Improving ASR Recognized Speech Output for Effective Natural Language Processing

C. Anantaram, Sunil Kumar Kopparapu, Nikhil Kini, Chiragkumar Patel
Innovation Labs
Tata Consultancy Services Ltd.
Delhi & Mumbai, India
{c.anantaram, sunilkumar.kopparapu, nikhil.kini, patel.chiragkumar}@tcs.com

***Abstract*—The process of converting human spoken speech into text is performed by an Automatic Speech Recognition (ASR) system. While functional examples of speech recognition can be seen in day-to-day use, most of these work under constraints of a limited domain, and/or use of additional cues to enhance the speech-to-text conversion process. However, for natural language spoken speech, the typical recognition accuracy achievable even for state-of-the-art speech recognition systems have been observed to be about 50 to 60% in real-world environments. The recognition is worse if we consider factors such as environmental noise, variations in accent, poor ability to express on the part of the user, or inadequate resources to build recognition systems. Natural language processing of such erroneously and partially recognized text becomes rather problematic. It is thus important to improve the accuracy of the recognized text. We present a mechanism based on evolutionary development to help improve the overall content accuracy of an ASR text for a domain. Our approach considers an erroneous sentence as a zygote and grows it through an artificial development approach, with evolution and development of the partial gene present in the input sentence with respect to the genotypes in the domain. Once the genotypes are identified, we grow them into phenotypes that fill the missing gaps and replace erroneous words with appropriate domain words in the sentence. In this paper, we describe our novel evolutionary development approach to repair an erroneous ASR text to make it accurate for further deeper natural language processing.**

Keywords-evolutionary development; artificial development; speech recognition; natural language processing.

I. INTRODUCTION

Speech and natural language interfaces are becoming rather important means of communication with enterprise systems. As more and more end-users of enterprise applications are targeted (e.g., online shopping, banking), the demand for human-speech and natural language interfaces to such online application systems seems to be growing. Automated recognition of the user's speech into natural language text and then processing that text is very important. It is imperative that this process becomes rather accurate. Similarly, some of the most used channels for customers to interact with human service-agents in an enterprise are still the telephony channel [1]. In several cases the customer actually speaks to a human agent to get an answer to the problem that he/she might face. With an increasing customer base and with a corresponding increase in transactional volumes, support personnel are rather stretched, and this results in a delay in responding to the customer. ASR systems with deep natural language processing have been found to help reduce this load by automatically routing calls to automated helpdesks, provided such recognition and processing is of good accuracy.

With self help solutions becoming popular, there has been a spurt of growth in Voice User Interfaces (VUI). A typical VUI-based solution would take as input a spoken utterance, recognize (speech to text conversion) the utterance, interpret it (natural language understating), fetch an answer in response from a structured or unstructured database, and communicate the response (text to speech) to the user. Clearly, the process of interpretation of the spoken query is subject to the accuracy of the speech recognition engine that converts the spoken speech into text. Most of the functional examples of speech recognition in day-to-day use work under the constraints of limited domain, and/or use of additional cues to enhance the speech-to-text conversion process. Typical recognition accuracies for state-of-the-art speech recognition systems have been observed to be about 50 to 60% for natural language spoken speech. Environmental noise, variations in accent, poor ability to express on the part of the user, or inadequate resources to build recognition system also affect the accuracy adversely. Natural language processing of such erroneously and partially recognized text becomes rather problematic. It is thus important to improve the accuracy of the recognized text. While it is desirable to have better speech recognition mechanisms through better training sets covering more sample scenarios for the speech recognition engine, the question of interest here is, how can we improve the accuracy of the recognized text that is output from an ASR engine for a particular domain?

We examine this problem and present a mechanism based on evolutionary development (evo-devo) processes [2][3] to help improve the overall content accuracy of a recognized text for a domain. Our approach considers an erroneous input sentence as a zygote and grows it through an artificial development approach, with evolution and development of the partial gene present in the input sentence with respect to the genotypes in the domain. Once the genotypes are identified, we grow them into phenotypes that fill the missing gaps and replace erroneous words with appropriate domain words in the sentence. This process of

 ISBN: 978-1-61208-381-0

artificial rejuvenation improves the accuracy of the sentence, which can then be processed by a natural language processing application such as question answering [1][4][5], and workflow management [6]. Thus, the main contribution of the paper is in terms of proposing a bio-inspired novel procedure to repair the erroneously recognized text output by a speech recognition engine, in order to make the output text suitable for deeper natural language processing.

The rest of this paper is arranged as follows: in Section II, we describe the current state of Speech and Natural Language Processing (NLP) self help systems. In Section III, we describe our proposed evolutionary development approach, and give a detailed example of how it works in Section IV. In Section V, we show the use of the ASR repair approach in a self help system scenario.

II. STATE OF ART

Although a number of attempts have been made to build speech and natural language interfaces for different applications [1][4][6][7], the attempts to build accurate speech and natural language processing systems for a domain is far from satisfactory [8].

While there are several ASR engines, both commercial and otherwise, their performance is highly dependent on the language, accent, dialect, and environmental noise. Even for the best of the ASR engine the accuracy of the recognition is as little as 50-60% for spoken natural language sentences [8]. Interpreting erroneously recognized text will result in erroneous interpretation of the query and the failure of the self-help solution to assist in addressing the queries of the users.

A typical speech recognition process is shown in Fig. 1. The input is the speech signal and the output is the recognized text. However, to achieve this simple process of speech to text conversion, there is a need for training. The training aspect involves the use of a well-structured speech corpus of a language containing several hours of speech to create Acoustic Model (AM) and Language Model (LM) for that particular language. The acoustic and language model can be assumed as a statistical representation of the spoken content, the dialect and the accent of a language.

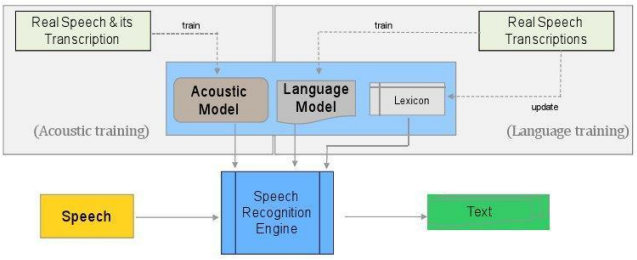

Figure 1. ASR Framework

There are several ways to improve the recognition performance of an ASR: (a) Fine tuning the ASR engine - This requires elaborate training, which in turn needs a rich amount of speech corpora. Very often, especially for not very popular languages, there is a dearth of speech corpora, and building a corpora is time consuming and expensive. There has been some work on building frugal speech corpora by exploiting the multimedia information on the Internet [9][10]. (b) Restricting what the user can say: This results in a restriction in the aspect of usability [11] and the VUI becomes user unfriendly.

In this paper, we examine the problem of improving the output of a speech recognition engine and present a mechanism based on evolutionary algorithms that help improve the overall content accuracy of a recognized text for a domain.

Much work has been done on automatic error detection in ASR output (a survey of this is presented in [12]), and also facilitating error correction for the user through easy-to-use interfaces [13][14], but to the best of our knowledge, there are no methods to automatically correct an ASR's output. Our work concerns not only detection, but automatic error correction after ASR. Previous work on this can be found in [15][16] [17].

As mentioned by Ringger and Allen [15], the reason this is an important problem is it allows for the ASR system to be a black box, whose output can be processed separately. This is particularly useful for improving proprietary systems where access to improve the system internally is not available. Also, such a post-correction system provides greater flexibility in terms of modeling domain variations and rescoring the output, in ways that are not possible in the ASR system [15].

Another example of error correction on ASR output can be found in [16]. In our research, we make use of evo-devo based repair methods that introduce an element of randomness in error correction, which is useful when training data for error correction is small or absent.

Our work is mainly concerned with ASR output that acts as input to another system, in this case, an NLP system that can retrieve answers to questions posed. IBM Watson Engagement Advisor [18] is an example of a commercial system that processes questions posed in text form and finds answers to them. Its basic working principle is to parse keywords in a clue while searching for related terms as responses [5]. Watson has deficiencies in understanding the contexts of the clues. Also, the setup cost and initial investment is too high, which makes it less suitable for being easily accessible and usable.

III. OUR APPROACH

We consider the situation where a speech recognition engine takes a sentence spoken by a person as input and outputs a text sentence that is not an accurately recognized text. For example, when a person spoke the following sentence "Give me all the existing customers", the output sentence from our ASR engine was "The the contact existing customers". The question we tackle here is, how do we repair (or grow) the sentence back to the original sentence as intended by the speaker? It is in this context that our approach considers an erroneous input sentence as a zygote and grows it through an artificial development approach, with evolution and development of the partial gene present in the input sentence with respect to the genotypes in the

 ISBN: 978-1-61208-381-0

domain. Once the genotypes are identified, we grow them into phenotypes that fill the missing gaps and replace erroneous words with appropriate domain words in the sentence. This process of artificial rejuvenation improves the accuracy of the sentence, which can then be passed onto a natural language processing application for further processing. The overall processing is described below.

A. Identifying the genotypes for the sentence

We consider each sentence as an individual in the population, i.e., as a zygote, and identify the genes that can apply on the sentence through partial match of concepts of the sentence with the ontological rules. The set of genes in the sentence form the genotypes.

1) Seeding the gene set for the domain

The ontology of a domain describes the domain terms and their relationships. A seed ontology details the meta-relations that are defined in the domain [4], for example "project has status", "project has start_date" etc.

The application data (i.e., the database of the business application) is taken as input to instantiate the terms and relationships defined in the seed ontology in order to form the actual ontology of the domain [4]. This ontology (stored as a Resource Description Framework graph) forms the basic genes of the domain and their relationships with a <subject-predicate-object> structure for each of the genes.

2) Identifying the genotypes in the input sentence

We match sub-parts (or sub-strings) of the input sentence with the genes of the domain. The match will be partial due to the error present in the input sentence. The genes that match the closest are picked up, provided they satisfy fitness criteria.

B. Simulate the evolution and development process of the genotypes

Once the basic genes are identified, we develop the genes to better fit the situation on hand with evolution and development of the genes, and then score against a fitness function and select the "fittest" gene that survives. This gives us the set of genotypes that will form the sentence.

C. Developing the genotype to produce / extract its phenotype

The overall genotypes are collated together to form the input sentence. In this context some of the genotypes may need further development to form the final sentence that the user actually intended.

D. Evaluate the developed sentence

The developed sentence is then presented to a human Oracle who ranks the sentence if he/she deems it as a better fit (i.e., more accurate) for the domain.

IV. A DETAILED EXAMPLE

Let us consider an input speech where the user says the following sentence: "What is the status of the project in which Vinay is a team member?" A general purpose speech recognition system recognizes this spoken sentence as follows: "What is a hat us of the project in itch Vinay is a tea ember?" Thus the recognized sentence has errors and is not accurate.

We run the artificial development approach on this input sentence in order to repair the input and make it more accurate.

A. Identifying the genotypes for the sentence

We assume that the domain has the following ontology that is formed from the seed ontology and application data:

ds:project	ds:has	ds:status
ds:project	ds:has	ds:start_date
ds:project	ds:has	ds:role
ds:role	ds:is	ds:Team_member
ds:Vinay	ds:allocated	ds:ArtDevPrj
ds:Vinay	ds:role	ds:Team_member
ds:ArtDevPrj	ds:status	ds:Active

(The "ds:" prefix above is the namespace for this schema)

Firstly, the ASR output is parsed for identifying the parts of speech in the sentence. This process identifies the nouns, verbs, adjectives and adverbs in the sentence. Since the sentence itself is inaccurate, the parts of speech may not be accurate. For our example, parts of speech tagging gives the following output: 'what/WP/what is/VBZ/be a/DT/a hat/NN/hat us/PRP/US of/IN/of the/DT/the project/NN/project in/IN/in itch/NN/itch vinay/NN/vinay is/VBZ/be a/DT/a tea/NN/tea ember/NN/ember'.

Using these identified parts of speech (especially nouns), the relevant subject-predicate-object of the domain that are referred in the sentence are marked (called partially matching genes). This is done through a partial-match algorithm wherein the ASR output sentence is matched with the ontology. The sub-string 'project' matches with an entry in the ontology, and thus a partial gene is triggered. The parts of speech identified before 'project' help narrow down to two possible genes: 'project has status' and 'project has start_date'. Similarly, the sub-string 'Vinay' matches with two entries in the ontology, namely ds:Vinay ds:allocated ds:ArtDevPrj and ds:Vinay ds:role ds:Team_member and thus, a partial gene is triggered. The parts of speech following Vinay, especially "is/VBZ/be a/DT/a" help identify that a Verb and a Determiner follows and some relationship with 'Vinay' is expected. Thus, both these ontology entries are considered as genes of the input sentence.

The set of all possible genes that are identified in the ASR output sentence are considered as the genotypes in the sentence that need to be evolved and developed.

B. Evolution and Development of the genotypes

Using phonetic match, i.e., match of phonemes in words, between the ASR output sentence and the identified genes, we develop the partial gene present in the sentence. Phonetic match algorithms, such as Soundex [19], can be used for such a match. Thus 'tea ember' and 'team member' phonetically close, and the gene 'Vinay role Team_member' is selected. Similarly 'hat us' has a close match with 'status' rather than 'start_date'. Hence, the fitness of the gene

'project has status' is better than the fitness of the gene 'project has start_date' in this context.

We use such fitness functions to select the genes that need further development for the ASR output. Currently in our approach, the development process is simulated by a replacement of the partially identified genes with the genes that are most likely. One can later introduce a more elaborate development process. Hence the ASR output sentence is modified to become "What is a status of the project in itch Vinay is a team_member?"

The genotypes in the sentence are 'status of the project' and 'Vinay is a team member'. The rest of the sentence needs to be further developed to make it more accurate.

C. Developing the genotypes to produce its phenotype

We now have a sentence that has been repaired through evolutionary development method that needs further development to make it accurate.

We parse the re-written sentence again to identify its new parts of speech. Thus, for the modified sentence we get: "what/WP/what is/VBZ/be a/DT/a status/NN/status of/IN/of the/DT/the project/NN/project in/IN/in itch/NN/itch vinay/NN/vinay is/VBZ/be a/DT/a team_member/NN/team_member".

We notice that there is a WP tag that refers to a Wh-Pronoun. However a WDT tag is missing that refers to a Wh-Determiner in the sentence. Using this clue we look for a phonetically matching word that could possibly match with a Wh-Determiner. Our match-function identifies "itch" as more phonetically close to "which" (that is a Wh-Determiner). This is a second-level fitness function and thus we can rewrite the modified sentence as follows "What is a status of the project in which Vinay is a team_member?" This sentence is now ready for accuracy evaluation.

D. Evaluate the developed sentence

In this step, we evaluate the accuracy of the artificially developed sentence to determine if it is a better fit for the domain than the ASR output. At present, we assume the presence of an oracle of the domain to evaluate the accuracy of the developed sentence. Later on, such a process can also be automated by formally defining accuracy and developing precise mechanisms to measure it.

The output of the artificial development approach is presented to an oracle who evaluates the accuracy of the sentence. The parts of speech for this newly developed sentence: "what/WP/what is/VBZ/be a/DT/a status/NN/status of/IN/of the/DT/the project/NN/project in/IN/in which/WDT/which vinay/VBP/vinay is/VBZ/be a/DT/a team_member/NN/team_member". The sentence has more ontology terms and relationships of the domain and the parts of speech are also complete. Thus, the oracle marks the newly developed sentence as accurate.

The artificially developed sentence, which is now marked as accurate, can now be processed by deeper natural language processing applications such as question-answering/workflow management/self help tools [1][4] [18].

V. SELF HELP CASE STUDY

In the scenario of a retail outlet that has a large number of products, lots of promotion offers, and catering to many customer queries, self help becomes a very important aspect of customer experience. Consider the following query asked by a customer via an interactive audio self help system:

User: Which camcorders have more than 20% discount?

The ASR system processes the speech and converts it to text. However, as described above, the output text may be erroneous. In this example:

ASR output: Itch came orders have more the 20% this count?

We will need to repair the output since there are recognition errors. Following our artificial development method described above we get the repair steps as:

Genes identified: Camcorder, discount
Genotype repair**:** "came orders" repaired to "camcorders" and "this count" repaired to "discount"
Phenotype repair**:** "Itch" repaired to "which"
Repaired sentence**:** "Which camcorders have more than 20% discount?"

This sentence is passed onto the self help question answering system. The output of the system is:

System**:** The Camcorders are
DXG 3MP Digital Camcorder - DXG-301V
Panasonic Mini DV Camcorder
Aiptek IS-DV2 Digital Camcorder
Panasonic 2.8" LCD Digital Camcorder with 3CCD Technology - Silver (SDR-S150).

Thus we can see how the speech and natural language system with repair of ASR output has answered the customer's question in a self help situation and improved the overall customer experience.

VI. CONCLUSION

We have described a mechanism to artificially develop and improve an ASR output sentence to make it more accurate for a domain by following the evo-devo based artificial development approach. The idea is to work with the inaccuracies in the recognition and repair/develop/grow-out the error and replace it with a more accurate sentence that can be processed further by a natural language processing system. This helps in better speech-and-natural language interface systems for enterprises and aids in self help systems.

VII. REFERENCES

[1] http://www-03.ibm.com/innovation/us/watson/science-behind_watson.shtml [Retrieved: January, 2015]

 ISBN: 978-1-61208-381-0

[2] S. Harding and W. Banzhaf, "Artificial Development," http://www.cs.mun.ca/~simonh/publications/evodevbookchapter.pdf [Retrieved: January, 2015]

[3] G. Tufte, "From Evo to EvoDevo: Mapping and Adaptation in Artificial Development," http://www.intechopen.com/books/evolutionary-computation/from-evo-to-evodevo-mapping-and-adaptation-in-artificial-development [Retrieved: January, 2015]

[4] S. Bhat, C. Anantaram, and H. Jain, "Framework for text-based conversational user-interface for business applications." In Proceedings of the 2nd international conference on Knowledge science, engineering and management, pp. 301-312. Springer-Verlag, 2007.

[5] IBM Watson - http://en.wikipedia.org/wiki/Watson_(computer) [Retrieved: January, 2015]

[6] S. Bhat, C. Anantaram and H. Jain, "An architecture for intelligent email-based workflow interface to business applications," International Conference on Artificial Intelligence (ICAI-2008), WORLDCOMP'08, Las Vegas, USA, pp. 344-350. July 14-17, 2008.

[7] A. Imran, S. K. Kopparapu, "Building a Natural Language Hindi Speech Interface to Access Market Information," The Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG, Hubli, 2011.

[8] C. Lee, S. Jung, K. Kim, D. Lee, and G. G. Lee, "Recent Approaches to Dialog Management for Spoken Dialog Systems," Journal of Computing Science and Engineering, vol. 4, no. 1, March 2010.

[9] I. Ahmed and S. K. Kopparapu, "Speech recognition for resource deficient languages using frugal speech corpus," in Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on, 2012, pp. 750–755.

[10] S. K. Kopparapu and I. Ahmed, "A Frugal Method and System for Creating Speech Corpus," Indian Patent 2148/MUM/2011; Jul 28, 2011.

[11] S. K. Kopparapu, "Voice Based Self Help System: User Experience Vs Accuracy," Book Chapter, Innovations and Advances in Computer Sciences and Engineering edited by Tarek Sobh, Springer, 1st Edition, ISBN-13: 978-9048136575, March 2010.

[12] Y. Shi, "An investigation of linguistic information for speech recognition error detection," PhD diss., University of Maryland, Baltimore County, 2008.

[13] J. Ogata and M. Goto. "Speech repair: quick error correction just by using selection operation for speech input interfaces," In INTERSPEECH, pp. 133-136. 2005.

[14] D. Harwath, A. Gruenstein, and I. McGraw. "Choosing Useful Word Alternates for Automatic Speech Recognition Correction Interfaces." In Fifteenth Annual Conference of the International Speech Communication Association. 2014.

[15] E. K. Ringger and J. F. Allen. "Error correction via a post-processor for continuous speech recognition," Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on, vol. 1, pp. 427-430. IEEE, 1996.

[16] M. Jeong, B. Kim, and G. Lee. "Using higher-level linguistic knowledge for speech recognition error correction in a spoken Q/A dialog," Proceedings of the HLT-NAACL special workshop on Higher-Level Linguistic Information for Speech Processing, pp. 48-55. 2004.

[17] R. López-Cózar and David Griol. "New technique to enhance the performance of spoken dialogue systems based on dialogue states-dependent language models and grammatical rules," In INTERSPEECH, pp. 2998-3001. 2010.

[18] IBM WATSON Engagement Advisor - http://www-03.ibm.com/innovation/us/watson/watson_for_engagement.shtml [Retrieved: January, 2015]

[19] http://en.wikipedia.org/wiki/Soundex [Retrieved: January, 2015]

 ISBN: 978-1-61208-381-0