한국음향학회지 제30권 제8호 pp. 428~435 (2011)

Removal of Heterogeneous Candidates Using Positional Accuracy Based on Levenshtein Distance on Isolated *n*-best Recognition

Young-Sun Yun
Department of Information and Communication Engineering, Hannam University
(Received August 24, 2011; revised October 12, 2011; accepted November 3, 2011)

Abstract

Many isolated word recognition systems may generate irrelevant words for recognition results because they use only acoustic information or small amount of language information. In this paper, I propose word similarity that is used for selecting (or removing) less common words from candidates by applying Levenshtein distance. Word similarity is obtained by using positional accuracy that reflects the frequency information along to character's alignment information. This paper also discusses various improving techniques of selection of disparate words. The methods include different loss values, phone accuracy based on confusion information, weights of candidates by ranking order and partial comparisons. Through experiments, I found that the proposed methods are effective for removing heterogeneous words without loss of performance.

Keywords: *Levenshtein Distance, Isolated Word Recognition, n-best Candidates Selection, Positional Accuracy*
Subject classification: *Speech Production and Perception (12.2)*

I. Introduction

Since voice is considered as the most intuitive and convenient communication method between human and machines, many researches are progressed over decades. Speech recognition systems are variously classified by size of vocabularies, utterance styles, and noisy environments. Most speech recognition systems are categorized as isolated or continuous. Isolated Word Recognition (IWR) requires a brief pause between each spoken word, whereas Continuous Speech Recognition (CSR) does not. In recent years, some mobile devices like iPhone or Android employ the large vocabulary CSR system over network (it means that an active Wi-Fi or 3G data connection is required), but there are still needs for IWR in offline (without data connection). For example, Google search and Dragon dictation application send user's recorded voice and receive transcribed text back from the recognition server (online mode). However, when users want to use speech recognition functions like command-and-controls on embedded device, voice dialing on the phone, destination input to car navigation system without data connection (offline mode), the devices require IWR function due to the limitation of size, cost, and network environments. Many IWR systems generate various results for a given utterance because they use only small amount of information. Since the recognition results are selected by acoustic scores, it is

Corresponding author: Young-Sun Yun (ysyun@hnu.kr)
Department of Information and Communication Engineering, Hannam University, 70, Hananm-ro, Daedeok-gu, Daejeon, 306-791, Korea

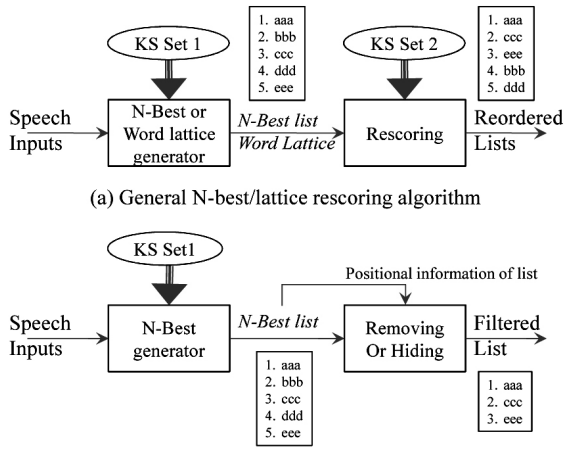

Fig. 1. The general *n*-best/lattice rescoring algorithm [1] and proposed list filtering method. KS stands for Knowledge Source.

possible that some unrelated words can be generated for candidates.

In this paper, I filter out the irrelevant ones from the candidate words by using their characters' distances or probabilities based on positional information. The difference between the proposed method and traditional rescoring algorithms [1−2] is that the selected words are removed or filtered out from the recognition results without adjusting their likelihoods. The general rescoring algorithms use the *n*−best list/lattice [1] to adjust the candidates' likelihoods based on more detailed acoustic and language information. However, I only use *n*−best results and their string information. Fig. 1 shows the difference between *n*−best rescoring algorithm and the proposed system. If the recognition results are filtered by the proposed method, users can have more confidence in new system than original system by obtaining relevant results.

The outline of the paper is as follows. I describe a Levenshtein distance and word similarities to measure correlations between candidates in Chapter II. Various approaches are discussed for applying the proposed similarity to improve accuracy of IWR systems in Chapter III. Chapter IV explains the experiments and their results. I end with a brief discussion and conclusions in Chapter V and VI.

II. Criteria of Word Correlations

In general, the recognition system generates *n* candidates for results. I consider word similarities among *n* candidates to find words that have relatively small correlation than others. To calculate the word similarities, I first use the Levenshtein distance and then get the corresponding characters between source and target strings. From the aligned characters, I calculate the positional probabilities according to the existence of the corresponding characters. If corresponding characters present on both strings, the position is called 'matched' though both characters are the same or not.

2.1. Levenshtein distance

The Levenshtein distance is a metric for measuring of difference between two sequences in sequence alignments and is widely used for information theory and machine translation [3−4]. Sequence alignment is a way of arranging the various sequences to find similarities or relationships between sequences. The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string to the other, with the allowable operations being insertion, deletion, or substitution of a character. This algorithm, an example of dynamic programming, reflecting loss values for edits or comparison errors is defined as follows.

$$D(i,j)=\min\begin{cases}D(i-1,j-1) & s_i=t_j(correct)\\ D(i-1,j-1)+\alpha & s_i\neq t_j(subst.)\\ D(i-1,j)+\beta & insertion\\ D(i,j-1)+\gamma & deletion\end{cases} \quad (1)$$

where s_i and t_j are the ith and jth characters of source and target strings, respectively and α, β, and γ mean the costs (loss values) needed for substitution, insertion and deletion edits, respectively. In the cases of correct and substitution, the characters are called as matched characters.

2.2. Word similarity

After the sequence alignment by Levenshtein distance is applied, each character is marked as inserted, substituted or correct. The deletion error is not considered because the character is not existed in target string. The correct and substitution attributes mean that a character in source string has a matched character in target string, but the insertion means that it has not.

To apply the Levenshtein algorithm to n-candidates, one is selected from the candidate list and is regarded as the source string. Then the others are considered as target strings. The similarity of the source is computed with one of target strings. The selections of source and target strings are repeatedly achieved by sequence for all combination.

In Table 1 and 2, for the same source string, the number of corresponding characters varies according to target strings. To solve this problem, I adopt null characters to equalize the number of comparing characters among different target strings like in Figure 2.

The null characters in source strings are compared to inserted characters in target strings. In some case, a null character is matched to multiple inserted characters. From the concept of null characters, the number of aligned positions (or corresponding characters) is same to all target strings for same source string. Word similarity is derived from two character's attributes: one is matched(correct or substituted) and the other is inserted.

1) Matched accuracy: Matched condition means that a character of target string has a corresponding character in source string. The corresponding characters are the same or not. If both characters are not same, the substitution error is occurred. In this case, accuracy is calculated by the frequency ratios of characters on the same position. For example, in Figure 2, the frequency ratio of the character 's' of source string is 1.0 for the first real character slot, but the ratios of two 'r's are calculated as 0.75 and 0.67, respectively. The first 'r' is found among three 'r's and one 'i'. On the other hand, the second 'r' is observed in two 'r's and one 'c'.
2) Inserted accuracy: Inserted character means that a character of target string has not a counterpart in source string. In this case, the frequency ratios are only calculated for target strings. The slots or positions of target string are corresponding to null characters (inserted positions) of source string and multiple characters of target strings can be exist in same slots by character alignments. In Figure 2, 'u' character of 'source' target is in the insertion slot and is observed in the slot including 'u', 'l', 'i', 't' and 'a'. Therefore, its frequency ratio is 1/5 = 0.2.

Table 1. Alignment of 'source' string to 'sorry' and corresponding characters' attributes (C: correct, I: insertion, S: substitution).

source	s	o	-	r	r	y
target	s	o	u	r	c	e
distance	0	0	1	1	2	3
attribute	C	C	I	C	S	S

Table 2. Alignment of 'sore' string to 'sorry' and corresponding characters' attributes.

source	s	o	r	r	y
target	s	o	r	-	e
distance	0	0	0	1	2
attribute	C	C	C	-	S

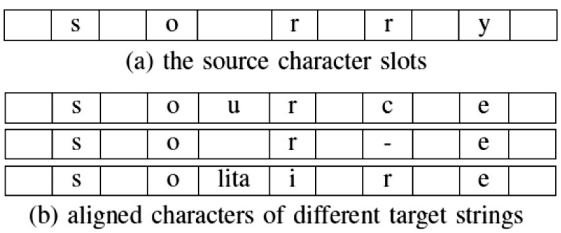

Fig. 2. Character slots for word comparison include null characters (insertion slots) (the string 'sorry' is used for source and the strings 'source', 'sore', and 'solitaire' are used for targets).

3) Word similarity: For the source string, the inserted accuracy is not considered and the target string has to reflect on both matched and inserted accuracy. By integrating both accuracies, word similarity for source i and target j is obtained as follows:

$$P_{acc}(i,j) = \left\{ \prod_{k=1,k\in matched}^{N_j} P_M(i,j,k) \right\}^{1/N_M^j} \cdot \left\{ \prod_{k=1,k\in inserted}^{N_j} P_I(i,j,k) \right\}^{1/N_I^j} \quad (2)$$

$$P_{acc}(j) = \left\{ \prod_{i=1,j\neq i}^{N_{cand}} P_{acc}(i,j) \right\}^{1/N_{cand}}$$

where P_M and P_I represent the matched and inserted accuracy, respectively. N_M^j and N_I^j mean the number of characters in matched and inserted characters of target string j comparing to ith source string, N_j is the number of characters of target string j and N_{cand} is the number of candidates (in general n-best recognition system, N_{cand} is n). In Figure 2, word similarity of 'sorry' to 'source' is $P_{acc}(0,1) = (1.0 \cdot 1.0 \cdot 0.75 \cdot 0.33 \cdot 0.75)^{1/5} \cdot (0.2)^{1/1} = 0.143$. In similar way, $P_{acc}(0,0) = 0.573$, $P_{acc}(0,2) = 0.866$, $P_{acc}(0,3) = 0.132$. From the example, the most dissimilar candidate to 'sorry' is decided as 'solitaire'.

III. Various Weighting Techniques

In the previous chapter, I define the word similarity for each candidate using positional accuracy on matched and inserted locations. I now consider various methods to improve performances to remove (or select) the irrelevant words from candidate list.

3.1. Basis

From the n-best candidate results, the word similarity for each candidate is calculated and it is excluded if the similarity is below threshold. Basis system means the proposed system adopting positional accuracy based on Levenshtein distance without any weighting techniques described in this chapter. On the other hand, the term of Dynamic Programming (DP) method is used to differentiate with the basis system. The DP method does not use the positional accuracy proposed in this paper. It only uses accumulated distance of edits cost between two strings.

3.2. Different loss value

In general DP approaches, the loss values for substitution, insertion and deletion errors are commonly used as 1:1:1, 4:3:3, 10:7:7 [5-6]. I investigate the effect of loss values by changing values for the insertion and deletion errors (α, β, γ in Eq. 1) in the proposed method. When the different loss values are applied to DP method, its overall performance is not significantly changed.

3.3. Phone accuracy

Phone accuracy is obtained from confusion matrix of recognition system output. If the phone accuracy is relatively higher than the other characters, it means that the substitution error is more critical than the insertion or deletion errors. Therefore, I increase the loss value α of substitution error to easily select the insertion or deletion error. Otherwise the phone accuracy is lower, I decrease the loss value α. This means that if the loss value is small, it prefers the substitution error to the insertion or deletion error.

3.4. Ranking order information

The most recognition system generates n words for outputs by recognition score such as likelihood. The top word means that it seems more reliable to be correct answer than the following candidates. To remove the irrelevant words from the candidate list, I cut off the word whose accuracy is below the given threshold by likelihood or score. From this consideration, I try to adopt the linear weights according to ranking order to word similarity Eq. 2 as follows

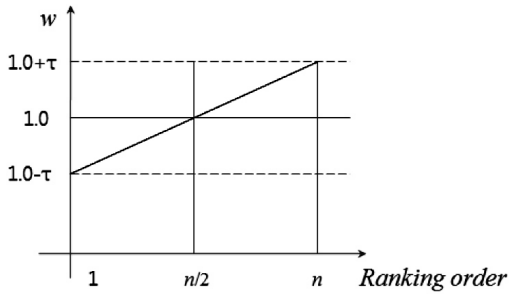

Fig. 3. Linear weight graph to adjust word similarities by ranking order.

(See Figure. 3).

$$w_i = \frac{2\tau}{n-1}(i-1) + (1-\tau), i = 1 \ldots n. \quad (3)$$

After determining the weights by ranking order, the weights are applied to distance or similarity score as follows:

$$\begin{cases} D(j)' = D(j) \cdot w_i & for distance \\ P_{acc}(j)' = P_{acc}(j)^{w_i} & for similarity \end{cases} \quad (4)$$

where $D(j)$ and $P_{acc}(j)$ are averaged distance and similarity score of jth candidate with ith ranking order, respectively.

3.5. Partial comparison

In the previous methods, all candidates act as source strings and the numbers of comparisons are all the same. Now, I try to use some of candidates, which have higher likelihoods, as source strings because they are closer to answers.

If the partial comparison method is adopted, each candidate acting as a source string, whose ranking is below m, has $n-1$ comparisons for all candidates, and candidates above m-ranking order have m comparisons as only target strings.

To reflect the number of comparisons, the distances are adjusted as

$$D_{i'} = \begin{cases} D_i/m & i \le m \\ D_i & otherwise \end{cases} \quad (5)$$

where D_i means the distance for ith ranking word. In Eq. 5, if m is too small, the distances of the precedence candidates are increased.

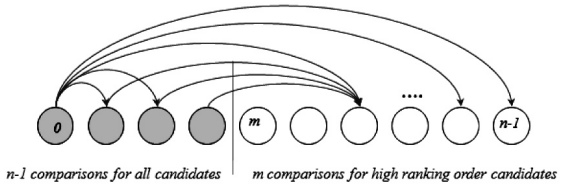

Fig. 4. The number of comparisons along to selected position (ranking order).

Therefore, I must determine the m-ranking to decrease their distances as

$$\frac{n-1}{m} < m \quad (6)$$

by considering the number of comparisons of precedence candidates and the denominator m. If n is 10, a possible value of m is 4 or greater.

IV. Experiments

To verify the proposed methods, I performed the selection of irrelevant words from results of Electronics and Telecommunications Research Institute (ETRI) recognition system. The ETRI system is developed for input systems of telematics POI (Point Of Interest) system with 260,000 words [7]. To reduce the computing time, the system adopts two-stage modeling consisting of monophone Semi-Continuous Hidden Markov Models (SCHMMs) and triphone SCHMMs. The system generates 10-best results and its performance is about 90 %. Before experiments, the speech researchers mark irrelevant words from recognition results on condition of with and without answers. Thereafter, the selected words by proposed methods are compared to the marked words. The data, which are used for recognition experiments, include In-Vocabulary (IV) 3,675 sentences and Out-Of-Vocabulary (OOV) 1,525 sentences for utterance verification purpose. I get 500 samples from IV sentences along to recognition performance ratio to find the performance changes. Table 3 shows the recognition performance for IV sentences and sampled 500 sentences and their counts.

Table 3. The recognition performance of baseline system and counts of words along to accuracy.

	Total	Top	10-best	Error
No. of sentences	3,675	2,975	3,299	376
	500	405	449	51
Accuracy	100 %	81.0 %	89.8 %	10.2 %

4.1. Evaluation Criteria

Depending on whether answers are given or not, the experts selected different words for irrelevant words. Thus, I first evaluate the recognition performances (the ratio that the remains include correct answer) for both cases. The results are shown in Table 4. From Table 4, if the experts know the reference (answer) for the given utterance, they did not select the references as irrelevant words. However, if they did not know the answer, many correct words are removed.

Precision is used for the metrics for evaluation instead of recall. In general, recall and precision depend on the outcome of a query and its relation to all relevant documents and the non-relevant documents. Precision can be seen as a measure of exactness or fidelity, whereas recall is a measure of completeness [8]. Our purpose is not that the proposed system finds all irrelevant words marked by the experts, but that the selected words are included in them. Thus, the precision is used for evaluations.

4.2. Experimental results

To compare the results of the proposed methods, I also performed the experiments of irrelevant word selection using typical DP. The DP method uses only the distance between source and target string without considerations of positional accuracy. The baseline system selects the word having the lowest distance as an irrelevant word in a candidate list, if 500 irrelevant words are required. If two irrelevant words are selected (words have the lowest and the 2nd lowest distance), then 1000-irrelevant words are obtained. The baseline1 and baseline2 systems are differed by the knowing

Table 4. Human evaluation with and without references (answers).

Reference	Accuracy		Total no. of irrelevant words
	Top	10-best	
Given	81.0 %	89.8 %	2,343
Not given	38.4 %	44.2 %	2,648

Table 5. Precision of the baseline systems for 500 irrelevant words (Baseline1 and precision1 are evaluated with reference words, baseline2 and precision2 are without them).

System	Top	10-best	Precision1	Precision2
Baseline1	81.0	89.8	82.6	-
Baseline2	38.4	44.2	-	83.8
DP	75.4	83.4	79.5	91.1
Proposed	75.8	84.2	76.9	86.4

Table 6. The precision of baseline system for 1,000 non-similar words.

System	Top	10-best	Precision1	Precision2
Baseline1	81.0	89.8	79.8	-
Baseline2	38.4	44.2	-	81.5
DP	69.4	76.8	78.0	88.8
Proposed	73.6	81.2	75.3	84.5

of the reference and the proposed system uses the positional information of Eq. 2. The baseline precisions are shown in Table 5.

From Table 5, I am able to consider that the good system approaches the accuracy of baseline1 system (81.0 % is the ideal performance of the proposed method in accuracy) and goes over the precision2 of baseline2 (83.8 %). The precision1 and precision2 are exclusively related to baseline1 and baseline2 according to the existence of answers, respectively. Thus, the combinations of baseline1 and precision2, of baseline2 and precision1 are not considered.

The proposed system's accuracy is over that of DP but its precisions are below those of DP. If the 1000-irrelevant words are selected, the accuracy of both DP and the proposed system decreased but the proposed system shows smaller performance loss than DP method in precision as in Table 6.

To improve the performance, I did several experiments presented in Chapter III. The

experimental results are sequentially shown in tables 7, 8, 9 and 10. 'Basis' system means the proposed system that does not apply any extra weighting techniques (exactly same to the proposed system in Table 5 and 6).

In Table 7 and 8, I changed α, β, γ to examine the effect of different loss values and confusion information. From Table 7, the performance slightly increased when the loss values 10:7:7 are used. The confusion information is effective for accuracy under the same condition, but for the precision it is not.

When the weights according to ranking order information are used, the performance for both DP and Basis system shows better results (Table 9).

However, on the partial comparisons, the DP system shows better performances than the Basis though the performance of both system increased (Table 10).

Table 7. Variation of precisions along to different loss value for 500-irrelevant words (†shows the value of $\alpha:\beta:\gamma$).

System	Top	10-best	Precision1	Precision2
Basis	75.8	84.2	76.9	86.4
10:3.5:3.5†	75.8	83.8	73.6	88.2
10:7:7†	76.4	84.6	77.4	87.0

Table 8. Variation of precisions with and without confusion information for 500-irrelevant words (†shows the system using phone accuracy based on confusion information).

System	Top	10-best	Precision1	Precision2
Basis	75.8	84.2	76.9	86.4
10:3.5:3.5	75.8	83.8	73.6	88.2
10:3.5:3.5†	75.2	83.6	73.6	87.2
10:7:7	76.4	84.6	77.4	87.0
10:7:7†	76.6	84.8	75.9	86.5

Table 9. Variation of precisions with ranking order weights for 500-irrelevant words (†shows the system using ranking order weights).

System	Top	10-best	Precision1	Precision2
DP	75.4	83.4	79.5	91.1
DP†	80.4	88.4	86.6	92.6
Basis	75.8	84.2	76.9	86.4
Basis†	80.0	88.2	85.7	92.1

Table 10. Variation of precisions based on partial comparisons for 500-irrelevant words (†indicates the partial comparison system).

System	Top	10-best	Precision1	Precision2
DP	75.4	83.4	79.5	91.1
DP†	81.0	88.8	90.0	91.0
Basis	75.8	84.2	76.9	86.4
Basis†	77.6	86.2	79.3	84.7

V. Discussion

From the experiments (Table 5 and 6), I found that the proposed filtering method shows better performance than the traditional DP methods without any improving techniques in top and 10-best accuracies. But, in precision metrics, the proposed method shows lower performance than the DP. This means that the proposed method keeps more answers and looses some irrelevant words than DP by using positional dependent information. The more irrelevant words, the bigger difference of performance between both systems.

If I change the loss values in case of substitution, insertion, and deletion, the larger loss values of insertions and deletions show better results than the equal loss values or small values of them (Table 7). The larger values of them have the effects that the insertion or deletion errors are relatively seldom occurred.

The phone accuracy based on confusion information of recognition results is always below than 1.0. Therefore, if the loss values of all errors are same, it does not affect the recognition accuracy and precision because the loss value for the substitution error is always less than or equal to those for insertion or deletion errors. From this observation, I apply the phone accuracy information to the case of different loss values along to error types. The results show that the performance is almost same in the condition of confusion information except precision metrics (Table 8).

However, the ranking information gives more effects in accuracy and precision criteria. This means that the precedence candidates (which

have lower ranking order) have greater influence. The similar results are shown in partial comparison methods (Table 9 and 10). The big difference between ranking order and partial comparison is the linear weights and stepwise weights. In partial comparison approach, the DP shows better results than the ranking order method.

In these experiments, the considered techniques in Chapter III played a positive role to improve the performance for both the proposed method and the traditional DP method. In particular, ranking order weights and partial comparison techniques took large performance gains in both systems.

VI. Conclusions

In this paper, I propose the removing or filtering method for less common candidates from the large vocabulary isolated recognition results in small IT or mobile devices. The proposed method assumes one of candidates as the reference (answer) and compares it to the remains using Levenshtein distance. After the comparisons, each word is aligned to the source and has the character positions to be marked matched or inserted. From this information, I calculated the positional accuracy along to matched or inserted position and filtered out the irrelevant words in *n*-best list. I also present the several methods to keep more accurate words and to select the disparate words more like experts' choices. From the experimental results, the considered methods show better accuracies and precisions than the pure (basis) method. The proposed methods can be also used in the machine translation or continuous speech recognition by considering positional word accuracies expanding character positions to word positions.

Acknowledgment

This research was supported by research fund of Hannam University in 2011.

References

1. X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a guide to theory, algorithm, and system development*, pp. 663-674, Prentice Hall, New Jersey, 2001.
2. J. Li, Y. Tsao and C.-H. Lee, "A study on knowledge source integration for candidate rescoring in automatic speech recognition," in *Proc. ICASSP*, pp. 837-840, Philadelphia, Pennsylvania, March 2005.
3. G. Leusch, N. Ueffing, and H. Ney, "A novel string-to-string distance measure with applications to machine translation evaluation," in *Proc. MT Summit IX*, pp. 240-247, New Orleans, Louisiana, September 2003.
4. L. Lita, "Dynamic machine translation evaluation methods: algorithmic analysis and generalization," CMU-LTI-05-193, 2005.
5. S. Young, J. Odell, D. Ollason, V. Valtchev and P. Woodland, *The HTK book version 2.1*, pp. 197, Cambridge University, 1997.
6. *NIST SCLITE Scoring Package Version 1.5*, http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm, 1997.
7. J. Park, H. Chung, and Y. Lee, "Development of the point-of-interest input system based on large-vocabulary embedded speech recognition," in *Proc. KSPSS*, pp. 108-111, November 2007.
8. I. Melamed, R. Green, and J. Turian, "Precision and recall of machine translation," in *Proc. HLT-NAACL,* pp.61-63, Edmonton, Canada, May 2003.

【Profile】

•Young-Sun Yun

Young-Sun Yun (M'01) received the B.S., the M.S. and the Ph. D. degrees in computer science in 1990, 1992, and 2001, respectively, from the Korea Advanced Institute of Science and Technology, Daejeon, Republic Of Korea. From 1992 to 1995, he worked on online character and gesture recognition at Handysoft Corp.. From 1995 to 2001, his research focused on the acoustic modeling using trajectory modeling. In 2001, he joined the Hannam University as Assistant Professor in the Department of Information and Communication Engineering. From Apr. 2006 to Feb. 2007, he was a visiting researcher of Electronics and Telecommunications Research Institute, Daejeon, Korea.
His current research interests include all issues related to speech recognition and understanding, especially acoustic modeling, speaker adaptation and utterance verification.