degree in Angola, the Central African Republic, and Southwest & Southcentral Republic of South Sudan. There are over 40 million lingalaphones [1].

Based on a study made in 2009 by youthpolicy[2] the population of the Democratic Republic of the Congo (DRC) is young and rejuvenating over 68% of people aged less than 25 years, a majority of whom live in rural areas (over 60 %), this situation has not much changed since. This young population is not always able to speak the official language (French) and this work is a start to making educational materials available to them.

One bottleneck in experimenting on ASR especially for low resources languages has been lack of aligned data, inspired by the Masakhane (Orife et al., 2020) initiative and (Agic and Vulic, 2020) we introduce in this paper **LiSTra**[3] which stand for **Ligala Speech Translation** a dataset of reading of the Bible, the corresponding transcription in English as well as the Lingala translation. The choice of the bible as a data source is motivated by missionary work on the African continent, which made indirectly available the transcription and the translation alignments. Despite the religious nature of the content in the Bible, some of its recent version provide a good starting point for experimentation in several NLP tasks.

The traditional approach in AST is what is known as a pipeline system where we first do Automatic Speech Recognition(ASR), then feed the output into a Machine Translation (MT) system, one pitfall in this approach is the error propagation (not back-propagation) that arise due to the fact that the 2 components are trained independently. In this work we will release a baseline for AST both in a pipeline (ASR -> MT) as well as in an end-to-end setting, in addition, we published what happens to be at the best of our knowledge the first dataset for

[1] https://en.wikipedia.org/wiki/Lingala

[2] https://www.youthpolicy.org/factsheets/country/congo-kinshasa.

[3] Anonymous

neural speech translation from English to Lingala.

Our main contributions are summarized are follows:

- Release a detailed methodology to create new datasets for Automatic Speech Translation (AST) for low resource languages which can be also useful both for Machine Translation (MT) and Automatic Speech Recognition tasks independently.
- Provide a baseline for AST for English-to-Lingala in both pipeline and end-2-end settings

2 Related work

The recent breakthroughs in end-to-end architectures in Machine Translation and Speech Recognition have lead to the investigation of having end-to-end architectures for Automatic Speech Translation (Bérard et al., 2016). Historically Automatic Speech Translation (ASR) was done in two steps: we first do automatic speech recognition on the source language and next feed the obtained transcription into a separate machine translation model, this is sometimes referred to in the literature as Cascade Speech Translation (Cascade-ST). One immediate issue with this approach is the error-propagation (not back-propagation) (Bahdanau et al., 2014).

Since the first AST proof of concept proposed by (Zong et al., 1999) there has been interesting works to improve on the state of the art, this is mostly because of it business side as well as community impact, for example, people with disability can use the outcome of this task to learn and get access to information. Due to the difficulty of the accessibility of aligned data, there has been some attempt to perform AST without source transcription (Bérard et al., 2016).

African languages have been for a long time left behind in the Major NLP conference. Recently, there have been initiatives like Deep Learning Indaba[4] and Data Science Africa[5] among others that aim to focus on solving and addressing African's problems using Machine Learning learning and AI. These movements have given birth to Masakhane which is an African initiative that focuses on Natural Language Processing related problem in the continent (Orife et al., 2020). The Masakhane initiative has been mostly at it current state making use of the JW300 dataset (Agic and Vulic, 2020) which is basically made of religious text that is inherently aligned on chapter and verse level and this has allowed the community to publish (Nekoto et al., 2020) baselines for several languages which were before untouched despite the number of people speaking and using them.

Our work in this paper aligned mostly with this work (Liu et al., 2020), that implemented a revolutionary architecture based on transformers that allow having 2 decoders that communicate among themselves in an intuitive way to perform Automatic Speech Translation but in our context, we will experiment with this same architecture in a low-resource setting to rapport its performance for English to Lingala translation.

3 Dataset

In the 20^{th} century, data is considered to be the new oil (Arthur), especially in supervise learning regimes where we can't talk of Machine learning without it. Africa currently has 2144 living languages (Eberhard et al., 2019). Despite this, African languages account for a small fraction of available language resources, and NLP research rarely considers African languages (Nekoto et al., 2020). Inspired by the work by (Orife et al., 2020) and (Agic and Vulic, 2020) we made use of the structural form of the bible, to create LiSTra.

Let $D = \left\{\mathrm{S}^{(j)}, \mathrm{E}^{(j)}, \mathrm{L}^{(j)}\right\}_{j=1}^{|D|}$ the dataset that we would like to create, with S the speech utterance (in English), E the corresponding transcription (in Lingala) and L the gold truth lingala translation.

3.1 Sources and structure

LiSTra is a systemic crawl of the new testament both at the jw.org for Lingala translation and bible.is for speech and English transcription. The bible is originally aligned by chapter and several websites provide audios waves of reading of the all bible in several languages. One big challenge with doing Automatic Speech research with the bible data in its original format is the alignment at the chapter, which usually is long and not suitable for ASR.

Automatic Speech Recognition (ASR) also known as Text-To-Speech (TTS) has been historically a close domain compare to others due to the expenses to train a fully working system and the

[4]https://deeplearningindaba.com
[5]http://www.datascienceafrica.org/

	LiSTra			
Text language Source	Split	Examples	Avg. text length	Total Unique Words
English (En)	train	23717	24.2712	13139
	test	5930	24.2076	7772
Text language Target	Split	Examples	Avg. text length	Total Unique Words
Lingala (ln)	train	23717	25.9165	16808
	test	5930	25.7489	8940
Speech Source	Split	Examples	Avg. audio length (seconds)	Total numb. hours
English (.wav)	train	23717	9.2880	61
	test	5930	9.2715	15

Table 1: Data statistics of LiSTra

difficulty that came with it, this lead to having only big techs companies working in that space.

In the next section, we will present our procedure to transform the data in the adequate format for Automatic Speech Translation (AST), from the web crawling step to the ready-to-use AST format.

3.2 Curation

The first step consists of scrapping the text and downloaded the audios files corresponding to the languages pair at study, English-Lingala in our case. We used the *English Standard Version - FCBH Audio Audio Non-Drama New Testament* from bible.is[6] and the *Biblia Libongoli ya Mokili ya Sika*[7] version for the Lingala version from the jw.org which will be used for the aligned translation[8].

The bible text being systematically organized by verse, make it perfect to keep the same alignment for automatic speech translation but the bottleneck remains the fact that all audios reading of the bible are only at book level with no way to manually split it at the verse level.

To split the chapter level reading waves at verse level we made use of the automatic segmentation service WebMAUSBasic of the Bavarian Archive for Speech Signals (BAS)[9] project similarly to (Boito et al., 2019). The code to perform this segmentation using a jupyter notebook can be found here Anonymous.

Given that the text is scrawled from two different websites (jw.org and bible.is) and in two different versions, we noticed inconstancy on some books that don't have the same number of verses and we decided to drop the concerned cases.

4 Experiments and Results

We have created what is at the best of our knowledge the first baseline for Automatic Speech Translation (AST) from English to Lingala, in both Cascade and End-2-End configuration[10].

4.1 Automatic Speech Translation: Cascade

The Cascade architecture is made of two separate models as described in figure1, a pre-trained Sirelo[11] Model and a traditional transformer-based Machine translation architecture which receive the output of the former one to perform Automatic Speech Translation.

[6] https://www.faithcomesbyhearing.com/audio-bible-resources/mp3-downloads

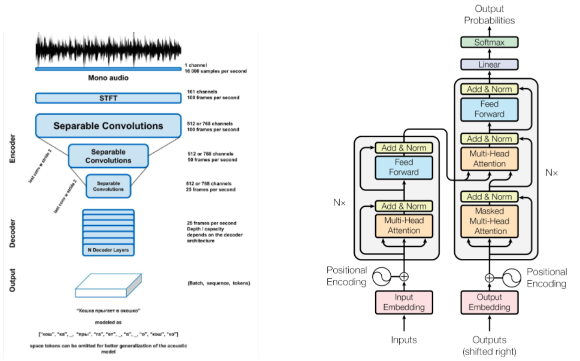
[7] https://www.jw.org/ln/Biblioteke/biblia/bi12/mikanda/matai/2/

[8] constrained by the licensing we have not released the audios files

[9] https://www.bas.uni-muenchen.de/Bas/BasHomeeng.html

[10] Anonymous

[11] https://github.com/snakers4/silero-models



(a) Silero Architecture (source: Silero AI Team)

(b) Standard Transformer for Machine Translation

Figure 1: Cascade Approach : Speech Recognition (a) + Machine Translation (b)

Sirelo Speech to text is among the recent efforts to bring the Imagenet moment to the field of speech recognition, the models we used have been trained on a proprietary dataset and have been reported to achieve performance that sometimes surpasses the state-of-the-art in some languages (Veysov, 2020).

The MT model[12] is based on the standard transformer architecture, but with a dimensionality of input and output of 256, refer on the original paper (Vaswani et al., 2017) as d_{model} and a inner-layer dimension d_{ff} of 512.

We pre-trained the Machine Translation model on the JW300 dataset (Agic and Vulic, 2020) and train further on LiSTra data. The recognized waves from silero are then fed into the trained MT to obtain our Speech translation output.

4.2 Automatic Speech Translation: end-2-end

In the end-2end setting, we used a transformer-based model3, that is made of one encoder and two decoders as shown in figure 2. This architecture has shown promising results recently e(Liu et al., 2020) specially due to the interaction between the recognition decoder and the translation decoder.

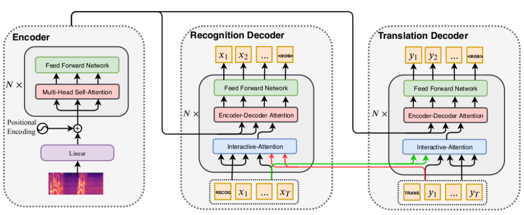


Figure 2: Synchronous AST Architecture (Liu et al., 2020)

[12]https://github.com/bentrevett/pytorch-seq2seq

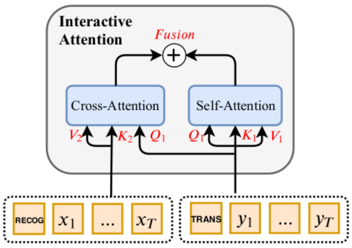


Figure 3: Interactive Attention (Liu et al., 2020)

The interactive attention sub-layer is basically the main revolutionary idea of this architecture, the intuition is to allow systematic information sharing between the transcription and the translation decoders. The right side of the Interactive Attention bloc is not very different from the vanilla attention formalism, but the difference is with the second bloc that queries from the gold translation.

The intuition is to provide direct context from the translation/recognition input to the "Cross-Attention" that will supply additional information to the recognition/translation decoder. The Interactive Attention box fuses the self-attention to the Cross-Attention using weighted addition but more complex fuse functions can be explored in future work.

Formally, the interactive attention can be written mathematically as follow :

$$\texttt{Attention\_transcription}(\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1) = softmax\left(\frac{\mathbf{Q_1 K_1}^T}{\sqrt{d_{k_1}}}\right)\mathbf{V_1} \quad (1)$$

$$\texttt{Attention\_translation}(\mathbf{Q}_1, \mathbf{K}_2, \mathbf{V}_2) = softmax\left(\frac{\mathbf{Q_1 K_2}^T}{\sqrt{d_{k_2}}}\right)\mathbf{V_2} \quad (2)$$

Where

- $Q_1,\ K_1\ V_1$ is the query, key, and value from the translation task, and $V_2\ K_2$ is the value, key of the transcription task respectively.
- $d_{k_1}\ and\ d_{k_2}$ is the dimension of the K_1 and K_2, respectively.

We can notice from the equation 1 that the hidden representation of the recognition task have as query the information for the translation ground truth, the final representation of the interactive attention will be written as :

$$\text{Interactive attention} = \text{Attention\_translation} + \lambda \times \ Attention\_transcription$$

With λ a hyper-parameter that allows controlling the amount of information shared between the two tasks.

	wait-1			wait-2			wait-3		
Architecture	WER ↓	BLEU (en) ↑	BLEU (ln) ↑	WER ↓	BLEU (en) ↑	BLEU (ln) ↑	WER ↓	BLEU (en) ↑	BLEU (ln) ↑
Pipeline[13]	8.27	84.90	13.92	x	x	x	x	x	x
End-2-End	8.06	84.40	26.45	**7.81**	**84.90**	**28.52**	7.87	84.73	26.99

Table 2: Results : Experimentation for different value of k

	vocab_src_size	vocab_tgt_size	train_steps	decode_alpha	gpu_mem_fraction
Transformer_params	30000	30000	80000	0.6	0.95

Table 3: LiSTra parameters, in addition to traditional transformer parameters

The prediction probability of both the translation and transcription can be formalized as

$$\log P(\mathrm{E} \mid \mathrm{S}, \mathrm{L}) = \sum_{i=0}^{N-1} \log p\left(e_i \mid e_{<i},\ \mathrm{S}, l_{<i}\right) \tag{3}$$

$$\log P(\mathrm{L} \mid \mathrm{S}, \mathrm{E}) = \sum_{i=0}^{N-1} \log p\left(l_i \mid l_{<i},\ \mathrm{S}, e_{<i}\right) \tag{4}$$

Our objective function is then expressed as

$$\mathrm{L}(\theta) = \sum_{j=1}^{|D|} \left(\log P\left(\mathrm{E}^{(j)} \mid \mathrm{S}^{(j)}, \mathrm{L}^{(j)}\right) + \log P\left(\mathrm{L}^{(j)} \mid \mathrm{S}^{(j)}, \mathrm{E}^{(j)}\right)\right) \tag{5}$$

Given that the Text to Speech task is often more difficult than Automatic Speech Recognition similarly to (Liu et al., 2020) we used the $wait - k$ policy approach that basically allows waiting for a certain time to allow the recognition decoder to transcribe some words before it can start translating. Table 3 summarized our experiments with different values of k and we empirically realized that we have better performance for $k = 2$.

The End-2-End architecture was pre-trained for 50000-steps on TED_Speech_Translation[14] which was constructed by collecting speech and corpus from TED talks and then fine-tuned on LiSTra, this is arguable the reason we have the recognition decoder with better performance than the translation one, pre-training the translation decoder is left for future work.

As observed in Table 3 for $k = 2$ we have a better Word Error Rate (WER) and BLEU score for both the recognition and translation decoder, in other words slowing down the translation decoder with a factor of 2 gives the translation decoder more context to provide better performance.

Compare with the Machine Translation results from masakhane (Orife et al., 2020) our translation decoder is performing poorly, probably because we don't have enough training examples and need to pre-trained the translation decoder separately to increase its performance. One probable direction to increase and produce unbiased data may be the use of platforms like Mozilla common voice or similar technology that can use a human-in-the-loop approach to collect qualitative data.

5 Conclusion

In this work, we presented LiSTra, the first dataset for automatic speech translation from English to Lingala, and a full pipeline to allow researchers working on low-resource languages to create a similar dataset for their language. Despite the dataset been bias toward religious languages this can serve as a starting dataset for proof-of-concept and can, later on, be improved with additional data.

In addition, we reported baselines in both Pipeline and End-2-End architecture and concluded that the End-2-End architecture performs quite well despite the limited amount of data.

For future work, one could extend LiSTra with other data sources, pre-train both the recognition and the translation decoder separately which may probably lead to better performances overall.

[14]http://www.nlpr.ia.ac.cn/cip/dataset.htm

References

Željko Agic and Ivan Vulic. 2020. Jw300: A wide-coverage parallel corpus for low-resource languages.

technology (2013-08-23). Arthur, Charles; editor. "tech giants may be huge, but nothing matches big data". *The Guardian. ISSN 0261-3077.*

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*.

Marcely Zanon Boito, William N Havard, Mahault Garnerin, Éric Le Ferrand, and Laurent Besacier. 2019. Mass: A large and clean multilingual corpus of sentence-aligned spoken utterances extracted from the bible. *arXiv preprint arXiv:1907.12895*.

David M Eberhard, Gary F Simons, and Charles D Fennig. 2019. *Ethnologue: Languages of Asia*. SIL International.

Yuchen Liu, Jiajun Zhang, Hao Xiong, Long Zhou, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2020. Synchronous speech recognition and speech-to-text translation with interactive decoding. In *AAAI*, pages 8417–8424.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Iroro Orife, Julia Kreutzer, Blessing Sibanda, Daniel Whitenack, Kathleen Siminyu, Laura Martinus, Jamiil Toure Ali, Jade Abbott, Vukosi Marivate, Salomon Kabongo, et al. 2020. Masakhane–machine translation for africa. *arXiv preprint arXiv:2003.11529*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Alexander Veysov. 2020. Toward's an imagenet moment for speech-to-text. *The Gradient*.

Wikipedia contributors. 2020. Lingala — Wikipedia, the free encyclopedia. [Online; accessed 30-October-2020].

Chengqing Zong, Taiyi Huang, and XU Bo. 1999. Technical analysis on automatic spoken language translation systems. *Journal of Chinese Information Processing*, 13(2):55–65.