Repairing ASR output by Artificial Development and Ontology based Learning

C. Anantaram, Amit Sangroya, Mrinal Rawat, Aishwarya Chhabra
TCS Innovation Labs, Tata Consultancy Services Limited, Gurgaon, India
(c.anantaram, amit.sangroya, rawat.mrinal, aishwarya.chhabra)@tcs.com

Abstract

General purpose automatic speech recognition (gpASR) systems such as Google, Watson, etc. sometimes output inaccurate sentences when used in a domain specific scenario as it may not have had enough training samples for that particular domain and context. Further, the accent of the speaker and the environmental conditions in which the speaker speaks a sentence may influence the speech engine to recognize certain words inaccurately. In the context of a domain and the environment in which a speaker speaks the sentences, gpASR output needs a lot of improvement in order to provide effective speech interfaces to domain-specific systems. In this paper, we demonstrate a method that combines bio-inspired artificial development (ArtDev) with machine learning (ML) approaches to repair the output of a gpASR[1]. Our method factors in the environment to tailor the repair process.

1 Introduction

General purpose Automatic Speech Recognition (gpASR) engines such as Google, Watson, Siri have significantly improved in their performance because of the increasing use of deep neural network architectures to learn over massive general corpus. The availability of these gpASR systems have influenced several enterprises to make use of the engines for domain specific applications. However there are several challenges, in terms of the reduced performance when a gpASR is used for domain specific application [Fusayasu *et al.*, 2015a; Morbini *et al.*, 2013]. One of the main challenges is due to the absence of domain specific words, sentences and context, that might not be part of the general vocabulary or may be very insignificant when used to train the MLmodels associated with a gpASR. As a result the performance of any gpASR degrades when it is used for a very specific domain or in untrained environmental conditions. It should also be noted that in case of low resource domains it is even difficult to build a deep learning based ASR because of the sparsity of the speech corpus required to build a deep learning machine models. While various attempts have been made to repair ASR output [Fusayasu *et al.*, 2015a; Peng *et al.*, 2013; Tür *et al.*, 2013; Gurunath Shivakumar *et al.*, 2018; Suhm *et al.*, 2001; Fusayasu *et al.*, 2015b; Kim *et al.*, 2016] the problem of repair of gpASR output for domain-specific terms is still not very well addressed. In this paper, motivated by our earlier work on repairing gpASR output to suit a specific domain, we propose a repair mechanism that is able to gainfully repair the output of a gpASR in a way that its accuracy can be improved significantly for domain specific applications.

2 ArtDev based gpASR Output Repair

Our repair process is motivated by Evolutionary Development (Evo-Devo) processes in biology [Anantaram *et al.*, 2016] to help adapt/repair the overall content accuracy of the gpASR output which has been acoustically repaired. A sub-field of AI called Artificial Development (Art-Dev) applies Evo-Devo principles to find elegant solutions to adaptation and repair problems in AI [Harding and Banzhaf, 2008; Tufte, 2009]. In our approach, we consider an inaccurate gpASR output, T_1', as an 'injured biological cell'. We repair that 'injured cell' through the development of the partial gene present in the input sentence with respect to the genes present in the domain. We assume that we have been provided with the domain ontology describing the terms and relationships of the domain. In our framework, we consider the domain ontology as the true 'genetic knowledge' of that 'biological organism'. In such a scenario, 'genetic repair' becomes a sequence of match-and-replace of words in the sentence with appropriate domain ontology terms and relationships. Once this is done, the 'genotype-to-phenotype repair' is initiated to repair the linguistic errors in the sentence after the 'genetic repair'.

2.1 Step 1: Matching Function

We start by finding matches between domain ontology terms and words that appear in the T_1' sentence through a sliding window of n-gram words. Some words of the T_1' sentence will match domain ontology terms exactly. The corresponding domain ontology entry consisting of subject-predicate-object triple is put into a candidate set. Next, other words in the input sentence that are not exact matches with domain ontology terms but have a 'closeness' match with terms in

[1] https://youtu.be/V3ssIREEA-Q

the ontology are considered. This 'closeness' match is performed through a mix of phonetic match combined with Levenshtein distance match. The terms that match help identify the corresponding domain ontology entry (with its $<$ subject-predicate-object $>$ triple) is added to the candidate set. This set of candidate genes is a shortlist of the 'genes' of the domain that is probably referred to in the T' sentence. The fittest candidate gene is picked up from this set and replaces the inaccurate words in the T' sentence. The sliding window runs through entire T' sentence and replaces as many inaccurate words as it can and gives us the T'_1 sentence.

2.2 Step 2: Mapping Function

Next, the context of the domain terms in the sentence T'_2 is considered, wherein in the context of domain terms, say A and C, the domain term B is more appropriate than word K that occurs in T'_1. This is done by checking if K is phonetically close to B or not using the fitness function. If K is phonetically close to B then we replace K with B. All such mapping and replace gives us the T'_2 sentence.

2.3 Step 3: ML Repair (Ontology based Learning)

In the third step of repair, to further improve the T'_2 sentence we create a machine learning model (MLmodel) based on selected sentences from a domain corpus that contains relevant ontology terms. The MLmodel uses a LSTM (Long Short Term Memory) network for the prediction of next and previous word given a sub-string. This model consists of an input layer, two hidden layers each containing 400 neurons, and an output layer. Both the hidden layers consists of a LSTM unit. To regularize the MLmodel, we introduce a dropout of 50% between the hidden layers and also between second hidden layer and output layer. We had a dense Word2vec representation with vector size of 300. This Word2vec representation is done by introducing an embedding layer before the first hidden layer. This model gave the perplexity of 188.31 for training data; but reasonably worse perplexity of 432.19 for testing data. We believe this is because of the data sparsity in our dataset.

In the matching and mapping steps, all the corrections are done using the terms in the domain ontology. Thus we assume that all those corrections are correct. Based on this assumption, we mark the genes (sub-strings) and the terms which were corrected in previous steps with <ed> </ed> tags and try to correct the unmarked portions. Since we assume that marked portion is already correct, we can use the marked sub-strings to predict the next (or previous) word and score them with respect to the corresponding unmarked word using the fitness function mentioned above. Based on these scores, we replace the unmarked word with the predicted word. If there is a replacement then we will mark the replaced portion of the string with <ed> tag. All such replacements in T'_2 sentence gives us the T'_3 sentence.

2.4 Step 4: Linguistic Repair

The repaired sentence, T'_3, may need further linguistic adaptation/ repair to remove the remaining errors in the sentence. To achieve this, the repaired ASR sentence is re-parsed and the POS tags are evaluated to find any linguistic inconsistencies, and the inconsistencies are then removed. For example, we may notice that there is a WP tag in a sentence that refers to a Wh-Pronoun, but a WDT tag is missing in the sentence that should provide the Determiner for the Wh-pronoun. Using such clues we can look for phonetically matching words in the sentence that could possibly match with a Determiner and repair the sentence. Linguistic repairs such as these form the genotype to phenotype repair/ adaptation of the sentence. Other rules like "an" cannot come before a consonant etc. can also apply. One can also use an open source Linguistic Tools such as "LanguageTool" to do the repair. Thus we are able to repair T'_3 to get T'' and improve the accuracy of the gpASR output sentence, T'.

3 Factoring the Environment in the Repair

One of the main driving factors of our system is the identification of the environment from which the human speaker speaks the sentences. In that sense we have a environment identification module which is able to identify the type of the environment that the speaker is speaking in. For example, the identified environment could be broadly classified as noisy, silent or fine-grained classified like railway station, canteen, auditorium, raining in the background, road noise, etc. Based on the environment that is identified our method selects the appropriate matching and mapping functions and their thresholds from a pre-trained ML model. This helps suitable repair.

4 Results & Discussion

We have tested our method with publicly available datasets on ASR output of sentences from financial domain available on github (www.github.com/speech-corpus/asr_outputs). Dataset contained 500 sentences collected from videos on company financial quarter results spoken by users. The videos transcription and audio clipping have been taken from Youtube links. The overall improvement in accuracy and Word Recognition Rate (WRR) is shown in Table 1. As we can see there is significant improvement in every stage of the ASR output.

In this paper we have proposed an approach that combines machine learning (ML) and bio-inspired evolutionary development (Evo-Devo) approaches to repair the output of a gpASR. The main contribution of this paper is the introduction of a pipeline which allows for repair of a gpASR output (shown with the help of an example natural spoken domain specific sentence).

Repair %	**Match**	**Map**	**ML**	**Ling**	**Overall**
Speaker 1	2.10	1.5	4.64	0.35	8.59
Speaker 2	3.52	0.65	6.24	0.09	10.5
Speaker 3	4.23	0.29	5.37	0.5	10.39
Speaker 4	2.09	0.54	4.42	0.34	7.39
Speaker 5	3.27	0.19	3.85	0.95	8.26
WRR (Overall)	3.04	0.63	4.90	0.44	**9.01%**

Table 1: 500 sentences corpus

References

[Anantaram *et al.*, 2016] C. Anantaram, Sunil Kumar Kopparapu, Chiragkumar Patel, and Aditya Mittal. Repairing general-purpose ASR output to improve accuracy of spoken sentences in specific domains using artificial development approach. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4234–4235, 2016.

[Fusayasu *et al.*, 2015a] Yohei Fusayasu, Katsuyuki Tanaka, Tetsuya Takiguchi, and Yasuo Ariki. Word-error correction of continuous speech recognition based on normalized relevance distance. In Yang and Wooldridge [2015], pages 1257–1262.

[Fusayasu *et al.*, 2015b] Yohei Fusayasu, Katsuyuki Tanaka, Tetsuya Takiguchi, and Yasuo Ariki. Word-error correction of continuous speech recognition based on normalized relevance distance. In Yang and Wooldridge [2015], pages 1257–1262.

[Gurunath Shivakumar *et al.*, 2018] P. Gurunath Shivakumar, H. Li, K. Knight, and P. Georgiou. Learning from Past Mistakes: Improving Automatic Speech Recognition Output via Noisy-Clean Phrase Context Modeling. *ArXiv e-prints*, February 2018.

[Harding and Banzhaf, 2008] Simon Harding and Wolfgang Banzhaf. *Artificial Development*, pages 201–219. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Kim *et al.*, 2016] Byeongchang Kim, Junhwi Choi, and Gary Geunbae Lee. Asr error management using rnn based syllable prediction for spoken dialog applications. In James J. (Jong Hyuk) Park, Gangman Yi, Young-Sik Jeong, and Hong Shen, editors, *Advances in Parallel and Distributed Computing and Ubiquitous Services*, pages 99–106, Singapore, 2016. Springer Singapore.

[Morbini *et al.*, 2013] Fabrizio Morbini, Kartik Audhkhasi, Kenji Sagae, Ron Artstein, Dogan Can, Panayiotis Georgiou, Shri Narayanan, Anton Leuski, and David Traum. Which asr should i choose for my dialogue system? In *Proceedings of the SIGDIAL 2013 Conference*, pages 394–403. Association for Computational Linguistics, 2013.

[Peng *et al.*, 2013] F. Peng, S. Roy, B. Shahshahani, and F. Beaufays. Search results based n-best hypothesis rescoring with maximum entropy classification. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 422–427, Dec 2013.

[Suhm *et al.*, 2001] Bernhard Suhm, Brad Myers, and Alex Waibel. Multimodal error correction for speech user interfaces. *ACM Trans. Comput.-Hum. Interact.*, 8(1):60–98, March 2001.

[Tufte, 2009] Gunnar Tufte. From evo to evodevo: Mapping and adaptation in artificial development. In Wellington Pinheiro dos Santos, editor, *Evolutionary Computation*, chapter 12. InTech, Rijeka, 2009.

[Tür *et al.*, 2013] Gökhan Tür, Anoop Deoras, and Dilek Hakkani-Tür. Semantic parsing using word confusion networks with conditional random fields. In Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier, editors, *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, pages 2579–2583. ISCA, 2013.

[Yang and Wooldridge, 2015] Qiang Yang and Michael Wooldridge, editors. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. AAAI Press, 2015.