Reducing Error Rate for ASR using Semantic Error Correction Approach

Theint Zarni Myint, Myo Kay Khaing
University of Computer Studies,Yangon
chutpit@gmail.com

Abstract

Many application environments have already used speech interface. But the low speech recognition rate makes it difficult to extend its application to new fields. In the human-computer interaction through spoken dialogue are being investigated. Automatic Speech recognition (ASR) is the process of converting a spoken speech into text that can be manipulated by the computer. The state of the art in automatic speech recognition has reached the point that searching for and extracting information from large speech repositories. This system presents semantic-oriented approach to correct both semantic and lexical errors, which is also more accurate for especially domain-specific speech error correction. This paper demonstrates the superior performance of this approach and some advantages over previous lexical-oriented approaches by comparing such approaches. Experiments carried out on various speeches in English syllable indicated a successful decrease in the number of errors and an improvement in overall error correction rate.

Keywords- Automatic Speech Recognition (ASR), Semantic oriented approach, Lexical oriented approach.

1. Introduction

With the ever increasing number of computer-based applications modern digital computers are no more solely used for crunching number and performing high-speed mathematical computations. Automatic Speech recognition (ASR) is one of the most involving computing fields that has already been exhaustively employed for an assortment of applications including but not limited to automatic telephone services (ATS) , voice user interface (VUI), voice-driven industrial control systems (ICS), speech-driven home automation system (Demotic), speech dictation systems, and automatic speech to text systems (STT). Lately, ASR has been of great attraction to computer researchers, manufacturing, and consumers [7].

Some researchers investigated the performance of spoken queries in NTCIR collections [1]. They evaluated a variety of speakers, and calculated the error rate with respect the query term, which is a keyword used for the retrieval. They showed that the WER of the query terms was generally higher than that of the general words irrespective of the speakers. In other words, recognition of content words related to the information retrieval (IR) performance was more difficult than that of normal words. So, they introduced a method to improve the precision of speech driven IR by suggesting a new type of IR system tightly integrated with a speech input interface. In their system, document collection provides an adaption of language model of s in the ASR, which results in a drop of word error rate.

For this reason, some appropriate adaptation techniques are required for overcoming speech recognition errors such as post error correction. ASR error correction can be one of the domain adaptation techniques to improve the recognition accuracy, and the primary advantage of error correction approach is its independence of the specific speech recognizer.

Some researchers used post error correction that could correct any correct errors by matching the string in the transcription with lexical patterns in the database. But their system has a disadvantaged in the correction is only feasible to train lexical error pattern. Some applied the noisy channel model to the correction of errors in speech recognition. All of their systems based on an lexical information of words have shown some successful results. But they have major drawback such as the performance of such systems depends on the size and the quantity of speech recognition results or the database correct error string.

So, this paper suggests a more improved and robust semantic error correction approach, which can be integrated into previous lexical-based approaches. This system uses high level syntactic and semantic information of the words in speech transcription. Therefore, this system obtains semantic information from a knowledge base and special domain dictionary to contain some domain specific knowledge to the target application.

2. Noisy Channel Error Correction Model

The noisy channel error correction framework has been applied to a wide range of problems, such as spelling correction, statistic machines translation, and

ASR error correction. The key idea of noisy channel model is that we can model some channel properties through estimating the posterior probabilities.

2.1. Word-Based Channel Model

The problem of ASR error correction can be stated in the model as follows: For an input sentence, O= o_1 , o_2, ……. o_n produces as the output sequence of ASR, find the best word *sequence*, *W*= w_1, w_2……..., w_n, that maximizes the posterior probability P(W/O). Then applying Bayes' rule and dropping the constant denominator, write as follow:

$$\hat{W} = argmax_W P(W/O) = argmax_W (P(W)P(O/W) \quad (1)$$

Therefore, we solve a noisy channel model for ASR error correction, with two components, the source model P(W) and the channel model P(O/W). The probability P(w) is given by the language model and can be decomposed as:

$$P(W) = \prod_i P(\omega_i/\omega_{1,i} - 1) \quad (2)$$

The distribution P (W) can be defined using n-grams, structured language mode;, or any other tool in the language modeling.

We called this model is word-based channel model, because this model is based on the word-to-word transformation. The word based model based on inter-word substitutions, so it requires enough results of ASR and transcription pairs. The substitution probability is based on the whole word-level, this fertility model requires enormous training data. Considering the cost of building the enough amounts of correction pairs, we need a smaller unit than a word for over -coming the data-sparseness.

Therefore, the fertility model can deal with (TO LEAVE, TOLEDO) substitution. But this improved fertility model only slightly increased the accuracy in experiments [] and we think the major reason is due to the data-sparseness problem. So, this system use syllable based channel model, discuss in next section.

3. Applying Syntactic and Semantic Knowledge

In some application such as spelling error correction or optical character recognition (OCR) error correction, natural language processing (NLP) researchers identified five levels of errors in text: (1) a lexical level, (2) a syntactical level (3) a semantic level, (4) a discourse structure level, and (5) a pragmatic level. In spelling correction and OCR correction problem, correction schemes mainly have focus on non-words errors at the lexical level, which is an isolated word correction problem.

But, errors of speech recognition tends to be continuous word errors which should be better classified into syntactic and semantic level because the recognizer only produces word sequences existing in a lexicon. So, this section presents continuous word error detection and correction, using syntactic and semantic knowledge and pipelines this high-level error correction method with the syllable based channel model.

3.1. Syllable Based Channel Model

The segmentation into syllable-like units is based on the phonetically alignment from a phoneme recognizer with long temporal context. Then each speech segment between two pauses is equally divided based on the number of vowels in this segment. Each vowel is considered as the nucleus of a syllable. In a second step, the estimated syllable boundary between two vowels can be shifted with regard to the measured pitch at the potential boundary candidates.

This system suggests an improved channel model for smaller training data. If we can use smaller unit such as letter, phoneme or syllable than word, relatively smaller training set is needed. A syllable based channel model, which can deal with syllable-to-syllable transformation, is suggested for dealing with inter-word transformation.

3.1. 1. Modeling

Suppose S= s_1, s_2, s_3 ,…,s_n is a syllable sequence at ASR output and W=w_1, w_2, w_3 ,…,w_n is a source word sequence. Find the best word sequence $\hat{W}$ as follows:

$$\hat{W} = \text{argmax}_W P(W/S) \quad (3)$$

The system applies the same Bayes' rule and decomposes the syllable to word channel model into syllable to syllable channel model.

$$P(\omega/s) = \frac{P(s/\omega)P(\omega)}{P(s)} \propto P(s/\omega)P(\omega)$$

$$\approx P(s/x)P(x/\omega)P(\omega)$$

So, final formula can be written as:

$$\hat{W} = argmax_W(P(W)P(X/W)P(S/X) \quad (4)$$

Here, *P*(S/X) is the probability of a syllable-to-syllable transformation, where P(S/X) is probability of s to s transformation, where X= x_1, x_2, x_3,…,x_n is a source syllable sequence. P(X/W) is a word model, convert syllable lattice into word lattice. The conversion can be done efficiently by dictionary look-up. The model is similar a standard Hidden Markov model (HMM) of continuous speech recognition. In speech recognition system, P(S/X) can be acoustic model in single-to-phoneme level, and P(X/W) can be a pronunciation dictionary. Then, this paper applied the fertility into our syllable-to-syllable channel model. This system set the maximum 2-fertility of syllable, which was determined experimentally.

3.2. Decoding the Model

Given a syllable sequence S, the system wants to find $argmax_W$ = (P (W) P(X/W) P(S/X)). It returns an N-best list of candidates according to the models, and then rescore the candidates by taking into the language model probabilities and rescores the candidates, the system used Viterbi search algorithm to find the best sequence. For implementation of candidate generation, the system will store the syllable channel probabilities P(s_i/x_i).The system can generate a candidate word sequence network using syllable channel model and a lexicon. The system can find optimal sequence which has the best probability through Viterbi decoding by including a language model.

3.3. Training the Model

To train the model, we need a training data consisting of {X, S} pairs which are manually transcribed strings and ASR outputs. And, this system aligns the pair based on minimizing the edit distance between x_i and s_i by dynamic programming. For example, (TOLEAVE, TOLEDO) pair can be divided into (TO, TO), (L, L), (EA,E), and (VE,DO) with the fertility 2.

This system can then calculate the probability of each substitution P (s_i/x_i) by Maximum Likelihood Estimation (MLE). Let C (x_i) be the frequency of source syllable and C(x_i,s_i) be frequency events where x_i substitute s_i. Then,

$$P_{MLE}(s_i/x_i) = \frac{C(x_i/s_i)}{C(x_i)} \quad (5)$$

4. Semantic-oriented Error Correction Process

Now, the working mechanism of post error correction of speech recognition shows the result using the domain knowledge of template database and domain-specific dictionary. Figure 1 is a schematic diagram of the post error correction process.

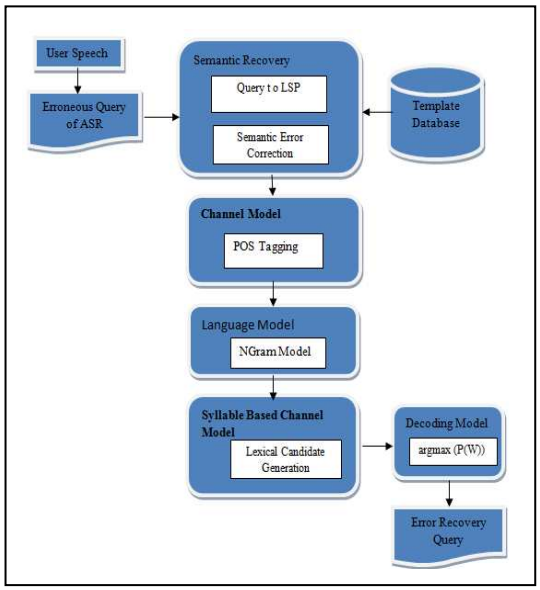

Figure1. Process of Semantic-oriented Error Correction

The process is divided into two stages: a syntactic/semantic recovery and a lexical recovery stage. In the semantic error detection stage, a recognized query is converted into the corresponding LSP. And then the system used a pre-collected template database to recover the semantic level errors, and the technique for searching most similar templates are based on a minimum edit distance dynamic programming search which has been used as a

similarity search in many areas such as spelling correction.

Semantic error correction is performed by replacing these syntactic and/or semantic errors using a semantic confusing table. We used a pre-collected template database to recover the semantic level errors, and the technique for searching most similar templates are based on a minimum edit distance dynamic programming search, which has been used as a similarity search in many areas such as spelling correction, OCR post correction, and DNA sequence analysis.

The semantic confusion table provides the matching cost, which can be semantic similarity, to the dynamic programming search process. This system compute the minimum edit distance between the erroneous LSP's and the template LSP's in the template database using similarly cost, which has the minimum distance among them. At this stage, replaced LSP elements can provide some clues of the recognition errors and the original query meaning to the next lexical recovery stage. Moreover, candidate error boundary can also be detected by this procedure.

After this procedure, lexical recovery is performed in the next stage. Recovered semantic tags and the erroneous queries produced by ASR are the clues of lexical recovery. Erroneous query and recovered template query are aligned by dynamic programming again, after which some lexical candidates are generated by our improved syllable-based channel model. Figure 2 shows an example of semantic error correction process using the same data.

Figure 2. Examples of Semantic Error Correction

4.1. Lexical-Semantic Pattern (LSP)

A lexical-semantic pattern (LSP) is a structure where linguistic entries and semantic types are used in combination to abstract certain sequences of the words in a text. In an LSP linguistic entry consists of words, phrases and part-of-speech (POS) tags such as 'YMCA,' 'Young Men's Christian Association,' and 'NNP' (proper noun, singular). Semantic types consist of semantic classes and domain-specific semantic classes. The common semantic tags again include attribute-values in database, such as '@company' for a company name like 'IBM' and pre-define 83 semantic category values such as '@location' for location name like 'Yangon'. Figure3 show an example of predefined common semantic category which will be used in ontology dictionary.

@a_lang	@event	@magazine	@person	@unit_area
@action	@family	@mammal	@phenomenon	@unit_count
@aircraft	@fish	@month	@planet	@unit_date
@belief	@food	@mountain	@plant	@unit_length
@bird	@game	@movie	@position	@unit_money
@book	@god	@music	@reptile	@unit_power
@building	@group	@nationality	@school	@unit_rate
@city	@language	@nature	@season	@unit_size
@color	@living thing	@newspaper	@sports	@unit_speed
@company	@location	@ocean	@state	@unit_temperature
@continent	@exam	@organization	@status	@unit_time
@country	@hobby	@method	@subject_area	@unit_volume
@date	@law	@address	@substance	@unit_weight
@direction	@level	@appliance	@team	@unit_age
@disease	@living_part	@art	@transport	
@drug		@computer	@week day	
		@course	@picture	
		@deed	@river	
			@room	
			@sex	

Figure3. Common semantic category values

In domain-specific application, the domain-specific semantic classes and well defined semantic classes are required. The domain-specific semantic classes include special attribute names in database such as '%action' for active and 'inactive', and semantic categories names, such as '%hobby' for 'reading ' and 'recreation' for which the users wants a specific meaning in the application domain. Moreover, the system used the classes to abstract out several synonyms into single concept. For example, a domain specific semantic class '%question' represents some word such as 'question', 'query', 'asking', and 'answer'.

4.1.1. Query to LSP Translation

Query to LSP translation transforms a given query into a corresponding LSP and the LSP's enhance the coverage of extraction by information abstraction through many to one mapping between queries and LSP. The words in a query sentence are converted into the LSP through several steps. First, a morphological analysis is performed, which segments a sentence of words into morphemes, and adds part-of-speech (POS) tags to the morphemes (Lee et al.,

2002). Second, Noisy Error (NE) recognition discovers all the possible semantic types for each word by constructing a domain dictionary and an ontology dictionary. Finally, NE tagging selects a semantic type for each word so that a sentence can be mapped into a suitable LSP sequence by searching several types in the semantic dictionaries.

4.2. Structure of Domain Knowledge

For semantic-oriented error correction, this system built a domain knowledge, which consists of a domain dictionary and ontology dictionary, and template queries. Query sentences are semantically abstracted by LSP's and are automatically collected for template database. Database of template queries of the source statements which are used for actual error detection and correction task after speech recognition. The template queries are automatically acquired by the Query-to-LSP translation from the source statements using domain dictionary and ontology dictionary.

Assuming that some speech statements for a specific target domain are predefined, a record of the template database is composed of a fixed number of LSP elements, such as POS tags, semantic tags, and domain-specific semantic classes. Table 1 shows an example of template abstracted by LSP conversion in a predefined domain of "on-line education".

Table1. Example of a template abstracted by LSP

Phrases	LSP
Reading trainer **Fairy tale trainer** **Recreation coach**	**%hobby@position**

4.2.1. Domain Dictionary

The domain dictionary is a subset of the general semantic category dictionary, and focuses only on the narrow extent of the knowledge it concerns, since it is impossible to cover all the knowledge of the world in implementing an application. On the other hand, the ontology dictionary for common semantic classes reflects the pure general knowledge of the world; hence it performs a supplementary role to extract semantic information. The domain dictionary provides the specific vocabulary which is used in semantic representation tasks of a user query and the template database.

5. Experiments and Results

In the experiments, speech recognition was performed several experiments on the domain of vehicle telemetric IR. The speech transcripts used in the experiments were composed of 462 queries, which were collected by one female speaker in a real application. For semantic-oriented error correction, this system built domain knowledge for target domain. This system constructed 3.195 entries of domain dictionary and 13165 entries of ontology dictionary and 436 semantic templates generated automatically using domain dictionary and ontology dictionary.

In the results, this system use word error rate (WER) to measure error correction performance. The minimum edit distance between two words is originally defined as the minimum number of deletions, insertions, and substitutions required transforming one word into the other. So,

$$WER = \frac{|S_{\omega}|+|I_{\omega}|+|D_{\omega}|}{W_{truth}} \quad (6)$$

The system focus on the detection of error rate operates the ASR. Figure 3 presents the experiments results of WER of baseline ASR, word based channel model, syllable based channel model and combined syllable-based channel model with LSP semantic correction model. The performances of baseline system were about 79% ~ 81% on the utterances in IR domain. This results shows that the semantic error correction of speech recognition result in a viable approach to improve the performance.

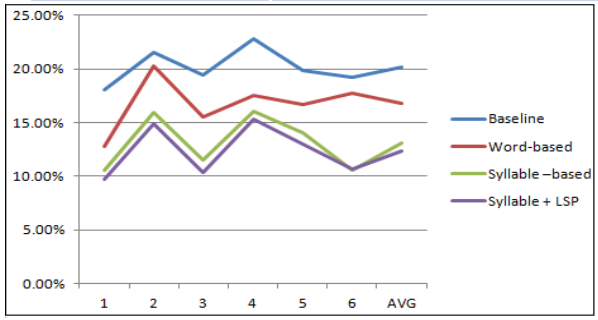

Figure3. Detection Error Rate on Same Testing set

6. Conclusion

The system proposed an improved syllable-based noisy channel model for semantic-oriented approach in a speech recognition error correction. The previous approaches require enormous results of ASR and are dependent on specific speaker and environment. This system takes in far smaller training corpus and it is possible to implement the method easily and in a short time to obtain the better error correction rate. The semantic-oriented approach has more advantages over lexical based ones since it is less sensitive to each error patterns. The LSP correction has a high possibility of generating semantically correct correction due to the massive use of semantic contexts. So, this system shows a high performance. For experiments, the system use trigrams language model generated by SRILM toolkit. The training program for channel model is made by the system.

References

[1] A. Fujii, Katunobu Itou, and Tetsuya Ishikawa. 2002. Speech driven Text Retrival: Using target IR collections for Statical Language Model Adaptation in Speech Recognition.

[2] E Izumi, K Uchimoto, T Saiga, T Supnithi, "Automatic error detection in Japanese learners, English spoken data", Proc. ACL.2003.

[3] F . James Allen and Eric K. Ringger. 1996. A fertility model for post correction of continuous speech recognition ICSLP'96, 897-900.

[4] Juhui An, Seungwoo Lee, and Gary Geunbae Lee. 2003.Automatic acquisition of Named Entity tagged corpus from World Wide Web. In Proceedings of the 41st annual meeting of the ACL (poster presentation).

[5] Jung, M Jecong, and G G Lee, "Speech recognition error correction using maximum entropy language model". In proceedings of the International conference on Spoken Language Processing. Pp.2137-2140, Jeju Island, Korea,2004.

[6] K. Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*,

[7] L. Rabiner and B.H Juang, Fundamentals of Speech Recognition, Prenticc Hall PTR Clffs, New Jersey.

[8] M. Zhang Huang, Y., and C.L. Tan. 2011. Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars. In Proceedings of ACL-HLT 2011: Short Papers, Portland, Oregon, pp.534-539.

[9] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," IEEE Trans. Speech Audio Processing, July, 2001.