Re-ranking ASR Outputs for Spoken Sentence Retrieval

Yeongkil Song, Hyeokju Ahn, and Harksoo Kim

Program of Computer and Communications Engineering, College of IT,
Kangwon National University, Republic of Korea
{nlpyksong, zingiskan12, nlpdrkim}@kangwon.ac.kr

Abstract. In spoken information retrieval, users' spoken queries are converted into text queries by using ASR engines. If top-*1* results of the ASR engines are incorrect, the errors are propagated to information retrieval systems. If a document collection is a small set of short texts, the errors will more affect the performances of information retrieval systems. To improve the top-*1* accuracies of the ASR engines, we propose a post-processing model to rearrange top-*n* outputs of ASR engines by using Ranking SVM. To improve the re-ranking performances, the proposed model uses various features such as ASR ranking information, morphological information, and domain-specific lexical information. In the experiments, the proposed model showed the higher precision of 4.4% and the higher recall rate of 6.4% than the baseline model without any post-processing. Based on this experimental result, the proposed model showed that it can be used as a post-processor for improving the performance of a spoken information retrieval system if a document collection is a restricted amount of sentences.

Keywords: Re-ranking, ASR outputs, spoken sentence retrieval

1 Introduction

With the rapid evolution of smart phones, the needs of information retrieval based on spoken queries are increasing. Many information retrieval systems use automatic speech recognition (ASR) systems in order to convert users' spoken queries to text queries. In the process of query conversion, ASR systems often make recognition errors and these errors make irrelevant documents returned. If retrieval target documents (so called a document collection) are a small set of short texts such as frequently asked questions (FAQs) and restricted chatting sentences (*i.e.*, chatting corpus for implementing an intelligent personal assistant such as Siri, S-Voice, and Q-Voice), information retrieval systems will not perform well because a few keywords that are incorrectly recognized critically affect the ranking of documents, as shown in Fig. 1 [1].

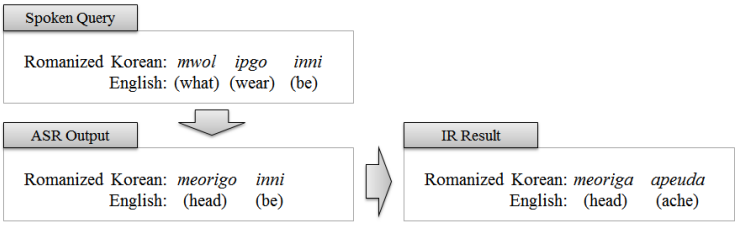

Fig. 1. Motivational example

To resolve this problem, many post processing methods for revising ASR errors have been proposed. Ringger and Allen [2] proposed a statistical model for detecting and correcting ASR error patterns. Brandow and Strzalkowski [3] proposed a rule based method to generate a set of correction rules from ASR results. Jung et al. [4] proposed a noisy channel model to detect error patterns in the ASR results. These previous models have a weak point that they need parallel corpus that includes ASR result texts and their correct transcriptions. To overcome this problem, Choi et al. [5] proposed a method of ASR engine independent error correction and showed the precision of about 72% in recognizing named entities in spoken sentences. Although the previous models showed reasonable performances, they have dealt with the first-ranked sentences among ASR results. The fact raised the result that low-ranked sentences are not considered although they are correct ASR outputs, as shown in the following Romanized Korean example.

Spoken query: *mwol ipgo inni* (What are you wearing?)
Rank 1: *meorigo inni* (Is a head?)
Rank 2: *mwol ipgo inni* (What are you wearing?)

To resolve this problem, we propose a machine learning model that re-ranks top-*n* outputs of an ASR system. In the above example, we expect that the proposed model changes *Rank 2* to *Rank 1*. If the volume of a document collection is big, it may be not easy to apply supervised machine learning models for re-ranking ASR outputs because the models need a large training data set that is annotated by human. However, if the document collection is a small set of short messages such as FAQs and chatting corpus, we think that the supervised machine learning models can be applied because the volume of the document collection is small enough to be annotated by human.

2 Re-ranking Model of ASR Outputs

2.1 Overview of the Proposed Model

The proposed model consists of two parts: a training part and a re-ranking part. Fig.1 shows the overall architecture of the proposed model.

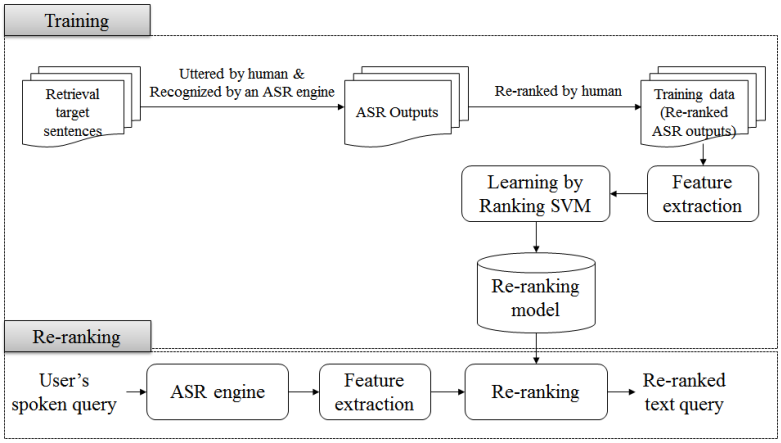

Fig. 2. Overall architecture of a re-ranking system

As shown in Fig. 1, we first collect top-*n* ASR[1] outputs of a document collection (a set of sentences in this paper) in which each sentence is uttered by 6 people. Then, we manually annotate the collected corpus with correct ranks. Next, the proposed system generates a training model based Ranking SVM (support vector machine) which is an application of SVM used for solving certain ranking problems [6]. When users input spoken queries, the proposed system re-ranks ASR outputs of the spoken queries based on the training model. Then, the system hands over the first ones among the re-ranked results to an information retrieval system.

2.2 Re-ranking ASR Outputs Using Ranking SVM

To rearrange top-*n* ASR outputs, we use a Ranking SVM which is a modification to the traditional SVM algorithm which allows it to rank instances instead of classifying them [7]. Given a small collection of ASR outputs ranked according to preference R^* with two ASR outputs $d_i, d_j \in R^*$, and a linear learning function f:

$$d_i \succ d_j \Rightarrow f(d_i) > f(d_j) \tag{1}$$

where the ASR outputs are represented as a set of features. The linear learning function f is defined as $f(d) = \mathbf{w} \cdot d$, as shown in Equation (2).

$$f(d_i) > f(d_j) \Leftrightarrow \mathbf{w} \cdot d_i > \mathbf{w} \cdot d_j \tag{2}$$

In Equation (2), the vector $\mathbf{w}$ can be learned by the standard SVM learning method using slack variables, as shown in Equation (3).

[1] We use Google's ASR engine which returns top-5 outputs per utterance.

$$\text{minimize } \langle \mathbf{w} \cdot \mathbf{w} \rangle + C \sum_{i,j \in |R|} \xi_{ij}$$

$$\text{subject to } \forall (d_i, d_j) \in R^* : \mathbf{w} \cdot d_i \geq \mathbf{w} \cdot d_j + 1 - \xi_{ij} \quad (3)$$

$$\forall (i, j) : \xi_{ij} \geq 0$$

To represent ASR outputs in the vector space of Ranking SVM, we should convert each ASR output into feature vectors. Table 1 show the defined feature set.

Table 1. Feature set of Ranking SVM

Feature Name	Explanation
ASR-Rank	Ranking of ASR outputs
ASR-Score	ASR score of the highest ranked ASR output
MOR-Bigram	Bigrams of morphemes
POS-Bigram	Bigrams of POS's
NUM-DUW	# of unknown content words that is not found in a domain dictionary
LEX-DUW	Unknown content words that is not found in a domain dictionary
NUM-GUW	# of unknown content words that is not found in a general dictionary
LEX-GUW	Unknown content words that is not found in a general in dictionary

In Table 1, *ASR-Rank* has an integer number from 1 to 5 because Google's ASR engine returns five ASR outputs ranked by descending order. *ASR-Score* is represented by 10-point scale of ASR scores 0.1 through 1.0. In other words, if the ASR score is 0.35, the score in 10-point scale is mapped into 0.4. *MOR-bigram* and *POS-Bigram* are morpheme bigrams and POS bigrams that are obtained from a result of morphological analysis. For example, if a result of morphological analysis is "I/prop can/aux understand/verb you/prop", MOR-bigram is the set { ^;I I;can can;understand understand;you you;$ }, and POS-bigram is the set { ^;prop prop;aux aux;verb verb;prop prop;$ }. In the example, '^' and '$' are the symbols that represent the beginning and the end of sentence, respectively. *NUM-DUW* and *LEX-DUW* are features associated with domain-specific lexicon knowledge. The domain dictionary used in *NUM-DUW* and *LEX-DUW* is a set of content words (so-called nouns and verbs) that is automatically extracted from a training data annotated with POS's by a morphological analyzer. *NUM-GUW* and *LEX-GUW* are features associated with general lexicon knowledge. The general dictionary used in *NUM-GUW* and *LEX-GUW* is a set of content words that is registered as entry words in a general purpose dictionary of a conventional morphological analyzer.

3 Experiments

3.1 Data Set and Experimental settings

We collected a chatting corpus which contains 1,000 sentences. Then, we asked six university students (three males and three females) for uttering the short sentences by

using a smartphone application that saves top-*5* outputs of Google's ASR engine. Next, we manually annotated with new rankings according to a lexical agreement rate between user's input utterance and each ASR output. In other words, the more an ASR output lexically coincides with user's input utterance, the higher the ASR output is ranked. Finally, we divided the annotated corpus into training data (800 sentences) and testing data (200 sentences). To evaluate the proposed model, we used precision at one (so-called P@1) and recall rate at one (so-called R@1) as performance measures, as shown in Equation (4). We performed 5-fold cross validation.

$$\begin{aligned} \text{P@1} &= \frac{\text{\# of sentences corectly ranked in top-1 by the proposed model}}{\text{\# of sentences ranked in top-1 by the proposed model}} \\ \text{R@1} &= \frac{\text{\# of sentences corectly ranked in top-1 by the proposed model}}{\text{\# of sentences correctly ranked in top-1 by an ASR engine}} \end{aligned} \tag{4}$$

3.2 Experimental Results

We computed the performances of the proposed model for each user, as shown in Table 2.

Table 2. Performances per user

User	ASR-only		Proposed Model	
	P@1	R@1	P@1	R@1
1	0.487	0.647	0.545	0.726
2	0.408	0.634	0.461	0.717
3	0.437	0.665	0.469	0.715
4	0.466	0.669	0.499	0.717
5	0.440	0.604	0.494	0.678
6	0.460	0.608	0.493	0.654
Average	0.450	0.638	0.494	0.702

In Table 2, ASR-only is a baseline model that returns a top-*1* output of an ASR engine without any re-ranking. The recall rate at five (so-called R@5) of Google's ASR engine was 0.705. This fact reveals that Google's ASR engine failed to correctly recognize 29.5% of the testing data. In other words, 29.5% of user's utterances are not included in top-*5* outputs of Google's ASR engine. As shown in Table 2, the proposed model showed the higher precision of 4.4% and the higher recall rate of 6.4% than the baseline model. This fact reveals that the proposed model can contribute to improve the performance of a spoken sentence retrieval system if a document collection is a small set of short texts.

4 Conclusion

We proposed a re-ranking model to improve the top-1 performance of an ASR engine. The proposed model rearranges ASR outputs based on Ranking SVM. To improve the re-ranking performances, the proposed model uses various features such as ASR ranking information, morphological information, and domain-specific lexical information. In the experiments with a restricted amount of sentences, the proposed model outperformed the baseline model (the higher precision of 4.4% and the higher recall rate of 6.4%). Based on this experimental result, the proposed model showed that it can be used as a post-processor for improving the performance of a spoken sentence retrieval system.

Acknowledgements

This work was supported by the IT R&D program of MOTIE/MSIP/KEIT. [10041678, The Original Technology Development of Interactive Intelligent Personal Assistant Software for the Information Service on multiple domains]. This research was also supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2013R1A1A4A01005074).

References

1. Kim, H., Seo, J.: Cluster-Based FAQ Retrieval Using Latent Term Weights. IEEE Intelligent Systems, 23(2), 58-65 (2008)
2. Ringger, E. K., Allen, J. F.: Error Correction via a Post-processor for Continuous Speech Recognition. In: Proceedings of IEEE International Conference on the Acoustics, Speech and Signal Processing, pp. 427-430 (1996)
3. Brandow, R. L., Strzalkowski, T.: Improving Speech Recognition through Text-Based Linguistic Post-processing. United States Patent 6064957 (2000)
4. Jeong, M., Jung, S., Lee, G. G.: Speech Recognition Error Correction Using Maximum Entropy Language Model. In: Proceedings of the International Speech Communication Association, pp.2137-2140 (2004)
5. Choi, J., Lee, D., Ryu, S., Lee, K., Lee, G. G.: Engine-Independent ASR Error Management for Dialog Systems. In: Proceedings of the 5th International Workshop on Spoken Dialog System (2014)
6. Joachims, T.: Optimizing Search Engines Using Clickthrough Data. In: Proceedings of ACM SIGKDD, pp. 133-142 (2002)
7. Arens, R. J.: Learning to Rank Documents with Support Vector Machines via Active Learning. Ph.D dissertation, University of Iowa (2009).