Lingua: Addressing Scenarios for Live Interpretation and Automatic Dubbing

Nathan J. Anderson njanders@cs.cmu.edu
Language Technologies Institute, Carnegie Mellon University

Caleb Wilson caleb.wilson@byu.net
Shift Technology

Stephen D. Richardson srichardson@cs.byu.edu
Department of Computer Science, Brigham Young University

Abstract

Lingua is an application that can perform near-real-time interpretation of video recordings and live speeches as well as synchronized automatic video dubbing. It has been developed and is being piloted at the Church of Jesus Christ of Latter-day Saints. The system pipeline includes customized automatic speech recognition (ASR) and machine translation (MT) components. A script may be uploaded in advance to improve the translation accuracy and decrease the lag behind the speaker, while flexibly handling instances when the speaker goes off script. The speed of the text-to-speech (TTS) outputs are dynamically adjusted to match the rate of speech. Lingua is currently capable of interpreting English to 38 other languages.

1 Introduction

Lingua is an automatic speech-to-speech (STS) interpreter and video dubber that is being developed and piloted at the Church of Jesus Christ of Latter-day Saints. This application is suitable for interpreting live speeches and video recordings on-the-fly from English to 38 other languages. It can also assist in creating exactly synchronized audio tracks for the professional dubbing of video recordings.

A common pitfall of STS systems built on a cascaded architecture is the propagation of errors from one component of the pipeline to the later components (Sperber et al., 2019). For instance, if the ASR module registers "ice cream" as "I scream", the MT module is typically incapable of rectifying the mistake, and it will indiscriminately translate the incorrect transcription. Lingua affords the possibility of course corrections by allowing the user to upload a script of the speech in advance. It then uses a dynamic programming algorithm to align the ASR transcription with the official script on the phonemic level and override the ASR when a likely match is found. Not only does this feature improve the accuracy of downstream outputs, it can also reduce the *décalage* or lag between the original speaker and the live interpretation (Riccardi, 2005). Rather than waiting for the speaker to finish their sentence, Lingua can detect which sentence is being uttered early on and get a head start on producing the appropriate outputs.

Lingua is intended to facilitate the distribution of multimedia content into languages for which it would otherwise be cost-inefficient to provide manual interpretation and dubbing. It is of special interest for improving the accessibility of live events and undubbed video recordings for speakers of underserved languages when interpreters are not readily available. It can also

accelerate the process of dubbing previously recorded content, including not only speeches but also short films and other multimedia presentations.

2 Related Work

There has been significant research over the past few years in the field of automatic speech-to-speech translation and more specifically in its application to automatic video dubbing (AVD). For example, improvements detailed by Amazon researchers include the use of human-like, customizable neural voices, integration with Neural MT, adjustments to the duration and prosody of translated utterances, and the handling of background noise and reverberation (Federico et al., 2020; Lakew et al., 2021).

Several companies are now joining the fray with their own research and product development. AppTek recently announced plans to release an AVD tool that includes speaker diarization, limited prosody transfer, and basic utterance length control, with planned improvements in transfer of emotion, utterance length adjustment, and simulated lip movement (Di Gangi et al., 2022). Other companies are already providing self-serve dubbing services on the internet. For example, both Maestra Video Dubber (Maestra, 2021) and Aloud (Google, 2022) provide capabilities for users to upload video files and corresponding text files. If the latter are not uploaded, the systems can transcribe the videos to produce text. Users can then edit the transcriptions and their corresponding, MT-produced translations, selecting from available synthetic voices, and making needed modifications to ensure acceptable video dubbing quality.

3 Speech-to-Speech Pipeline

Lingua is similar to these systems in that it can also perform automatic dubbing given video files and, optionally, corresponding text files in either SRT or a proprietary XML format. It currently uses Microsoft's Cognitive Services APIs in a traditional ASR – MT – TTS pipeline. The ASR component has been customized with 50+ hours of audio from Church speeches and their aligned human transcriptions, resulting in a reduction in word error rate from 7.2% to 3.3%. The MT systems have also been customized using hundreds of thousands to millions of sentence pairs from the Church's extensive translation memories, resulting in an increase of BLEU scores over generic MT baselines between 6 and 22 points, with an average increase of 13 points. The TTS component has not been customized, but the switch to neural voices for all languages in early 2021 was a significant, albeit subjectively evaluated, improvement.

3.1 Real-Time Interpretation

Lingua's unique contribution is that it can operate in near-real-time with a slight delay of a few seconds on average to perform ASR, MT, and TTS of live speeches and videos, while also providing additional processing to produce synchronized automatic video dubbing. If no source-language script is provided, the spoken translations are exactly as recognized by ASR and translated by MT. However, if a script is provided, Lingua uses a fuzzy matching algorithm to align the recognized ASR segments with the corresponding script segments and then passes the latter to the MT component for translation and subsequent TTS. If a human translation is also provided, it passes that translation directly to TTS.

An important feature of Lingua is that it can switch between these three modes dynamically, as illustrated in Figure 1. Thus, if a script is provided but the speaker makes off-script comments, the unmatched ASR segments are translated by MT and the spoken translations are generated. As soon as the speaker comes back on script, matching continues, and the script's sentences are passed to MT. This results in more accurate MT output based on the well-formed sentences in the script while providing reasonable translations of interjected comments. If human translations are provided for some languages and not for others, the result is high-quality

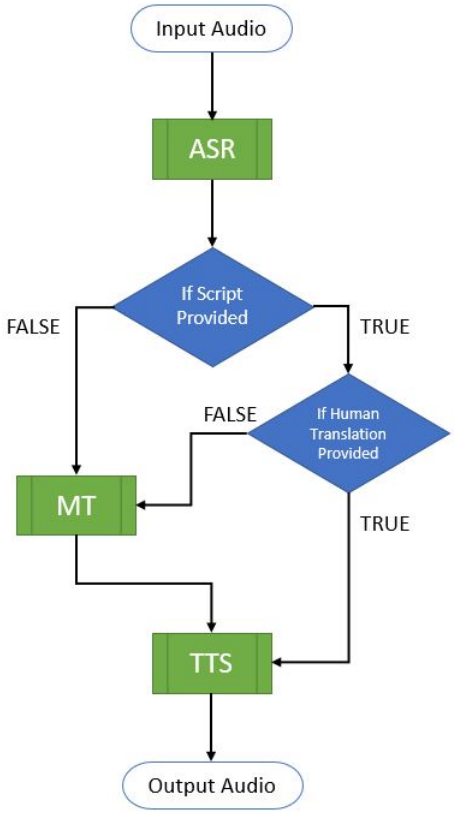

Figure 1: Flowchart of Lingua's modes

human translations for the former languages and good quality machine translations for the latter ones.

3.2 Automatic Dubbing

As Lingua processes incoming audio, translating it directly, using a monolingual script, or using a multilingual script with aligned translations, it records the times at which utterances started as supplied by the ASR. These timestamps can be exported into an SRT file along with the source language transcriptions of the speech. When generating a synchronized dub, the speech timestamps and machine or human translations are passed to an audio export thread, which generates an audio file containing spoken translations occurring exactly in sync with the corresponding spoken English. For each timestamped utterance, the audio file is filled (with silence) up to the time at which it was uttered, then Lingua uses TTS to obtain audio in the target language to write to the audio file. If the target audio speech is too long to fit in the same time as the source utterance, its speed is increased before it is written to the file. While generating this audio file, the translated speech and its timestamps are exported as another SRT file, which may also be used to add subtitles to the dubbed video. The audio file can then be mixed with the original video to provide fully synchronized dubbing.

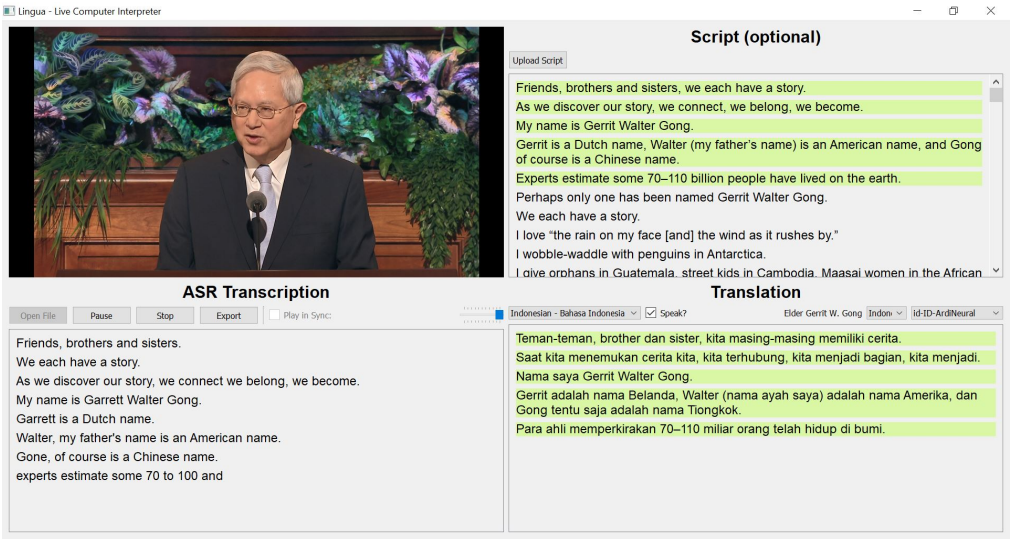

Figure 2: Screenshot of the Lingua interface in action. As the ASR transcription updates in the bottom left quadrant, Lingua uses the fuzzy matcher to align it with the script in the top right quadrant. The translations are displayed in the bottom right.

4 Fuzzy Matching Classifier

4.1 Algorithm

The fuzzy-matching algorithm continuously compares the incoming ASR partial transcriptions with a window of the next n utterances in the speech. It converts both the transcriptions and the official script to phonemic representations and uses a dynamic programming algorithm to compute the Levenshtein edit distance between the current utterance and each of the candidate sentences. To avoid penalizing longer sentences, the initial pass truncates the candidate transcriptions to the same length as the partial transcription. This truncation does not necessarily occur in the optimal location, as the ASR transcription may contain more or fewer phonemes than the correct match. Therefore, whenever the algorithm determines that the final phoneme in the alignment is an insertion or a deletion, it iteratively shifts the truncation location until the final phoneme is a match or a substitution. See Figure 3 for a simplified demonstration of this alignment algorithm.

Each alignment with a candidate sentence is rated according to a cost formula:

$$cost = Lev/Phon \tag{1}$$

where Lev is the Levenshtein distance and $Phon$ is the number of phonemes in the ASR transcription. A predefined threshold determines the maximum cost that qualifies as a match.

4.2 Evaluation

We developed a test set to evaluate the performance of the fuzzy matching classifier. This set contains 62 minutes of speech that has been hand-annotated with time stamps and gold-standard labels. It includes 6 different speakers, representing multiple age ranges, genders, and nationalities. These speakers rarely went off script, so we artificially increased the difficulty of the test set by randomly deleting, adding, and shuffling approximately 10% of the lines in the scripts.

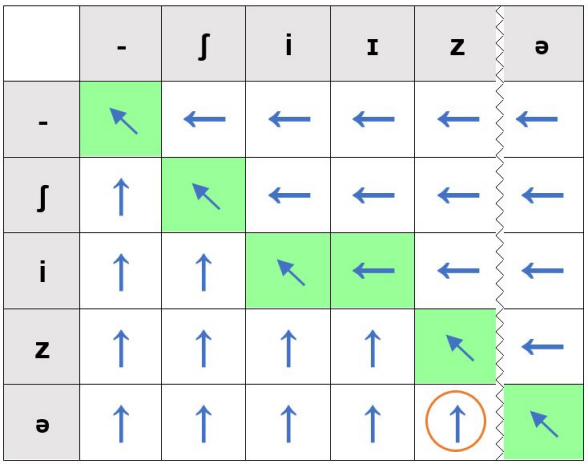

Figure 3: Toy example of fuzzy matching dynamic algorithm. The partial ASR transcription ("She's a...") is displayed on the first column, and the candidate sentence from the script ("She is a good person.") is on the top row. The candidate sentence was originally truncated to 4 phonemes to match the ASR transcription, along the jagged line. However, the final backtrace (circled) was not a match/substitution, so the window was expanded an additional column. As the new column ends in a match, this matrix is considered the optimal alignment for these sentences.

The relevant performance metrics are:

1. F1 Score: Harmonic mean of precision and recall. To calculate these measures, we defined "true positives" as utterances that are correctly matched to the script, "false positives" as utterances that were assigned to an incorrect sentence in the script, and "false negatives" as utterances that were incorrectly not matched to any sentence in the script.

2. Average Lag: The time (in seconds) that it takes to identify the correct sentence from the script. We count the time from the start of the utterance to the moment the decisive phoneme is uttered, disregarding the time subsequently spent computing the answers, because calculation speeds are largely dependent on the hardware.

See Table 1 for the baseline metrics.

Metric	Score
Precision	0.9529
Recall	0.9701
F1	0.9614
Avg. Lag	0.72 s

Table 1: Fuzzy matching classifier baseline performance metrics

4.3 Hyperparameters

The fuzzy matching classifier relies on three hyperparameters: 1) **Beamwidth**: the number of candidate sentences considered at a time, 2) **Phoneme threshold**: the minimum number of

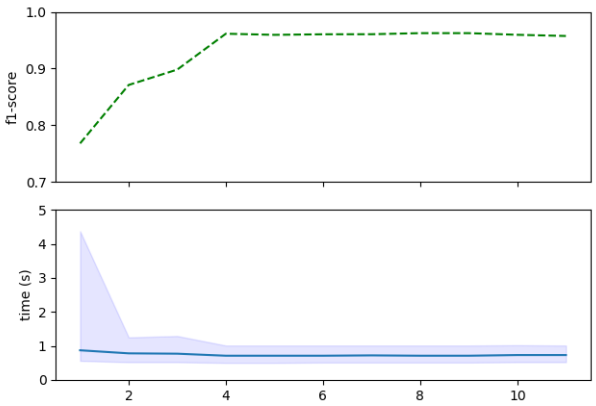

Figure 4: Beamwidth

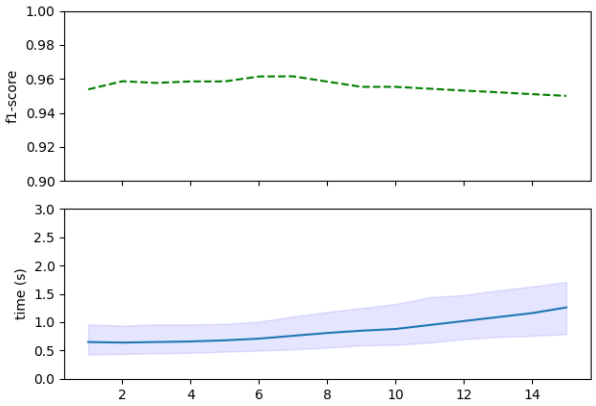

Figure 5: Minimum phoneme count threshold

phonemes required to make a decision, and 3) **Cost threshold**: the maximum allowable cost for a match.

We tuned these parameters to maximize the F1 score and minimize the décalage. Figures 4 - 6 show the effect of manipulating each parameter while holding the others constant. Within each figure, the x-axis represents values for the relevant parameter. The top plot tracks the F1 score, and the bottom plot displays the average lag time and shades the interquartile range.

Increasing the beamwidth increases the time and space requirements to complete the computation. It could theoretically increase the likelihood of false positive matches from later in the speech, although this error did not occur frequently in our tests. On the other hand, it is also possible for the window to be too small so that if the speaker skips a portion of the planned speech, the fuzzy matcher won't be able to identify the current location and it will have to default to MT.

As expected, increasing the phoneme threshold results in a greater lag behind the speaker. We had hypothesized that it would also improve the overall accuracy, as the model would gather

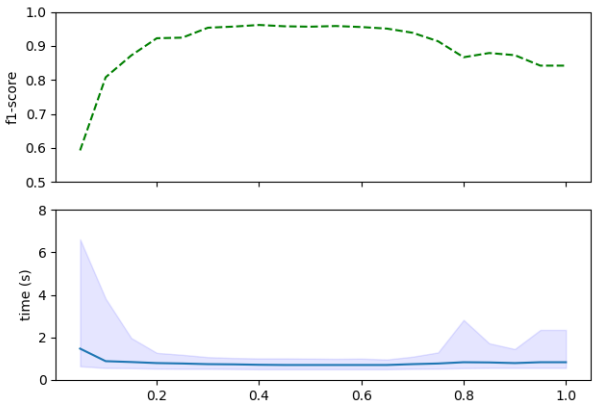

Figure 6: Maximum cost threshold

additional information before making an informed conclusion. However, our empirical tests revealed that this is only true to a certain point, after which the F1 score gradually decreases. Apparently, the model sometimes loses confidence in a correct decision as errors crop up in additional ASR partial transcriptions.

Extreme cost threshold values result in lower performance for opposite reasons. High values are cause the model to become too stringent, rejecting true matches due to noisy transcriptions or slight changes in the speaker's wording. Low values are too accepting, so the model requires very little evidence before making a decision.

The hyperparameters we selected based on these results are shown in Table 2.

Param	Value
Beamwidth	4
Phoneme Thresh	6
Cost Thresh	0.4

Table 2: Selected hyperparameters

5 Future Work

The dynamic speed adjustment for the TTS sometimes results in dubbing that is unnaturally fast. Additional hyperparameter tuning may help to distribute the dubbing more evenly. However, this strategy is unlikely to resolve the problem entirely, as translations are generally longer than the source (Frankenberg-Garcia, 2009). Manual dubbings often intentionally abbreviate the translations to improve the alignment with the original. Methods similar to those described in Federico et al. (2020) and Lakew et al. (2021) may be implemented to optimize the MT to generate outputs that are roughly equivalent in length to the input.

To prepare Lingua for deployment in real-world settings, we will need to run additional user studies. These experiments will be essential to assess the subjective acceptability of the outputs across all of the target languages.

6 Conclusion

In this paper, we described Lingua, an application that is capable of interpreting live speeches and creating synchronized dubbings for videos. It mitigates the shortcomings of cascaded STS systems by following an optional uploaded script, although it can revert to the default cascade if the speaker goes off script. We assessed the performance of the fuzzy matching classifier, and we found that it achieves F1 scores > 0.95 on a difficult test set. We discussed the primary hyperparameters and demonstrated how they affect the performance of the fuzzy matching algorithm. Finally, we proposed future avenues of research to improve the TTS component and prepare Lingua for deployment.

We anticipate that Lingua will increase accessibility to church multimedia content for underserved linguistic communities. It will decrease the time, cost, and expertise required to perform live interpretation and produce professional dubbings.

References

Di Gangi, M., Rossenbach, N., Pérez, A., Bahar, P., Beck, E., Wilken, P., and Matusov, E. (2022). Automatic video dubbing at AppTek. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 349–350.

Federico, M., Enyedi, R., Barra-Chicote, R., Giri, R., Isik, U., Krishnaswamy, A., and Sawaf, H. (2020). From speech-to-speech translation to automatic dubbing. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 257–264.

Frankenberg-Garcia, A. (2009). Are translations longer than source texts. *A corpus-based study of explicitation In: Beeby, A., Rodríguez P., & Sánchez-Gijón, P.(eds.) Corpus use and learning to translate (CULT): An Introduction. Amsterdam & Philadelphia: John Benjamins*, pages 47–58.

Google (2022). Aloud. `https://aloud.area120.google.com/`, Accessed 07/25/2022.

Lakew, S. M., Federico, M., Wang, Y., Hoang, C., Virkar, Y., Barra-Chicote, R., and Enyedi, R. (2021). Machine translation verbosity control for automatic dubbing. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7538–7542. IEEE.

Maestra (2021). Maestra Video Dubber. `https://maestrasuite.com/video-dubber`, Accessed 07/25/2022.

Riccardi, A. (2005). On the evolution of interpreting strategies in simultaneous interpreting. *Meta: Journal des traducteurs/Meta: Translators' Journal*, 50(2):753–767.

Sperber, M., Neubig, G., Niehues, J., and Waibel, A. (2019). Attention-passing models for robust and data-efficient end-to-end speech translation. *Transactions of the Association for Computational Linguistics*, 7:313–325.