



Linguistically Informed Post-processing for ASR Error correction in Sanskrit

Rishabh Kumar¹, Devaraja Adiga¹, Rishav Ranjan¹, Amrith Krishna², Ganesh
Ramakrishnan¹, Pawan Goyal³, Preethi Jyothi¹

¹IIT Bombay, Mumbai, India; ²Uniphore; ³IIT Kharagpur, WB, India

krrishabh@cse.iitb.ac.in, pdadiga@iitb.ac.in, 180070045@iitb.ac.in, amrith.krishna@uniphore.com,
ganesh@cse.iitb.ac.in, pawang@cse.iitkgp.ac.in, pjyothi@cse.iitb.ac.in

Abstract

We propose an ASR system for Sanskrit, a low-resource language, that effectively combines subword tokenisation strategies and search space enrichment with linguistic information. More specifically, to address the challenges due to the high degree of out-of-vocabulary entries present in the language, we first use a subword-based language model and acoustic model to generate a search space. The search space, so obtained, is converted into a word-based search space and is further enriched with morphological and lexical information based on a shallow parser. Finally, the transitions in the search space are rescored using a supervised morphological parser proposed for Sanskrit. Our proposed approach currently reports the state-of-the-art results in Sanskrit ASR, with a 7.18 absolute point reduction in WER than the previous state-of-the-art.

Index Terms: speech recognition, human-computer interaction, computational linguistics, Sanskrit ASR

1. Introduction

Sanskrit is one of the oldest language of humanity with an unbroken oral tradition spanning more than two millennia [1]. With more than 25,000 native speakers, the language is still used as a medium for scholarly and philosophical discourses in India. Efforts toward building automatic speech recognition (ASR) systems for Sanskrit have gained some traction in the recent years [2, 3]. However, ASR in Sanskrit is challenging as it is a low-resource language that is both morphologically rich and lexically productive [4, 5].

Consider a four-word sentence पीताम्बरात् फलम् मुनिनैष्यत्¹ (/pītāmbarāt phalam muninaiṣyata/), which means ‘the fruit was desired from Pītāmbara(a god) by the sage’. Here, the word पीताम्बर (/pītāmbara/) is a compound word, while मुनिनैष्यत् (/muninaiṣyata/) is a phrase with two words मुनिना (/muninā/) and ऐष्यत् (/aiṣyata/). The former is a single compound word formed from two different stems *pīta* (yellow) and *ambara* (cloth). However, the compound word, referring to a god, is semantically different from either of its components, making it an exocentric compound [6]. Compounds are highly prevalent in Sanskrit corpora, constituting as high as 75 % of the vocabulary, in contrast to a mere 3-4 % vocabulary share for corpora in English. Similarly,

¹पीताम्बरात् – from Pītāmbara (A god’s name), फलम् – fruit, मुनिना – By the sage, ऐष्यत् – desired

मुनिनैष्यत् (/muninaiṣyata/) is a phrase with two words fused due to phonetic transformations at the word boundaries typically seen in connected speech. Such transformations are faithfully preserved in writing as well in Sanskrit and are collectively referred to as *Sandhi*. Though such transformations do not modify the words involved syntactically or semantically, they lead to ambiguity in the analysis. Moreover, ऐष्यत् (/aiṣyata/) (desire) is an inflected word form applicable to six different stems, each varying semantically due to the phenomenon of homonymy. Sanskrit, being morphologically rich, has a rich tag set with 1,635 possible tags. Finally, sentences in Sanskrit tend to follow relatively free-word ordering.

In this work, we employ a subword vocabulary-based language model, followed by a post-processing module, which enriches the search space using linguistic information at the morpho-syntactic level. Typical word-based n-gram language models (LM) can lead to high rates of out-of-vocabulary (OOV) words and a long tail of rare words in a Sanskrit corpus, arising out of rich compounding and inflection in Sanskrit. Subword LMs have shown to effectively address these challenges, as it can decompose any given word in a corpus in terms of the vocabulary it has learned. However, such approaches have shown to be of limited utility in handling various linguistic phenomena in natural languages, owing to their purely statistical nature in learning a vocabulary. For a language like Sanskrit, these phenomena include morpho-syntactic regularities, Sandhi between words and morphemes, ambiguities arising out of homonymy, syncretism and also from free word-order constructions [7].

In our proposed approach, we first train a Time Delay Neural Network (TDNN) based acoustic model (AM), which uses the Byte-Pair Encoding (BPE) based subword vocabulary for its LM. Using the model, we generate a subword-level confusion network-based search space. We then convert the subword level search space into a word confusion network (WCN), pruned using a lexicon-driven shallow parser [8]. Moreover, the shallow parser enriches the word-level search space in four ways. 1. Since it is lexicon driven, it helps identify invalid character combinations that can not form a valid word. 2. It enumerates all possible morphological analyses for each word and provides the possible alternative tags and stems for the word. The morphological analyses help in identifying cases of homonymy and syncretism. 3. It provides segmentation of compounds into individual component stems, thereby handling lexical productivity and identifying alternate compound split hypotheses. 4. It provides

segmentation of words joined by Sandhi, thereby identifying alternate possible word split hypotheses. Finally, we integrate the scores by leveraging the energy-based framework (EBM) proposed for morpho-syntactic tasks in Sanskrit [7, 9] for rescoring the edges in the WCN. The EBM framework has yielded state-of-the-art results in word segmentation and morphological parsing by effectively disambiguating alternate hypotheses involving cases of syncretism, homonymy, Sandhi, and compound splits in free word order sentence construction.

We first observe that rescoring the top 10 sentences from our base TDNN model using sequence level energy scores from our energy-based model leads to statistically significant error reduction (from 23.49 to 22.41). However, our use of word-level confusion networks enriched with morphological level information leads to an absolute 7.18 point error reduction from 23.49 to 16.31 WER.

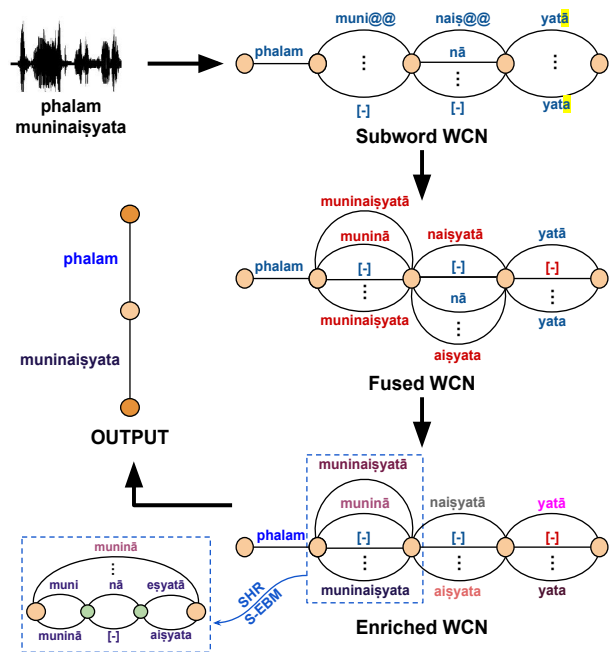


Figure 1: Different phases of search space of the proposed model for a sample audio.

2. Related Work

Morphological information has been extensively used in ASR for performance enhancement [10, 11, 12]. The morphological information can help decompose the tokens into morphemic or sub-morphemic units such as demisyllables, morphemes, stems, etc. [13, 14], thereby bringing down the vocabulary size and reducing the out-of-vocabulary (OOV) frequency. Similarly, purely statistical approaches, such as word-piece [15] and BPE [16], which learn their vocabularies by identifying recurrent patterns of subwords, have also been employed in ASR [17, 18]. These approaches have either been incorporated directly into the language model component of the ASR or as a post-processing module.

For post-processing and downstream tasks, several of these approaches make use of the language model's search space in the form of ASR 1-best [19], top K-list [20], lattice [21], or word confusion networks [22, 15]. The word

confusion networks [23] is a time-aligned and normalised compact lattice representation. WCNs have been generally shown to have lower latency and outperform 1-best or top-k list decoding approaches over lattices [24].

3. Methodology

Our proposed model takes in the audio input and processes it using a Time Delay Neural Network (TDNN) based acoustic model, followed by an n-gram based language model with a learned sub-word vocabulary. Sanskrit Library Phonetic (SLP1) representation of Sanskrit [25] follows phonemic orthography, implying that a one-to-one correspondence between graphemes and phonemes exists. Consequently, we do not need an explicit pronunciation mapping mechanism. Instead, we follow the same vocabulary for the AM and the LM.

3.1. Search Space Construction:

Using the LM, we generate word confusion networks (WCN) as our search space. WCNs group together multiple alternate hypotheses aligned for a given acoustic unit (or a time step), such as *yata* and *yatā* in last bin of the *Subword WCN* in Figure 1. Each alternate hypothesis, annotated with blue colour, is an entry from the subword vocabulary and is represented as a transition in the bin. From the subword WCN, we formulate a new WCN where we obtain lexical level entries as the transitions. These entries, annotated with red in Figure 1, are formed by deterministically combining the subword entries in the subword WCN. We refer to the new WCN as *fused WCN* and its transition labels as fused entries. Here, a fused entry can either be a simple word, which has a stem in its inflected form, or a compound word, which has multiple stems with inflection applied to the last stem or can be a phrase with multiple inflected words fused together due to Sandhi. Each fused entry is then passed onto a shallow parser to obtain additional linguistic information.

3.2. Linguistically guided search space enrichment

We use a lexicon-driven shallow parser [26] that performs segmentation and morphological analysis for a given fused entry. The parser enumerates all possible word split hypotheses and compound segmentation hypotheses for a given sequence. Further, it also provides the morphological analysis for each inflected form in a hypothesis, leading to multiple possible analyses per inflection due to homonymy and syncretism. We construct the *Enriched WCN* by incorporating the information from the shallow parser. Finally, as the shallow parser is lexicon-driven, it also acts as a filter to distinguish valid word forms from invalid word forms, which can be formed due to variations in the acoustic input.

Ideally, in our fused WCN, *muninaiṣyata* will form a transition label from one bin in the WCN to another, grouped with other alternate similar-sounding fused entries. WCNs are compositional in nature, implying that the structure can further be subdivided into a substructure of the same nature. We use the compositionality of the structure to enrich it with additional information. For instance, a fused entry such as *muninaiṣyata* would

lead to two alternate hypotheses at the surface level itself, owing to a word split of *muninā* and *aiṣyata*, and the alternative split as a compound - *muni* + *nā* and *aiṣyata*. Similarly, another fused entry of the same bin as *muninaiṣyatā* can be split of *muninā* and *eṣyatā*, and *muni* + *nā* and *eṣyatā*. Further, *aiṣyata* has 12 possible alternative analyses, of which five are cases of homonymy, and the remaining are cases of syncretism. With enriched information for *muninaiṣyata* alone from the shallow parser, the transition label has 24 different possible combinations, and hence we form 24 different transitions, each being an enriched version of the original fused entry. Each analysis is a collection of a unique combination of one or more words, with at least one entry in the combination differing in terms of inflected form, morphological tag or stem from any other alternate hypotheses. We present one such case in Figure 1. Moreover, we ensure that the hypothesis we form exhaustively covers a fused entry. For instance, *muninā* alone would not suffice as a hypothesis for *muninaiṣyata*, as it does not have any hypothesis for *aiṣyata*.

3.3. Scoring the Transitions in the WCN

The subword WCN is obtained from the lattice, where the latter is a compact representation of the former. The posterior probability of a transition in the subword WCN is the sum of normalised probabilities of all the lattice paths, which contains the transition as a part of it. The probabilities are obtained based on the weighted combination of acoustic model (AM) and language model (LM) probabilities. The posterior probabilities of the transitions in the fused WCN are obtained using the product of the posterior probabilities of the subword WCN transitions, which are part of a transition label in the fused WCN. Further, after the search space enrichment, each fused entry results in multiple possible exhaustive entries. This enriched search space can now be annotated with morpho-syntactic information to help the model make an informed decision. Here, we rely on the Energy-based model framework (EBM), proposed for multiple sentence-level structured prediction tasks in Sanskrit [9, 7]. We specifically make use of their joint word segmentation and morphological parsing model.

The model takes a Sanskrit sentence as input and predicts the segmented words in it, along with the morphological tag for each of these words. For this, the model converts the input sentence into a graph where the nodes contain the linguistic information provided by the shallow parser [8]. Further, edges are formed between every pair of nodes as long as they are not suggested as alternative hypotheses. Being an arc-factored model, the graph is factored into its edges, where each edge is featured, and a function is learned to score these edges. Now, the inference procedure searches for a maximal clique as the solution, where each clique is scored as the sum of the edges in the clique. Ideally, the clique with the minimum score (or energy) is the best prediction. In our case, we are interested in the intermediate structure that the model produces, or more specifically, the edges and their scores. To obtain the score for a node, we simply aggregate the score of each word by summing up all the edge scores in which a given node forms the source node.

Since we use the same lexicon parser as that of the

EBM model, each transition label in our enriched WCN is a combination of one or more nodes in the EBM model’s graph representation. Hence, each transition is scored with the sum of the scores of each node in the EBM, implying each transition in the enriched WCN is simply a substructure in the EBM graph. Finally, the EBM scores are also normalised for a given bin in the fused-word WCN. While each of these transitions in the enriched WCN already has a normalised score assigned based on the ASR model, we now obtain a weighted average of these scores and that of the EBM. Final selection proceeds with the greedy decoding process finding the transition with the maximum weighted average of the probability scores obtained from both subword-LM based ASR and EBM.

4. Experiments

4.1. Dataset

We use वाक्सञ्जयः² (/Vākṣaṅcayah/), a Sanskrit speech corpus with a train-test split of 56 and 11 hours, resulting in 34,309 sentences (12 speakers) and 6,004 sentences (6 speakers), respectively. While maintaining the original train data split, we further divide the test split into test and development sets consisting of 4,424 and 1,580 sentences, respectively. All our experiments are performed on the new train-dev-split containing a disjoint set of speakers (12, 3 and 3 speakers, respectively).

4.2. Experimental Setup

We use the best-performing configuration of Adiga *et al.* [3], which is the current state of the art in Sanskrit ASR, as our baseline. We will henceforth refer to this configuration as *VakASR*. *VakASR* uses an automatically learned vocabulary using BPE as their AM and LM units. These units are represented using SLP1, a phonetic encoding scheme. For the acoustic model, we employ TDNNs [27] with 14 layers, with input features consisting of 40-dimensional MFCCs and 100-dimensional i-vector based speaker embeddings [28]. For the LM, a 3-gram BPE language model with Kneser-Ney smoothing is employed using the SRILM toolkit [29].

Our work is pivoted on two central hypotheses. First, the subword tokenisation strategy for LM in ASR is more beneficial than word-based LMs for Sanskrit. Second, the integration of lexical and morphological information to enrich the search space can lead to significant improvements in ASR performance. Given that *VakASR* currently uses the subword-based vocabulary for its LM, we first compare it with a configuration that uses word-based LM, specifically a 3-gram LM, and phonemes as the AM units. We will refer to this system as *VakASR-Word*.

For the second hypothesis, we experiment with three different configurations. All three configurations use *VakASR-BPE* to generate subword-based search space and then use Sanskrit Heritage Reader (SHR) as the shallow parser to enrich the search space with lexical and morphological information. Further, all the three configurations first prune the search space by considering only the tokens in the 10-best candidates from *VakASR-BPE*. The first configuration, *viz.*, *MorphASR-WCN-morphLM* initially forms fused entry WCN and

²<https://www.cse.iitb.ac.in/~asr/>

then rescores individual transitions in the WCN based on a language model, which is a weighted average of co-occurrence probabilities of the inflected form, stem and morphological tag of a given analysis. The second configuration, *MorphASR-NBest-EBM* obtains n-best candidate sequences for given speech input and then rescores them by obtaining sentence level EBM scores for each candidate. Finally, *MorphASR-WCN-EBM* obtains the WCN-based search space, followed by the EBM scoring of individual transition edges.

Evaluation: We evaluate all our ASR based experiments using word error rate (WER).

4.3. Result

Method	DEV	TEST	TEST *
<i>VakASR-Word</i>	35.68	42.52	32.22
<i>VakASR-BPE</i>	18.62	23.49	18.79
<i>MorphASR-NBest-EBM</i>	17.80	22.41	18.5
<i>MorphASR-WCN-morphLM</i>	16.18	20.26	17.91
<i>MorphASR-WCN-EBM</i>	14.15	16.31	13.7

Table 1: *WER for ASR systems using different methods (*modulo substitution deletion WER for TEST results)*

Recall our two central hypotheses (*c.f.*, Section 4.2), *viz.*, subword tokenisation strategy being more beneficial and that search space enrichment can lead to significant improvements in ASR performance. In Table 1, we present the WER for all the ASR systems we compare. For the first hypothesis, we find that subword-based LM outperforms word-based LM by a huge margin, as can be observed from the reduction in errors between *VakASR-Word* and *VakASR-BPE*. All the three search space enrichment approaches, with the prefix *MorphASR-*, outperform *VakASR-BPE*, the current state-of-the-art, validating our second hypothesis. Of the three, our proposed search space enrichment approach, namely *MorphASR-WCN-EBM*, performs the best with a WER of 16.31% against that of 20.26% of *MorphASR-WCN-morphLM*. Both of them use the same search space as a fused WCN, though they differ in how rescoring is used. *MorphASR-WCN-EBM* uses EBM for scoring the transitions in the WCN. Although *MorphASR-NBest-EBM* also uses EBM for rescoring, it is done at a sentence level, thus limiting its ability to combine partial solutions from multiple candidates. We find that 17.97% of its predicted tokens are incorrect, of which 47.17% are correctly predicted by *MorphASR-WCN-EBM* as it can combine partial solutions from multiple candidate sentences using WCN. Summarily, our proposed approach, *MorphASR-WCN-EBM*, achieves a reduction in the WER by 4.47 and 7.18 absolute points on the dev and test sets, respectively, over the current state-of-the-art.

Adiga *et. al.* [3] previously observed that speakers tend to pause arbitrarily when pronouncing long compound words, leading to spaces at arbitrary points in the generated text. This directly affects ASR performance by adding or deleting the required space between two correctly recognised subwords or words. After accounting for this, the *modulo substitution deletion* results are also shown for the Test set in the last column of Table 1.

The shallow parser we employ for search space enrichment is lexicon driven and hence is prone to Out of Vo-

Method	With OOV	Without OOV
VakASR-BPE	19.59	-
MorphASR-NBest-EBM	18.12	17.16
MorphASR-WCN-EBM	14.06	12.78

Table 2: *WERs for different Strategies in 200MAU(Manually annotated utterances) dataset. EBM* = EBM with Length Penalty*

cabulary (OOV) entries. However, using subword-based LM eliminates the issue during search space generation. Since our rescoring mechanisms generate features based on linguistic information from the shallow parser, OOV entries are provided with default values for those features, and Table 1 shows the results after using default values for OOV entries. However, given the wide coverage of our shallow parser, we have only 0.16% tokens as OOV, forming 1.12% of the vocabulary.

We sample 200 sentences from our test data with OOV entries and annotate them manually to generate features for those entries. This sample contains 8.17% OOV tokens. For the sample, the manual annotations result in further error reduction, as shown in Table 2. As manual annotation, all we need to provide to our shallow parser is the stem information, based on which our shallow parser can handle the cases of inflection, compounding and Sandhi for those stems [26]. Overall there is a 1.28 absolute points reduction in WER for *MorphASR-WCN-EBM* by resolving the OOV entries.

Strategy	VakASR-BPE	MorphASR-NBest-EBM	MorphASR-WCN-EBM
Compound Analysis	43.33	44.01	55.67
Syncretism	46.25	47.28	59.47
Homonymy	51.54	52.57	64.79

Table 3: *Results for linguistic analysis for the three different ASR configurations. F-score is used as the metric*

The improvement in the results for the model *MorphASR-WCN-EBM* is in agreement with the various morphological analysis as shown in Table 3. We manually annotated 500 sentences from the test data with morphological and compound information³. *MorphASR-WCN-EBM* reports significant improvements in recognising compound words and resolving ambiguity due to homonymy and syncretism.

5. Conclusion

In this work, we show that both subword tokenisation strategies and search space enrichment with morphological and lexical information significantly reduce ASR errors for Sanskrit. Our improvements, as compared to the previous state of the art, come primarily from using a wide-coverage lexicon-driven shallow parser and energy-based model for joint word segmentation and morphological parsing. In future work, we intend to extend our approach to other morphologically rich Indic languages.

³Data available at <https://www.cse.iitb.ac.in/~asr/ebm>

6. References

- [1] A. Kulkarni and M. Das, "Discourse analysis of sanskrit texts," in *Proceedings of the workshop on advances in discourse analysis and its computational aspects*, 2012, pp. 1–16.
- [2] C. Anoop and A. Ramakrishnan, "Automatic speech recognition for sanskrit," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1. IEEE, 2019, pp. 1146–1151. [Online]. Available: <https://ieeexplore.ieee.org/iel7/8967528/8993111/08993283.pdf>
- [3] D. Adiga, R. Kumar, A. Krishna, P. Jyothi, G. Ramakrishnan, and P. Goyal, "Automatic speech recognition in sanskrit: A new speech corpus and modelling insights," *arXiv preprint arXiv:2106.05852*, 2021.
- [4] C. Anoop and A. Ramakrishnan, "Ctc-based end-to-end asr for the low resource sanskrit language with spectrogram augmentation," in *2021 National Conference on Communications (NCC)*. IEEE, 2021, pp. 1–6.
- [5] A. Ramakrishnan, "Investigation of different g2p schemes for speech recognition in sanskrit."
- [6] M. Nußbaum-Thom, A. E.-D. Mousa, R. Schlüter, and H. Ney, "Compound word recombination for german lvcsr," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [7] A. Krishna, B. Santra, A. Gupta, P. Satuluri, and P. Goyal, "A graph-based framework for structured prediction tasks in sanskrit," *Computational Linguistics*, vol. 46, no. 4, pp. 785–845, 2021.
- [8] G. Huet, "A functional toolkit for morphological and phonological processing, application to a sanskrit tagger," *Journal of Functional Programming*, vol. 15, no. 4, pp. 573–614, 2005.
- [9] A. Krishna, B. Santra, S. P. Bandaru, G. Sahu, V. D. Sharma, P. Satuluri, and P. Goyal, "Free as in free word order: An energy based model for word segmentation and morphological tagging in sanskrit," *arXiv preprint arXiv:1809.01446*, 2018.
- [10] M. Elbeze and A.-M. Derouault, "A morphological model for large vocabulary speech recognition," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 577–580 vol.1, 1990.
- [11] C.-H. Lee, J.-L. Gauvain, R. Pieraccini, and L. R. Rabiner, "Large vocabulary speech recognition using subword units," *Speech communication*, vol. 13, no. 3-4, pp. 263–279, 1993.
- [12] F. Diehl, M. J. Gales, M. Tomalin, and P. C. Woodland, "Morphological decomposition in arabic asr systems," *Computer Speech & Language*, vol. 26, no. 4, pp. 229–243, 2012.
- [13] P. Geutner, "Using morphology towards better large-vocabulary speech recognition systems," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 1995, pp. 445–448.
- [14] L. Lamel, A. Messaoudi, and J.-L. Gauvain, "Investigating morphological decomposition for transcription of Arabic broadcast news and broadcast conversation data," in *Proc. Interspeech 2008*, 2008, pp. 1429–1432.
- [15] C. Liu, S. Zhu, Z. Zhao, R. Cao, L. Chen, and K. Yu, "Jointly encoding word confusion network and dialogue context with bert for spoken language understanding," *arXiv preprint arXiv:2005.11640*, 2020.
- [16] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: <https://aclanthology.org/P16-1162>
- [17] C. Wang, K. Cho, and J. Gu, "Neural machine translation with byte-level subwords," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9154–9160.
- [18] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5621–5625.
- [19] M. Henderson, M. Gašić, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative spoken language understanding using word confusion networks," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 176–181.
- [20] J.-P. Robichaud, P. A. Crook, P. Xu, O. Z. Khan, and R. Sarikaya, "Hypotheses ranking for robust domain classification and tracking in dialogue systems," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [21] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "Latticernn: Recurrent neural networks over lattices," in *Interspeech*, 2016, pp. 695–699.
- [22] G. Tür, A. Deoras, and D. Hakkani-Tür, "Semantic parsing using word confusion networks with conditional random fields," in *INTERSPEECH*. Citeseer, 2013, pp. 2579–2583.
- [23] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.
- [24] D. Z. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tür, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Comput. Speech Lang.*, vol. 20, pp. 495–514, 2006.
- [25] P. M. Scharf and M. D. Hyman, *Linguistic Issues in Encoding Sanskrit*. The Sanskrit Library, 2011. [Online]. Available: https://sanskritlibrary.org/Sanskrit/pub/lies_sl.pdf
- [26] P. Goyal and G. Huet, "Design and analysis of a lean interface for sanskrit corpus annotation," *Journal of Language Modelling*, vol. 4, no. 2, pp. 145–182, 2016.
- [27] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH*, 2015. [Online]. Available: https://188.166.204.102/archive/interspeech_2015/papers/i15_3214.pdf
- [28] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 55–59, 2013. [Online]. Available: <https://ieeexplore.ieee.org/iel7/6695806/6707689/06707705.pdf>
- [29] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *INTERSPEECH*, 2002. [Online]. Available: https://www.isca-speech.org/archive/archive_papers/icslp_2002/i02_0901.pdf