Using Monolingual Speech Recognition for Spoken Term Detection in Code-switched Hindi-English Speech

Sanket Shah
Microsoft Research India
Bangalore, India
t-sansha@microsoft.com

Sunayana Sitaram
Microsoft Research India
Bangalore, India
Sunayana.Sitaram@microsoft.com

***Abstract*—Code-switching is the alternation of two or more languages in a single utterance or a conversation and is prevalent in multilingual communities all over the world. Spoken Term Detection (STD) is the task of detecting a given word or phrase in audio. STD has applications in audio indexing and mining. In this work, we explore Spoken Term Detection for code-switched conversational Hindi-English speech. Code-switching provides various challenges to this problem, including, 1. lack of training data to build robust code-switched Automatic Speech Recognition (ASR) systems, 2. non-standardized transcription due to borrowing and cross-transcription, 3. presence of translated or code-switched variants of the terms. In this work, we assume that a code-switched ASR System for Hindi-English does not exist, and make use of only a monolingual Hindi ASR to retrieve audio containing Hindi and English keywords. We use various techniques to normalize the output of a monolingual ASR system. We evaluate our techniques using Term Weighted Value (TWV) and find that phonetic matching of the query and ASR hypotheses at the utterance level is the most promising approach.**

***Index Terms*—Spoken Term Detection, Code-Switching, Code-Mixing, Keyword Spotting, Low-Resource**

I. Introduction

Code-switching or mixing is the alternation of two or more languages in a single conversation or utterance respectively. Both these phenomena occur along with borrowing in multilingual communities across the world. Handling code-switching (or mixing, which we use interchangeably henceforth) is a major challenge for Speech and NLP systems that interact with multilingual users. Spoken Term Detection (STD), sometimes also referred to as Keyword Spotting (KWS) is the task of detecting an occurrence of a word or phrase in a corpus of audio. The word or phrase may be specified textually or using an audio example, known as Query-by-Example KWS. STD has many applications, including audio indexing and mining.

Code-switched STD poses many challenges that are sometimes found in monolingual settings but may be magnified due to the presence of multiple languages. For example, the query term may be present as a translated or code-switched variant in the audio. In cases where an Automatic Speech Recognition (ASR) system is used for STD, a code-switched ASR may not exist for the language pair under consideration or may be difficult to build due to lack of code-switched data. In cases where the two languages being mixed have different scripts, there may be issues of cross-transcription of borrowed words. Languages that are mainly spoken and not written often may have non-standardized spellings in human transcriptions.

Spoken Term Detection can be solved in a variety of ways depending on the application. In this work, we set up the problem as follows: given a keyword, we want to find if a particular audio sample contains the word or not. Currently, we do not attempt to find the exact location of the word in the audio in terms of timestamps. A straightforward way to approach this problem would be to run an ASR system on the data and look for the keyword within the hypothesis output by the ASR system. In this paper, we describe the problems with this approach and preliminary experiments in the code-switched setting.

Our main contributions are as follows: 1. We discuss challenges in code-switched STD for Hindi-English, due to non-standardized spellings and cross-transcription, 2. We carry out experiments on Hindi-English conversational speech using a monolingual ASR in the matrix language, Hindi, 3. We present preliminary results based on the output of the Hindi ASR, evaluated using a standard metric for STD. The rest of the paper is organized as follows. Section 2 describes the data and experiments. Section 3 discusses the evaluation metric used and results obtained. Section 4 describes related work in STD. Section 5 concludes with ongoing and future work.

II. Data and Experiments

A. Data

For this work, we used a proprietary Hindi-English conversational speech corpus available in-house, described in Table I. This corpus contains conversations between pairs of bilinguals who were given a topic to talk about. The data is transcribed in Devanagari (for Hindi words) and Latin scripts (for English words). The matrix language of these utterances is Hindi, with the exception of very short utterances such as *Okay*. This data is not annotated with word or phone-level timestamps.

Identify applicable funding agency here. If none, delete this.

We ran state-of-the-art in-house monolingual Hindi and English ASR systems, and calculated the Word Error Rate (WER) to determine how well a standard ASR would perform on this data. As expected, the WER of the Hindi ASR was much lower than the English ASR, since *Hindi* is the matrix language of the utterances. However, due to the presence of code-switching, the monolingual Hindi ASR also performs quite poorly on the data.

Languages	Hindi & English
Matrix Language	Hindi
Utterances	52k
Unique words	21.6k
Avg. words per utterance	21.4
% of Hindi unigrams	86%
% of English unigrams	14%
Avg. code-switching points per utterance	3.17
WER Hindi ASR	48%
WER English ASR	80%

Table I
Code-switched Speech Data

Since we wanted to perform STD, we needed a list of terms or keywords. We selected ~1k keywords from this data consisting of both Hindi and English words, distributed almost equally among the two languages. English keywords were between 8 and 11 characters in length while Hindi keywords were at least 6 characters in length. All keywords had a reasonable frequency in the data.

Cross-transcription is a common problem in code-switched speech transcription since borrowed words can be transcribed in either of the two scripts being used [1] and code-switching and borrowing are not always clearly distinguishable [2]. Table II contains examples of keywords, the output of the Hindi ASR and the reference human transcriptions. We see instances of cross-transcription, along with spelling variations, spelling errors, and variations in the transcription of compound words. As seen from the table, the borrowed word *doctor* is specified in both scripts, while other words have spelling variations. The word *hard-working* is transcribed in the Devanagari script in the human annotated transcript, which can be considered to be a transcription error, since it is an example of code-switching and not borrowing. While some of these issues can also be present in monolingual data, they are present to a larger extent in code-switched data.

B. STD using the ASR hypothesis

We ran a monolingual Hindi ASR on the data and performed transformations on the ASR hypothesis.

1) Text normalization: We tried three normalization techniques - translation, transliteration and stemming. Given a keyword, we wanted to find all occurrences of the word, its transliterated variant, and also its translated variant, since it may be present in the other language. For this, we used the Azure Cognitive Services Hindi-English translator system [1] that translates all Hindi words into English but transliterates all borrowed words. If we choose to simply transliterate all words in the ASR output, we may end up with homographs that do have the same meaning as the keyword. For example, the word लोग in Hindi will be translated into *people*, but transliterated into English as *log*, which may trigger a false alarm. In addition to translation and transliteration, we also perform stemming of the keyword and the words in the ASR output and match the root words.

2) Pronunciation-based matching: Next, we tried matching terms in the ASR hypothesis in the pronunciation space. We divided phonemes into classes based on phonetic features and treated each phoneme in a single class as equal. We ran a Grapheme to Phoneme (g2p) system on the keywords and the words in the ASR hypothesis and mapped each phoneme to its class. If the phoneme class of each phoneme in the word matched with that of the keyword, we considered the keyword found. This helped take care of minor spelling errors such as long/short vowels and nasalization.

Since word-level matching does not solve the compound word problem and more crucially, instances where the ASR's performance is poor and it fails to recognize even a variant of the word, we performed utterance level matching. For this, we found the substring within the utterance phone sequence which gave the best alignment with the keyword phoneme sequence using the Needleman–Wunsch Global Alignment Algorithm [3] as shown in Figure1, which was originally used for aligning gene sequences. It takes two strings as parameters and outputs the alignment score between the two strings. We assigned +2 for match, -1 for mismatch, -0.5 for gap and -0.1 for extended gap. These scores were determined empirically by us.

We took chunks of substrings from the utterance phoneme sequence of length equal to the length of the keyword phoneme sequence. The maximum alignment score is equal to 2*(length of the keyword phoneme sequence) in case of exact match. We took this score as the reference score (RS). The least score is for a complete mismatch, which is equal to –1*(length of the keyword phoneme sequence). We took the chunk which had the highest alignment score (HAS) with the keyword phoneme sequence. We took the difference between HAS and RS to calculate the percentage difference (PD) or threshold (θ). We set the threshold (θ) before hand and if PD was below the threshold then we determined that the keyword was present within the utterance.

III. Evaluation and Results

A. Evaluation metric

We used a standard metric for Spoken Term Detection, Term Weighted Value (TWV) proposed for the NIST Spoken Term Detection evaluation [4], which is measured in terms of misses and false alarms. TWV of 1 indicates no misses and no false

[1] https://docs.microsoft.com/en-us/azure/cognitive-services/translator/translator-info-overview

Problems	Keywords	Transcript Examples (Hindi Monolingual ASR)	Transcript Examples (Bilingual Human Annotators)
Cross-Transcription / Borrowing	Doctor / डॉक्टर	मैं **डॉक्टर** के पास जा रहा हूँ।	मैं **doctor** के पास जा रहा हूँ।
Part Of Speech Variations	फ्रेंड्स /Friend	क्या हम **फ्रेंड** बन सकतें हैं ?	क्या हम **फ्रेंड** बन सकतें हैं ?
Spelling Mistakes / Variations	निचे	मेंनीचे जाके आइस-क्रीम लेके आता हूँ।	में **निचे** जाके ice-cream लेके आता हूँ।
Compound Words	hard-working	वह काफी **हार्डवर्किग** लड़का हैं।	वह काफी **हार्ड वर्किग** लड़का हैं।

Table II
Transcription issues for STD using Monolingual ASR

alarms, while a system that outputs nothing gets a TWV of 0. Negative TWVs are possible. The Detection Error Tradeoff (DET) curve plots miss probability (PMiss) versus false alarm probability (PFA). The θ parameter is the detection threshold, which is averaged across all terms to compute the DET curve.

$$P_{Miss}(term,\theta) = 1 - \left(\frac{N_C(term,\theta)}{N_T(term)}\right) \quad (1)$$

$$P_{FA}(term,\theta) = \left(\frac{N_{spurious}(term,\theta)}{N_{NT}(term)}\right) \quad (2)$$

$$AE(term,\theta) = \frac{P_{Miss}(term,\theta) + \beta * P_{FA}(term,\theta)}{N_{TT}} \quad (3)$$

$$TWV(term,\theta) = 1 - AE(term,\theta) \quad (4)$$

$N_C(term,\theta)$ is number of correct (true) detections of the term with a detection score greater than or equal to θ, $N_{spurious}(term,\theta)$ is number of spurious (incorrect) detections of term with a detection score greater than or equal to θ, $N_{true}(term)$ is true number of occurrences of term in the corpus, $N_{NT}(term)$ is number of opportunities for incorrect detection of term in the corpus (= *Non-Target* term trials).

B. Results

Table III shows the results of the text normalization and pronunciation experiments in terms of TWV. We use the exact match (no transformations to the keyword or ASR hypothesis) as a baseline, and unsurprisingly, it performs poorly with a TWV of 0.06. Both stemming and translation/transliteration lead to an improvement in TWV. The word-level phonetic match performs better in terms of TWV because it can handle spelling variations and cross-transcriptions. The utterance level phonetic match performs the best when the θ parameter is set to 0.45. However note that we do not check where in the utterance the term is spotted. So, it is possible that the utterance level match hallucinates a match in the wrong location in the audio and we still mark it as a match.

Figure 2 shows how Miss Probability (P_{Miss}), False Alarm Probability (P_{FA}) and the TWV score varies as we gradually increase the θ parameter for the utterance level match. We see that at θ equals to ~0.45, P_{Miss} curve and the P_{FA} curve intersect giving a global maxima of ~0.64 for the TWV curve which we mention in Table III.

	Experiment	θ	TWV
Text Normalization	Baseline (exact match)	–	**0.06**
	Stemming	–	**0.11**
	Translation/Transliteration	–	**0.11**
Pronunciation	Phonetic Match (Word)	–	**0.32**
	Phonetic Match (Utterance)	0.45	**0.64**

Table III
Term Weighted Value for all experiments

IV. Relation to prior work

Our work is related to prior work in Spoken Term Detection for low resource languages [5]–[12], in which considerable research was carried out as part of the IARPA BABEL project [13]. Our work also contains similarities with Out of Vocabulary (OOV) STD [14], in which subword units and phonetic search are used, however, the OOV and in-vocabulary words belong to the same language.

There has also been some work on STD for code-switching. [15] describe work on Chinese-English mixed-lingual KWS, in which they build a Taiwanese accented English ASR and decode speech using both Chinese and English ASRs. They combine scores from both systems to determine the most probable output. In our work, we chose not to use both ASR systems due to the poor performance of the English ASR.

[16] use syllables as subword units for Mandarin and word fragments as subword units for English for Mandarin-English STD. They construct lattices using these subword units and observe that the performance on Mandarin queries does not improve due to the difference in duration of these units in the two languages. They also notice that the performance on English queries does not improve upon using subword units.

[17] model the pronunciation of English words in a Swahili KWS system and find that code-switching degrades the performance of ASR and KWS. Swahili words are modeled in the system using g2p rules, however, the presence of English words decreases the KWS accuracy. The authors identify English words automatically and map their pronunciation to Swahili phonemes.

V. Conclusion

We describe preliminary work on STD for Hindi-English code-switched speech using a monolingual Hindi speech rec-

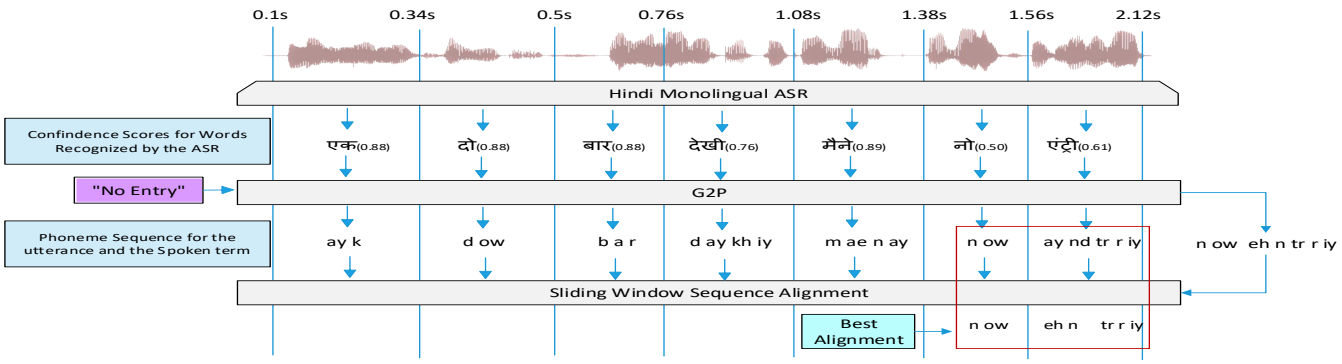

Figure 1. Workflow showing how Needleman-wunsch Global Alignment Algorithm is used for phone sequence alignment.

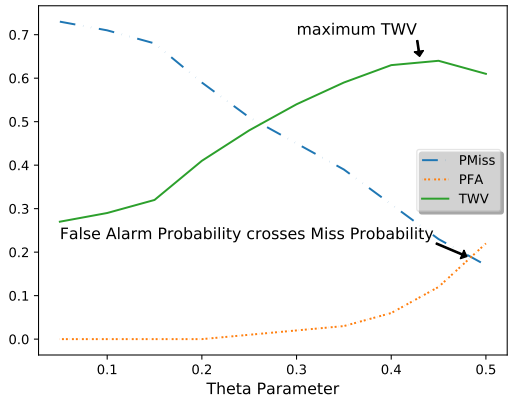

Figure 2. Detection Error Tradeoff Curve showing variation of Miss Probability (P_{Miss}), False Alarm Probability (P_{FA}) and TWV with change in the θ parameter for utterance level phonetic match.

ognizer and find that utterance-level phonetic matching between the keyword and the transcript performs best. In future work, we would like to explore better ways to do the utterance level match, particularly when the ASR performance is poor. Instead of treating all phonemes in a class as same, we would like to use a weighted distance between phonemes.

We do not verify whether the keyword is found in the correct location in the audio, since we do not have reference timestamps for words in our data. We plan to use ASR forced alignment to obtain approximate timestamps for the words in the audio. Additionally, we only use the top hypothesis from the ASR system for searching for a match. In future work, we plan to use the n-best list to create a word and phone lattice, and incorporate ASR confidence values in the scoring metric.

We applied our techniques on clean, well recorded speech. We plan to test our techniques on other data such as Youtube videos, where the ASR performance may be worse. Finally, we would like to compare the performance of our techniques to the performance of a code-switched ASR system.

References

[1] B. M. L. Srivastava and S. Sitaram, "Homophone identification and merging for code-switched speech recognition," *Proc. Interspeech 2018*, pp. 1943–1947, 2018.

[2] K. Bali, J. Sharma, M. Choudhury, and Y. Vyas, "" 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*, 2014, pp. 116–126.

[3] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[4] J. Fiscus, J. Ajot, and G. Doddington, "The spoken term detection (std) 2006 evaluation plan," *NIST USA, Sep*, 2006.

[5] M. J. Gales, K. M. Knill, A. Ragni, and S. P. Rath, "Speech recognition and keyword spotting for low-resource languages: Babel project research at cued," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[6] H. Wang, A. Ragni, M. J. Gales, K. M. Knill, P. C. Woodland, and C. Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[7] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. Schwartz, and S. Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[8] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1. IEEE, 2006, pp. I–I.

[9] S. P. Rath, K. M. Knill, A. Ragni, and M. J. Gales, "Combining tandem and hybrid systems for improved speech recognition and keyword spotting on low resource languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[10] N. F. Chen, C. Ni, I.-F. Chen, S. Sivadas, H. Xu, X. Xiao, T. S. Lau, S. J. Leow, B. P. Lim, C.-C. Leung *et al.*, "Low-resource keyword search strategies for tamil," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5366–5370.

[11] W. Hartmann, V.-B. Le, A. Messaoudi, L. Lamel, and J.-L. Gauvain, "Comparing decoding strategies for subword-based keyword spotting in low-resourced languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[12] R. Hsiao, T. Ng, F. Grézl, D. Karakos, S. Tsakalidis, L. Nguyen, and R. Schwartz, "Discriminative semi-supervised training for keyword search in low resource languages," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 440–445.

[13] M. Harper, "Iarpa solicitation iarpa-baa-11-02," *IARPA BAA*, 2011.

[14] D. Karakos and R. Schwartz, "Subword and phonetic search for detecting out-of-vocabulary keywords," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[15] S.-R. You, S.-C. Chien, C.-H. Hsu, K.-S. Chen, J.-J. Tu, J. S. Lin, and S.-C. Chang, "Chinese-english mixed-lingual keyword spotting," in *2004 International Symposium on Chinese Spoken Language Processing*. IEEE, 2004, pp. 237–240.

[16] H.-Y. Lee, Y.-L. Tang, H. Tang, and L.-S. Lee, "Spoken term detection from bilingual spontaneous speech using code-switched lattice-based structures for words and subword units," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 410–415.

[17] N. Kleynhans, W. Hartman, D. Van Niekerk, C. Van Heerden, R. Schwartz, S. Tsakalidis, and M. Davel, "Code-switched english pronunciation modeling for swahili spoken term detection," *Procedia Computer Science*, vol. 81, pp. 128–135, 2016.