CoSSAT: Code-Switched Speech Annotation Tool

Sanket Shah **Pratik Joshi** **Sebastin Santy** **Sunayana Sitaram**
Microsoft Research, Bangalore, India
{t-sansha, t-prjos, t-sesan, susitara}@microsoft.com

Abstract

Code-switching refers to the alternation of two or more languages in a conversation or utterance and is common in multilingual communities across the world. Building code-switched speech and natural language processing systems are challenging due to the lack of annotated speech and text data. We present a speech annotation interface CoSSAT, which helps annotators transcribe code-switched speech faster, more easily and more accurately than a traditional interface, by displaying candidate words from monolingual speech recognizers. We conduct a user study on the transcription of Hindi-English code-switched speech with 10 annotators and describe quantitative and qualitative results.

1 Introduction

Code-switching is a phenomenon that occurs in multilingual societies wherein speakers who are fluent in two or more languages switch between these languages in the same conversation or an utterance. Code-switching is a challenging problem for speech and natural language processing systems to handle due to the lack of manually annotated data and resources. However, due to the ubiquitous nature of code-switching in speech and text produced by multilingual speakers, it is an important problem for speech and NLP systems to tackle.

Automatic Speech Recognition (ASR) is used by a variety of systems to convert speech to text for further processing. Deep Neural Network (DNN) based systems have increased the accuracy of ASR systems to match human-level performance. However, these gains are only obtained in high-resource languages that have thousands of hours of manually transcribed speech data. Code-switched languages suffer from a lack of manually annotated training data, as described in (Sitaram et al., 2019), with the largest publicly available speech corpus in Mandarin-English being 63 hours long (Lyu et al., 2015).

In cases where the two languages being mixed are in different scripts, the transcriber needs to switch between two scripts while annotating an utterance. This paper introduces an interface which assists in the transcription of code-switched Hindi-English speech data by displaying candidate words generated by monolingual Hindi and English speech recognizers, without the need for a code-switched ASR. We present quantitative and qualitative results from a user study with 10 users who use our proposed interface as well as a traditional typing-only interface for transcribing code-switched speech.

2 Related Work

Using hypotheses produced by an ASR system is a common approach used to reduce human effort in transcribing speech (Sperber et al., 2016). However, this approach often induces a bias amongst the annotators while transcribing text (Levit et al., 2017). To mitigate this bias, we do not provide the hypothesis as a suggested transcription, but rather provide a collection of suggested words for the annotators to choose from. We leverage a combination of monolingual ASRs rather than a code-switched ASR for our task. Our work is inspired by efforts in Spoken Term Detection (STD) for Hindi-English code-switched speech, in which (Shah and Sitaram, 2019) use post-processing techniques to improve hypothesis produced by monolingual ASR for code-switched speech. Similarly for Chinese-English code-switched speech, (Shan-Ruei You et al., 2004) combine scores from monolingual Chinese and English ASRs to determine the most probable output. In contrast, for this work, we neither determine a single combined

Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP, pages 48–52
Hong Kong, China, November 3, 2019. ©2019 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

ASR hypothesis nor do any post-processing on the ASR hypothesis, but rather use the output of the two recognizers to display candidate words for annotation purposes.

3 Methodology

Given a speech utterance, we generate an ordered sequence of candidate code-switched words using monolingual speech recognizers. Our method consists of two main steps, (1) Dynamic Audio Segmentation, (2) Combining ASR hypotheses.

3.1 Dynamic Audio Segmentation

Due to the low accuracy of the monolingual speech recognizers on code-switched input, it is important to find segments in the audio where the monolingual recognizers have high confidence. To enable this, we automatically segment the audio according to ASR confidence. This audio segmentation task can be formulated as an optimization problem for a given set of possible boundaries. We try to optimize the segment size based on the confidence scores of the monolingual ASRs on code-switched speech. We start with a segment of size 0.5 seconds from the beginning of the audio, and pass this audio segment through the monolingual Hindi and English ASRs. Each ASR provides an utterance (or audio chunk, in this case) level confidence value in the range of 0-1.

If the confidence values given by both recognizers are less than 0.3, we increase the segment size for that particular segment by 0.25 seconds at the beginning and end of the audio. We repeat the process until one of the recognizers outputs a confidence score of more than 0.3. We then select the next segment of 0.5 seconds having an overlap of 0.25 seconds with the current optimal segment. The entire process is repeated until the entire audio is segmented. At the end, we combine all the hypotheses generated for each chunk to create an utterance level hypotheses for each ASR.

3.2 Combining ASR hypothesis

We use off-the-shelf monolingual Hindi and Indian English ASRs for decoding speech. To measure the performance of the ASRs on code-switched speech, we test them on an in-house conversational speech corpus consisting of 52k Hindi-English mixed utterances. The corpus is transcribed using the Devanagari script for Hindi words and the Latin script for English words. The English ASR gives a Word Error Rate (WER) of 80% and Hindi ASR gives a WER of 48% on the corpus. The high error rate of both ASRs can be attributed to the difference in script between the reference and hypotheses words as well as the poor performance of monolingual ASRs at code-switch points.

We hypothesize that each ASR will recognize a set of words in the given audio segment, and the collection of the sets will contain all the words present in that particular audio segment. We conduct a quantitative evaluation on 10k code-switched utterances by passing them through both monolingual ASRs and checking whether the ground truth words are present in either of the recognition hypotheses. We obtain a recall of 0.84, which indicates that most words present in the utterance are also present in the output of the two recognizers. We pass each segment obtained through dynamic chunking through Hindi and English monolingual ASRs respectively to obtain two ASR hypotheses for each segment, which we then combine to form utterance level hypotheses.

4 Interface Overview

The annotation interface as shown in figure 2 consists of a button to play audio and a text box. We display the predicted words in a time-linear fashion as clickable blocks. As the user clicks on each word button, all the buttons before and including it becomes disabled, to allow the user to easily focus on the progression of the transcriptions. If the user presses the backspace or attempts to remove certain words, the respective disabled word buttons appear again. Users also have an option of typing out the transcription if they wish to, in both the languages.[1]

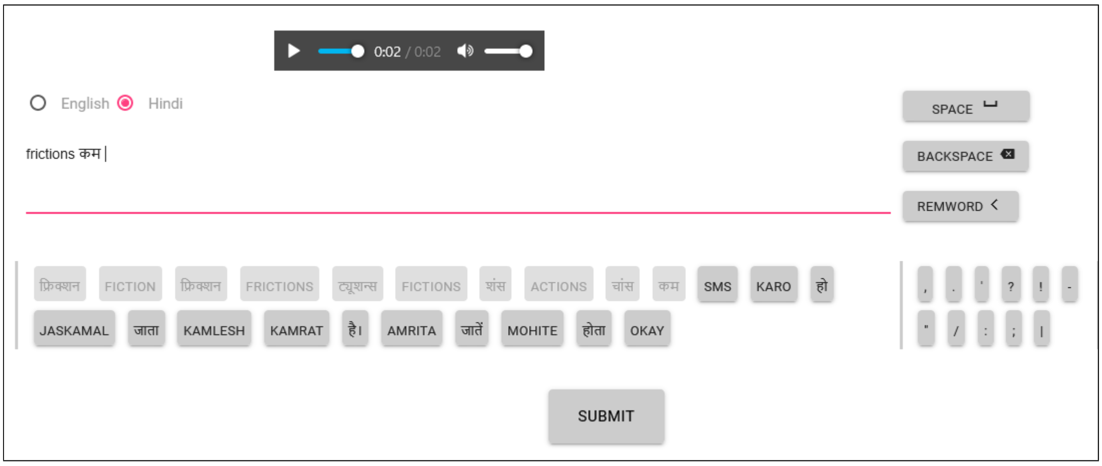
If the user wishes to use the keyboard, we also provide quick keyboard shortcuts to improve the efficiency of transcription. These shortcuts allow the user to play/pause the audio, and toggle scripts easily. More details about the interface can be found in the appendix section.

5 Experiments

5.1 Setup

We performed a user study to evaluate the efficacy of our system. We measured transcription quality, annotator effort and the net time taken to transcribe utterances. We compared our annotation

[1] The transliteration is powered by Google's Input Tools API



frictions कम |

Figure 1: CoSSAT (Code-Switched Speech Annotation Tool)

tool (CoSSAT) against a baseline system, where no ASR hypothesis is shown and the annotators are expected to type out the entire transcription. 10 users annotated 14 code-switched speech utterances. All participants were Hindi-English bilinguals and had no knowledge about the system before conducting the study.

We implemented multiple measures to reduce biases during the annotation task. 4 utterances out of 14 that each user is shown were practice exercises for the annotators to get used to the interface and were not used for the final evaluation. The sequence of the interface (baseline vs. CoSSAT) displayed changed for each user such that each utterance and interface was paired at least five times.

Users were asked to listen to the audio displayed on the page and could play the audio as many times as they wanted. They were asked to transcribe all audible words in the audio sample except speech fillers (e.g., “uh” and “eh”). The users were required to enter the tokens in the script to which they belong - Hindi in Devanagari script and English in Latin script, although this distinction was difficult to make sometimes due to the prevalence of borrowing between the two languages. We did this instead of having all tokens in one script to ensure correctness of transcribed Hindi and English tokens. Often, the transcribing of tokens in a different script can cause certain tokens to be transcribed in different ways (for example), which would have resulted in a cumbersome and misleading evaluation process, even with post-transliteration.

5.2 Quantitative Evaluation

To evaluate our system, we used the following three metrics (1) Transcription Quality (2) Annotation Speed (3) Annotation Effort. For every utterance, we had 10 transcriptions, 5 transcriptions from our proposed interface, and 5 from the baseline interface.

5.2.1 Transcription Quality

Transcription quality was determined by computing word error rate (WER) using a standard procedure[2], using the transcriptions present in our in-house dataset as the gold standard. We calculated WER for the transcriptions created by users using our system as well as for the transcriptions created using the baseline approach. From table 1 we see that transcriptions created using our system have a WER of 19.7%, while the number is much higher for the baseline at 34.74%. After analysing transcripts which had high WER, we noticed that for words where the ASR hypothesis was not present, errors could be attributed to spelling variants, spelling errors, hyphenated words, and grammatical errors. Besides these errors, in the baseline method users made errors in phonetically similar phrases like “of score” instead of “of course”.

Another major source of errors was the use of a different script for transcribing a borrowed word, which meant that the annotator used the Hindi script to transcribe a word that was in the Latin script in the reference transcription or vice versa. This is a very challenging problem for transcription of code-switched speech where the two languages are written in different scripts, as it is difficult to make the distinction between loan words and code-switching (Bali et al., 2014).

[2] https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation

To make the comparison fair, we also computed a relaxed WER by converting the transcriptions into phoneme sequences by running a Grapheme to Phoneme (g2p) system on all words and post-processing phoneme sequences. We divided phonemes into classes based on phonetic features and treated each phoneme in a single class as equivalent. If the phoneme classes of all phonemes in a word was the same as those in the reference word, it was treated as a match. This helped take care of minor spelling errors and variants such as long/short vowels and nasalization. Relaxed WER for both systems are reported in table 1. We observe that the relaxed WER numbers for both techniques are significantly lower than the baseline. Crucially, the CoSSAT WER is even lower than the relaxed WER of the baseline, which shows that our interface helps even if we discount the fact that it helps users select the correct spelling variation of a word.

Metrics	CoSSAT	Baseline
WER	19.7%	34.74%
Relaxed WER	9.3%	25.6%

Table 1: WER and relaxed WER for measuring Quality of Transcriptions

5.2.2 Annotation Speed

In the case for transcription task using CoSSAT, we recorded time taken by the user to transcribe from the moment the user clicks on the first word or clicks on the text-box provided to the moment the user clicks submit. In case of the baseline system, we recorded time from the moment the user clicks on the text-box to the moment the user clicks submit. We normalized the time recorded for each audio using the formula (A).

$$\text{Normalized Time } (NT) = (\tfrac{t}{TT}) \text{ ——— (A)}$$

where t is the time taken for transcribing the audio by user X and TT is the total time taken by user X to transcribe all utterances. Figure 2 shows a plot of NT v/s utterance length. We observe that for utterance having ground truth transcriptions of 50 characters or less, CoSSAT takes less time for transcription but for longer utterances, the baseline system is faster. This might be attributed to the fact that longer utterances led to a larger set of word hypothesis resulting in more time for visual search of the tokens across the interface. We intend to address this issue by weeding out improbable suggestions based on confidence and language model scores in future work.

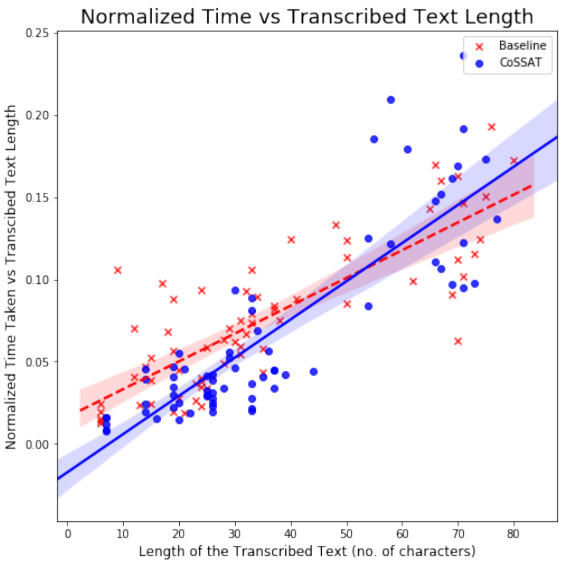


Figure 2: Annotation Speed Plot for each audio. Y axis is the Normalized Time Taken for the utterance. X axis is number of characters present in the utterance. Red colour (cross) is the Baseline system. Blue (dots) colour is CoSSAT.

5.2.3 Annotation Effort

One way to measure annotation effort is to measure the number of keystrokes and mouse clicks. The CoSSAT system resulted in 8 keystrokes and 8 mouse clicks on average, while the baseline system had 57.1 keystrokes and 5.4 mouse clicks. This is explained by the fact that the annotators relied on typing for the baseline interface and clicking on candidate words for the CoSSAT interface, however, overall, the total annotation effort was much lower for the CoSSAT system.

5.3 Qualitative Evaluation

In addition to the metrics collected during the study, users were asked rate their experience on both the interfaces. Questions consisted of rating each system from 1 (worst) to 5(best), on criteria such as Convenience (how easy it was to use), Speed (how fast they felt they could annotate), User-Friendliness (how simple it was to understand the interface), and Error Robustness (how much each system prevented them from making annotation errors). In all cases, the ratings were higher for CoSSAT than the baseline system. Finally, we asked them which system they would prefer using as a potential speech annotator tool. 7 out of 10 annotators said they preferred CoSSAT over the baseline system.

We also asked annotators for feedback and suggestions. One suggestion was to put larger sized or bold buttons for words that had higher probability according to the ASR confidence. Another suggestion was to show candidate words incrementally, rather than all at once. We plan to take this feedback into account while creating the next version of our tool.

6 Conclusion

In this paper, we propose an annotation tool for transcribing code-switched speech, which makes use of dynamic audio chunking and combines ASR hypotheses from two monolingual ASR systems to present candidate words to annotators. We compare our tool to a baseline system where the user has to type the entire transcription using two scripts and find that our proposed system performs better in terms of transcription quality, speed and annotation effort in a user study conducted with 10 annotators. Annotators report that our system is faster, easier to use, more user-friendly and more robust to annotation errors.

In this work, we present the hypotheses from both monolingual ASRs as two-word streams. In future work, we plan to create an aligned structure such as a word lattice and show candidate words to users as they are annotating the utterance instead of all at once. We also plan to collapse cross-transcribed borrowed words in both languages into a single variant using statistics from corpora, so that annotators can be more consistent in annotating such words.

Since our system does not rely on the existence of a code-switched ASR system, it can be used to bootstrap data collection for a code-switched language pair for which monolingual ASRs exist. This can help collect transcribed speech data faster, which can, in turn, help build better code-switched ASR systems.

7 Acknowledgments

We would like to thank all the reviewers' for their valuable comments. We would also like to thank all the users who participated in the user-study.

References

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Michael Levit, Yan Huang, Shuangyu Chang, and Yifan Gong. 2017. Don't count on asr to transcribe for you: Breaking bias with two crowds. In *Proc. Interspeech 2017*, pages 3941–3945.

Dau-Cheng Lyu, Tien-Ping Tan, Eng-Siong Chng, and Haizhou Li. 2015. Mandarin–english code-switching speech corpus in south-east asia: Seame. *Language Resources and Evaluation*, 49(3):581–600.

Sanket Shah and Sunayana Sitaram. 2019. Using monolingual speech recognition for spoken term detection in code-switched hindi-english speech. In *ICDM 2019 Workshop on Multilingual Cognitive Services*.

Shan-Ruei You, Shih-Chieh Chien, Chih-Hsing Hsu, Ke-Shiu Chen, Jia-Jang Tu, Jeng Shien Lin, and Sen-Chia Chang. 2004. Chinese-english mixed-lingual keyword spotting. In *2004 International Symposium on Chinese Spoken Language Processing*, pages 237–240.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Matthias Sperber, Graham Neubig, Satoshi Nakamura, and Alex Waibel. 2016. Optimizing computer-assisted transcription quality with iterative user interfaces. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1986–1992, Portorož, Slovenia. European Language Resources Association (ELRA).