IMPROVED SPOKEN TERM DETECTION USING SUPPORT VECTOR MACHINES BASED ON LATTICE CONTEXT CONSISTENCY

Hung-yi Lee †1, Tsung-wei Tu #, Chia-ping Chen †, Chao-yu Huang #, Lin-shan Lee †#2

Graduate Institute of Communication Engineering, National Taiwan University †
Graduate Institute of Computer Science and Information Engineering, National Taiwan University #

tlkagkb93901106@yahoo.com.tw1, lslee@gate.sinica.edu.tw2

ABSTRACT

We propose an improved spoken term detection approach that uses support vector machines trained with lattice context consistency. The basic idea is that the same term usually have similar context, while quite different context usually implies the terms are different. Support vector machine can be trained using query context feature vectors obtained from the lattice to estimate better scores for ranking, and significant improvements can be obtained. This process can be performed iteratively and integrated with the pseudo relevance feedback in acoustic feature space proposed previously, both offering further improvements.

Index Terms— Spoken Term Detection, Query Context Consistency, Support Vector Machine

1. INTRODUCTION

Spoken term detection (STD) refers to the retrieval from a large spoken document archive of a list of spoken segments containing the term requested by the user. This technology is crucial to accessing multimedia content, including audio signals. In general, there are two stages in STD [1]. The audio content is first recognized and transformed into transcriptions or lattices using a set of acoustic and language models. The retrieval engine searches through the recognition results and then based on the entered query returns to the user a list of possible relevant spoken segments. The returned segments are usually ranked by the relevance scores derived from the recognition output. As a result, STD performance depends heavily on the acoustic and language models used in recognition. However, in practice the relatively poor performance of STD is due to the limited robustness of the available acoustic and language models, in particular with respect to the various topics represented in the audio content on the Internet, as well as the variety of speakers under different conditions in varying environments.

Recently it was proposed that by adjusting the relevance scores of the spoken segments using either pseudo relevance feedback (PRF) [2, 3, 4] or user relevance feedback [2, 5, 6], STD performance may be made less dependent on recognition results that have relatively high error rates. The acoustic models used in recognition can be adapted using user relevance feedback [5, 6]. Also, instead of relying solely on the recognition output, information obtained from the acoustic features such as the MFCCs can be used to improve retrieval performance [3]. These works notably take into account only acoustic-level information and ignore the linguistic context of the query.

A major problem in STD is the uncertainty in speech recognition, in particular the confusion among similarly pronounced words. The use of context information has been proposed to verify the presence of spoken terms [7]. Occurrences of a given term are usually characterized by similar context, while widely-varying contexts typically denote different terms. A previously mentioned example [7] is shown in Fig. 1 to illustrate this concept. For the user query “mouse”, spoken segment A with the phrase “the mouse trap” is relevant, whereas spoken segment B with the phrase “a house boat” is irrelevant. However, as the term “house” may be easily misrecognized as “mouse”, spoken segment B may be retrieved as a false alarm. If the system knows (for instance via language model constraints) that “mouse” is unlikely to have “boat” as its right context, as in the phrase “a mouse boat”, the corresponding path score in the lattice for “a mouse boat” will be lower, and with it the relevance score of segment B. Thus we use the query context, that is, the context of the query terms in the recognition results, to refine the relevance score of the spoken segments [7]. We obtain the query context directly from the recognition lattices and use this information with support vector machines (SVM) [8, 9, 10]. No external training corpora are needed. While it is true that language models to some extent already take into account query context, in real applications, it is difficult to obtain in-domain text data, that is, language model training data that matches the test domain. Also, lattice-based query context can yield information about recognition errors; language models trained on pure text, however, cannot properly model query context that is generated by such recognition errors.

2. PROPOSED APPROACH

Fig. 2 shows the framework of the proposed approach. In first-pass retrieval (Section 2.1), conventional STD technologies rank the spoken segments X based on the relevance scores derived from the

![Fig. 1: Example [7], showing that query context information has the potential to discriminate relevant and irrelevant spoken segments.](image-url)
recognition lattices with respect to query $Q$. On the left in Fig. 2 is shown the list of first-pass retrieval results. Every spoken segment in the list is represented by a feature vector representing the context of the hypothesized query terms found in the corresponding lattice; different context feature representations are proposed in Section 2.2. As described in Section 2.3, the top $N$ and bottom $N$ spoken segments are selected as the pseudo relevant or irrelevant spoken segments. These pseudo segments serve as the training set for the model to re-rank the first-pass retrieved segments. In Section 2.4, we propose two models, one based on cosine similarity and the other on SVM. We complete our description of the proposed framework in Section 2.5, which details segment re-ranking, and then describe in Section 2.6 an integration with the previously-proposed feature space pseudo-relevance feedback [3].

2.1. First-Pass Retrieval

The audio content to be retrieved is divided into spoken segments, each with a length of approximately one utterance, after which a recognition lattice is generated for each segment. The relevance score used to rank the first-pass retrieval results is denoted as $S_Q(X)$, and is defined as the expected occurrence count of query $Q$ within the lattice for segment $X$ [2, 5, 3]. A similar relevance score is widely used in other STD techniques [11, 12].

2.2. Query Context Feature Representations

The context feature representation for the segment $X$ lattice given query $Q$ is defined as the feature vector $f_Q(X)$. Fig. 3 shows different ways of defining this vector. Here, the word hypotheses (e.g., A, B, etc.) and posterior probabilities are shown beside each lattice arc. Green arcs are the immediate context of the query, that is, arcs adjacent to the query. Feature vector $f_Q(X)$ takes into account only the immediate context of the query. As each dimension corresponds to a lexical word, the vector dimensionality is the number of words in the lexicon. The value of each vector component is the posterior probability summed over all corresponding word arcs immediately connected to the query in the lattice. Vector $f_Q(X)$ is similar, except that all words appearing in the lattice are included, not only those adjacent to the query. Thus it contains the context information for the query throughout the entire segment. Vector $f_Q(X)$ separates the left and right immediate contexts of the query and hence has twice the dimensionality of $f_Q(X)$. Vector $f_Q(X)$ is the concatenation of $f_Q(X)$ and $f_Q(X)$ and is therefore three times the size of $f_Q(X)$.

2.3. Pseudo Relevant and Irrelevant Spoken Segments

As shown in Fig. 2, the top $N$ and bottom $N$ first-pass retrieved spoken segments are taken as the pseudo relevant ($X_Q^r$) and irrelevant ($X_Q^i$) spoken segments, respectively.

2.4. Context Consistency

We propose two ways to use the context feature vectors. In both cases, a context consistency score is obtained for each first-pass retrieved spoken segment for re-ranking.

2.4.1. Cosine similarity

In this approach, the context consistency score for each segment is

$$r_1(X) = \frac{1}{N} \sum_{X_Q^r} f_Q(X_Q^r) \cdot f_Q(X) - \frac{1}{N} \sum_{X_Q^i} f_Q(X_Q^i) \cdot f_Q(X)$$

(1)

where $N$ is the number of pseudo relevant and irrelevant segments and $f_Q(X)$ is the feature vector, which can be any of those described in Section 2.2. Eq. (1) is then the averaged cosine similarity between segment $X$ and all pseudo relevant segments minus the averaged cosine similarity between the segment and all pseudo irrelevant segments. That is, if the query context in segment $X$ is similar to that in the pseudo relevant segments, or if it dissimilar to that in the pseudo irrelevant segments, the segment is most likely relevant. The value of $r_1(X)$ is then linearly normalized between 0 and 1 as $r_1(X)$.

2.4.2. Support vector machine

For each query, the pseudo relevant and irrelevant segments are used to train a support vector machine (SVM) to classify all of the segments as relevant and irrelevant based on the query context feature vectors. Because the distance of each feature vector to the SVM hyperplane $d(X)$ represents the confidence of the classification result, the context consistency score for this approach is

$$r_2(X) = \begin{cases} d(X) & X \text{ is classified as relevant} \\ -d(X) & \text{otherwise} \end{cases}$$
A negative sign is needed here because $d_l(X)$ as in Fig. 2 is always positive. The value of $r_2^f(X)$ is then linearly normalized between 0 and 1 as $r_2(X)$.

### 2.5. Re-ranking and Iterations

The context consistency score is then integrated with the original relevance score $S_Q(X)$ mentioned in Section 2.1 for re-ranking:

$$\hat{S}_Q(X) = S_Q(X) r_1(X)^\alpha \text{ or } S_Q(X) r_2(X)^\alpha,$$

(2)

where $\alpha$ is a constant.

The above process can be conducted iteratively. That is, the retrieval results re-ranked by $\hat{S}_Q(X)$ in Eq. (2) can be taken as the first-pass retrieval results, and the above process can be repeated over the re-ranked results again.

### 2.6. Integration with Pseudo Relevance Feedback in Acoustic Feature Space

The retrieval results re-ranked as proposed above can be further taken as the first-pass results for pseudo relevance feedback (PRF) in the acoustic feature space [3]. The top $M$ spoken segments are selected as the pseudo relevant spoken segment set$^1$, and the system calculates the dynamic time warping distances of each segment and the $M$ pseudo relevant segments over the hypothesized query occurrence region based on their corresponding MFCC sequences. The distances are then transformed into similarity measures to be used to further re-rank the segments.

### 3. EXPERIMENTS

#### 3.1. Experimental Setup

We used as the test data 33 hours of recorded lectures of a course offered at National Taiwan University. The acoustic model was trained using the maximum likelihood criterion with 4602 state-tied triphones spanned from 37 monophones using a corpus of noiseless Mandarin read speech, including 24.6 hours of data produced by 100 males and 100 females. 39-dimension MFCCs were used as the feature. There were five states per triphone and 24 mixtures per state. A lexicon with 10.7K words was used, and a trigram language model was trained on a news corpus. The character accuracy was 50.26%. We used mean average precision (MAP) as the retrieval evaluation measure, for which we manually selected 80 single-word testing queries. For SVM we used the tool SVM$^\text{light}$ with the default parameters. The baseline (first-pass retrieval) MAP was 0.4819.

#### 3.2. Experimental Results

##### 3.2.1. Different context consistency scores

First, we compared the context consistency scores derived using the two approaches proposed in Section 2.4. $\alpha$ in Eq. (2) was set to either 1 or 10. The top or bottom $N$ spoken segments in the first-pass retrieval results were taken as the pseudo relevant and irrelevant spoken segments, and $N$ was set to 5, 10, 15, 20, 25, or 30. Note that this means that it was possible for a spoken segment to belong to both the pseudo relevant and irrelevant sets in this experiment. We used the $f_2^f(X)$ context feature vector. The experimental results in Table 1 show that the context consistency scores derived using SVM outperformed those based on cosine similarity. The SVM context consistency scores $r_2(X)$ improved the retrieval performance regardless of the choice of $\alpha$. For $\alpha = 10$ and $N = 20$, the relative MAP improvement using SVM was 14.61%. In the following experiments, we used SVM context consistency scores and set $\alpha$ equal to 10.

##### 3.2.2. Different query context feature representations

In this section, we compared the different query context feature vector representations described in Section 2.2. The experimental results in Table 2 show that $f_2^f(X)$, which takes into account the context of the query throughout the entire segment, outperforms $f_1^f(X)$ (considers immediate query context only). It is not surprising that $f_0^f(X)$ (both left and right immediate query context) also outperformed $f_1^f(X)$. $f_0^f(X)$, the concatenation of $f_1^f(X)$ and $f_2^f(X)$, slightly outperformed $f_2^f(X)$ in all cases and outperformed $f_1^f(X)$ except for $N = 5$ and $N = 10$. $N = 20$ yielded the best results for all types of feature vectors. The maximum relative improvement (18.22%) was obtained using $f_2^f(X)$ for $N = 20$.

##### 3.2.3. Iterations

In this experiment, we used the $f_2^f(X)$ feature vector and set $N$ equal to 20 to re-rank the first-pass results iteratively using SVM-based context consistency. The results are shown in Fig. 4, for which the vertical axis is MAP, and the horizontal axis is the number of

---

$^1$The size of the pseudo relevant set for context consistency ($N$) and PRF in the acoustic feature space ($M$) need not be the same.

---

### Table 1: Comparison between different context consistency models using $f_2^f(X)$ with different values of $\alpha$ in Eq. (2). Max RI represents maximum relative improvement.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Cosine similarity</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baseline</td>
<td>0.4819</td>
</tr>
<tr>
<td>10</td>
<td>0.4647</td>
<td>0.4359</td>
</tr>
<tr>
<td>5</td>
<td>0.4830</td>
<td>0.4541</td>
</tr>
<tr>
<td>10</td>
<td>0.4835</td>
<td>0.4590</td>
</tr>
<tr>
<td>15</td>
<td>0.4825</td>
<td>0.4583</td>
</tr>
<tr>
<td>20</td>
<td>0.4821</td>
<td>0.4582</td>
</tr>
<tr>
<td>30</td>
<td>0.4796</td>
<td>0.4571</td>
</tr>
<tr>
<td>Max RI(%)</td>
<td>0.33</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 2: Comparison of different query contextual feature representations with SVM and $\alpha = 10$. Max RI represents maximum relative improvement.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$f_1^f$</th>
<th>$f_2^f$</th>
<th>$f_3^f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.4819</td>
<td>0.4819</td>
<td>0.4819</td>
</tr>
<tr>
<td>N=5</td>
<td>0.5185</td>
<td>0.5196</td>
<td>0.5528</td>
</tr>
<tr>
<td>N=10</td>
<td>0.5437</td>
<td>0.5503</td>
<td>0.5552</td>
</tr>
<tr>
<td>N=15</td>
<td>0.5506</td>
<td>0.5663</td>
<td>0.5653</td>
</tr>
<tr>
<td>N=20</td>
<td>0.5523</td>
<td>0.5678</td>
<td>0.5662</td>
</tr>
<tr>
<td>N=25</td>
<td>0.5516</td>
<td>0.5667</td>
<td>0.5634</td>
</tr>
<tr>
<td>N=30</td>
<td>0.5514</td>
<td>0.5676</td>
<td>0.5528</td>
</tr>
<tr>
<td>Max RI(%)</td>
<td>14.61</td>
<td>17.83</td>
<td>17.49</td>
</tr>
</tbody>
</table>

---

1http://www.cs.cornell.edu/People/tj/svm_light/
iterations. Fifteen iterations were conducted, of which iteration 0 corresponds to the baseline first-pass results. With iteration, MAP improved from 0.5697 (iteration 1) to 0.6055 (iteration 12).

3.2.4. Integration with PRF in acoustic feature space

In this experiment, the iteration 12 results from above were taken as the first-pass retrieval results for PRF in acoustic feature space. The results are shown in Fig. 5, for which the vertical axis is MAP, and the horizontal axis is the number of pseudo relevant spoken segments $M$ (from Section 2.6), which range from 1 to 10. $M = 0$ corresponds to iteration 12 results from above (MAP 0.6055). While all values of $M$ yielded improved retrieval performance, the best MAP (0.6204) was achieved at $M = 4$. This experiment shows that the proposed approach complements PRF, presumably because they take into account different information: context for the former but acoustic features for the latter.

4. CONCLUDING REMARKS

We calculated the query context consistency within the lattices of the spoken segments and used the resultant pseudo relevant and irrelevant segments to train SVMs, which we used to estimate better scores for STD ranking. We used various query context feature vectors to yield significant improvements. The process can be performed iteratively, and can be integrated with PRF in acoustic feature space, both of which yield further improvements.

5. REFERENCES