Enhanced Spoken Term Detection Using Support Vector Machines and Weighted Pseudo Examples

Hung-yi Lee and Lin-shan Lee, *Fellow, IEEE*

***Abstract*—Spoken term detection (STD) is a key technology for retrieval of spoken content, which will be very important to retrieve and browse multimedia content over the Internet. The discriminative capability of machine learning methods has recently been used to facilitate STD. This paper presents a new approach to improve STD using support vector machines (SVM) based on acoustic information. The concept of pseudo-relevance feedback (PRF) well used in the retrieval of text, image and video is used here. The basic idea of using PRF here is to assume some spoken segments in the first-pass retrieved results are relevant (or pseudo-relevant) and some others irrelevant (or pseudo-irrelevant), and take these segments as positive and negative examples to train a query-specific SVM. This SVM is then used for re-ranking the first-pass retrieved results, and only the re-ranked results are shown to the user. In this paper, feature vectors representing the spoken segments based on acoustic information to be used in SVM are considered and analyzed. Furthermore, conventionally in PRF the items with the highest and lowest scores in the first-pass retrieved results are respectively taken as pseudo-relevant and -irrelevant, but in this way some incorrect examples are inevitably included in the training data especially when the recognition accuracy is poor. Here we further propose an enhanced SVM which not only better selects positive/negative examples considering the reliability of the spoken segments, but emphasizes more on more reliable training examples by modifying the SVM formulation. Experiments on two different sets of spoken archives with different speaking styles and different levels of recognition accuracies demonstrated significant improvements offered by the proposed approaches.**

***Index Terms*—Pseudo-relevance feedback, spoken term detection.**

I. Introduction

In the Internet era, digital content over the Internet covers almost all the information and activities of human life. The most attractive form of network content is multimedia including audio signals. The subjects, topics, and core concepts of such multimedia content can very often be identified based on the speech information within the audio part of the content. Hence spoken content retrieval will be very important in helping users retrieve and browse efficiently across the huge quantities of multimedia content in the future [1]. In general, there are two stages in most conventional spoken content retrieval approaches [2]. In the first stage, the audio content is recognized and transformed into transcriptions or lattices by a recognition engine based on a set of acoustic models and language models. In the second stage, after the user enters a query, the retrieval engine searches through the recognition output and returns a list of relevant spoken segments to the user. The returned segments are usually ranked by the relevance scores derived from the recognition output. This paper is focused on a subtask of the above spoken content retrieval, spoken term detection (STD), in which the query is a term in text form and a spoken segment is taken as relevant if it includes the query term. However, it is certainly possible to generalize the discussions here to other tasks in spoken content retrieval.

Substantial research effort has been made in STD, and many successful techniques have been developed. Lattice-based approaches taking into account multiple recognition hypotheses [3], [4] were used to take care of the problem of relatively low accuracy in 1-best transcriptions. Lattices are usually converted into sausage-like structures to make the indexing task easier and save the memory space. Good examples of such sausage-like lattice-based structures include Position Specific Posterior Lattices (PSPL) [5], [6], Confusion Networks (CN) [6], [7], etc. Weighted finite state transducer (WFST) algorithm also provides another effective way for indexing and retrieving lattices [8]. The out-of-vocabulary (OOV) query is another important issue because typically many queries contain OOV terms [9]. The most fundamental approach for handling the OOV problem is to represent both the queries and the spoken segments by properly chosen subword units and then try to match them on the subword unit level [10]–[20]. Word-based and subword-based indexing can be further properly integrated to yield better performance [11], [21], [22]. Many successful applications have been demonstrated with good examples including those browsing over broadcast news [23], course lectures [24], [25], podcasts [26], YouTube videos [27], etc.

There have been some previous works [28]–[31] taking advantage of the discriminative capabilities of machine learning methods such as support vector machines (SVMs) or multi-layer perceptrons (MLPs) to facilitate STD. In those previous works, a set of training queries and associated relevant/irrelevant segments are assumed available. In such cases, SVMs or MLPs can be trained to classify whether a spoken segment contains the entered query term or not. In order to have these machine learning classifiers properly work for the target spoken archive, the training data must be reasonably matched to the target spoken archive, but such data is usually not available or difficult to collect. In fact, it is very possible

Manuscript received August 16, 2012; revised December 30, 2012 and January 27, 2013; accepted February 03, 2013. Date of publication February 25, 2013; date of current version March 13, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yang Liu.

H. Lee is with the Research Center for Information Technology Innovation, Academia Sinica (e-mail: tlkagkb93901106@gmail.com; tlkagkb93901106@yahoo.com.tw).

L. Lee is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: lslee@gate.sinica.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2013.2248721

that the spoken archive includes content produced in different parts of the world by different speakers on different domains under different acoustic conditions. This makes collecting a reasonably good training set very difficult. Moreover, since the characteristics of the queries are usually very diverse, a single classifier optimized for many different training queries may not be able to offer the best solution for the high variety of many different testing queries. This seems to imply a goal to train a specific classifier for each query. However, this goal is even more difficult to realize because it presumes that the training data good for every specific possible query covering proper diversity of acoustic and linguistic conditions are available.

On the other hand, pseudo-relevance feedback (PRF), also known as blind relevance feedback, has been widely used in information retrieval to obtain relevance information for each query without actually involving any action from the user. It has been successfully applied on different retrieval domains like text [32]–[36], image [37] and video [38], [39]. Conventionally, in PRF, a first-pass retrieval is performed first, and it is assumed that a small number of top-ranked objects in the first-pass retrieved results are relevant (or "pseudo-relevant"), and sometimes in addition some bottom-ranked objects are irrelevant (or "pseudo-irrelevant"), and these pseudo-relevant (and -irrelevant) objects can then be taken as extra information to improve the retrieval results including used as the training data for various machine learning approaches. In this way, a set of training data for each specific query can be obtained, and a query-specific classifier or model can be learned. Although the training data thus obtained does not necessarily cover the proper diversity of acoustic and linguistic information as desired, it should be reasonably matched to the target spoken archive considered. Techniques of using machine learning methods in PRF scenario have been extensively developed for video retrieval [40] and image retrieval [41], although not yet properly explored for STD.

This paper reports a new approach to improve STD by SVM with training data generated by PRF, so a query-specific model can be trained with a query-specific training data set, which is optimized for the given query to a certain extent. In this approach when a query term is entered, the system first generates first-pass retrieved results ranked according to the relevance scores derived from the lattices. This ranked list is not shown to the user. The system then selects some spoken segments from this ranked list as *pseudo-relevant segments* (or *positive examples*) and *pseudo-irrelevant segments* (or *negative examples*) based on some criteria. These positive and negative examples are then used to train a query-specific model based on SVM. Then all segments on the first-pass retrieved list are re-ranked using this query-specific SVM obtained with query-specific training data. The system finally displays the re-ranked results to the user.

To train the above SVM model, each spoken segment in the first-pass retrieved list should be represented as a feature vector containing sufficient information for evaluating its relevance with respect to the query term. Approaches of using acoustic information for constructing such feature vectors are proposed here in this paper. This is inspired from the previous works [42]–[45], in which the concept all segments including the query terms should include some parts of the signal exhibiting similar characteristics in acoustic features was verified to be useful. In these previous works, the STD systems simply increased the relevance scores of the spoken segments similar to those with higher scores, and the similarities were evaluated based on the dynamic time warping distances between the acoustic feature sequences of the signal regions hypothesized to be the query terms. In this paper, we realize the above concept using completely different methods of machine learning. The top- and bottom-ranked spoken segments in the first-pass retrieved results can be respectively taken as positive and negative examples, and this approach has been shown to be very helpful [46]. However, in this way some incorrect examples are inevitably included in the training data especially when the recognition accuracy is poor. In this paper, to further enhance the scenario of PRF, we propose an approach to select better positive and negative example sets not restricted to top- and bottom-ranked segments from the first-pass results, and evaluate a presumable reliability for each example. We further modify the SVM formulation such that less reliable examples have less effect upon the SVM model learned.

In comparison with the previous works which utilize a set of training queries for learning a discriminative model to facilitate STD [28]–[31], the proposed approach has several benefits. First of all, the proposed approach does not really need any training data because the training examples for each query are automatically generated. Second, since in the proposed approach the retrieval result of the query entered is re-ranked by a query-specific SVM model, the proposed approach may outperform those previous methods which learn a general model from a training query set. However, those previous methods can be integrated with the approach proposed here. With a set of training queries, the system is able to learn a model generating better first-pass retrieved results, which yield more accurate positive/negative examples for SVM training in PRF.

Below, the framework for improved STD using SVM with PRF is first presented in Section II. The feature vectors used to represent each spoken segment are introduced in Section III. In Section IV, we describe the enhanced PRF which includes a better example selection strategy in Section IV-A and modified SVM in Section IV-B. Sections V and VI then report the experimental results. The concluding remarks are finally made in Section VII.

II. Basic Framework for STD Using SVM With PRF

Fig. 1 shows the basic framework for the proposed approach. In the first-pass retrieval process as described in Section II-A below, conventional STD technologies are used to rank the spoken segments based on the relevance scores derived from the lattices with respect to the entered query term Q. On the left lower part of the figure is the list of first-pass retrieved results, which is not shown to the user. As will be presented in Section II-B below, some spoken segments in the first-pass retrieved list are selected as the pseudo-relevant and -irrelevant spoken segments, or positive and negative examples, serving as the training data for the query-specific SVM model, which will be used for determining the relevance between all segments in the first-pass retrieved results and the query term Q

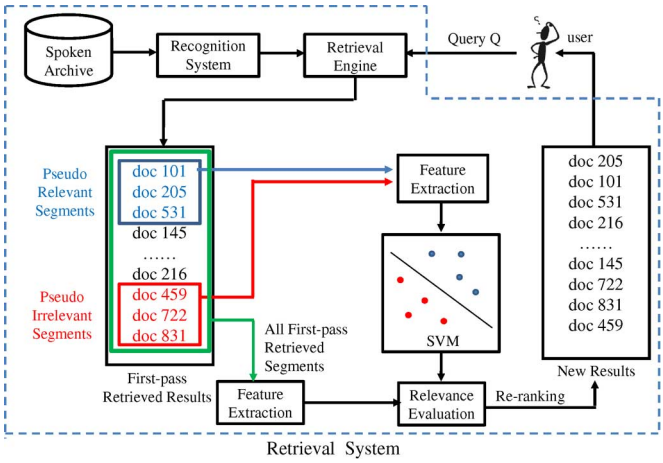

Fig. 1. The framework for spoken term detection (STD) using support vector machines (SVM) with pseudo-relevance feedback.

for re-ranking. In Section II-C, based on the relevance of the spoken segments derived from the SVM model, the segments are finally re-ranked. This re-ranked list is then shown to the user.

A. First-Pass Retrieval

The whole spoken archive to be retrieved is first divided into spoken segments, each corresponding to approximately an utterance. Each spoken segment x in the spoken archive is then transcribed into a lattice $L(x)$. When a query term Q, which can be either a word or a phrase in text form, is entered, the retrieval engine searches through the lattices $L(x)$ for all segments x in the spoken archive and returns a list of spoken segments possibly containing the query term. Here all spoken segments x retrieved are ranked by their degree of relevance with respect to the query Q, represented by a relevance score $S_Q(x)$,

$$S_Q(x) = \frac{\sum_{u \in L(x)} P(x|u)P(u)C(u,Q)}{\sum_{u \in L(x)} P(x|u)P(u)}, \tag{1}$$

where u is an allowed word sequence in the lattice $L(x)$, $P(x|u)$ is the likelihood for the observation sequence for the segment x given the word sequence u based on the acoustic model set, $P(u)$ is the prior probability of u from the language model, and $C(u,Q)$ is the occurrence count of the query Q in u. Since the denominator in (1) is the sum of the likelihoods of all word sequences u in the lattice, while the numerator of (1) is the same but with all word sequences weighted by the occurrence count of the query Q, (1) can be interpreted as the expected occurrence count of the query Q based on the lattice $L(x)$. Such a relevance score in (1) is standard and widely used in STD [5], [31], [47]–[50].

B. SVM Model Training (Plain SVM)

As shown in Fig. 1, some spoken segments in the first-pass retrieved list are respectively taken as pseudo-relevant and -irrelevant spoken segments, and they are considered as positive and negative examples to train an SVM model. To train this SVM model, each spoken segment x should be represented by a feature vector $f(x)$ as will be presented further in Section III. A simple but effective way for the above positive and negative example selection is to respectively take the top and bottom N segments on the first-pass returned list as positive and negative examples. More sophisticated approach for example selection will be described later in Section IV.

Suppose that a positive example set, $\mathcal{X}_T = \{x_1^t, \ldots, x_i^t, \ldots, x_{n_t}^t\}$, and a negative example set, $\mathcal{X}_F = \{x_1^f, \ldots, x_j^f, \ldots, x_{n_f}^f\}$, are obtained from the first-pass results, where n_t and n_f are respectively the numbers of positive and negative examples.[1] An SVM model characterized by a weight vector w can be learned to measure the relevance of each segment with respect to the query based on these examples. The SVM model w is learned by solving the following optimization problem [51]:

$$\min_{w,\epsilon_i^t,\epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}_T} \epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}_F} \epsilon_j^f, \tag{2}$$

such that

$$\forall x_i^t \in \mathcal{X}_T, \quad w \cdot f\left(x_i^t\right) \geq 1 - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$
$$\forall x_j^f \in \mathcal{X}_F, \quad w \cdot f\left(x_j^f\right) \leq -1 + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

The constraints above require that the inner products of w and the feature vectors $f(x_i^t)$ of all positive examples x_i^t should be larger than one, while the inner products of w and the feature vectors $f(x_j^f)$ of all negative examples x_j^f smaller than negative one. Each constraint is padded with a per-example slack variable (ϵ_i^t for positive example x_i^t and ϵ_j^f for negative example x_j^f). The sum of the slack variables over the training examples is minimized to reduce the degree of constraint violations to the smallest extent. The norm of the vector w to be learned and the scale of the slack variables for positive and negative examples are respectively traded off with the parameters γ and γ'. Based on (2), $w \cdot f(x)$ tend to be larger for positive examples (or pseudo-relevant segments), and smaller for negative examples (or pseudo-irrelevant segments), so $w \cdot f(x)$ for each segment x in the first-pass retrieved results can be used for measuring the confidence to be relevant with respect to the query term. This SVM model formulated in (2) will be referred to as Plain SVM below, as compared to the enhanced version referred to as Enhanced SVM to be presented later.

C. Re-Ranking

SVM-derived confidence score $R(x)$ is then obtained by linearly normalizing the inner product $w \cdot f(x)$ into a real number between 0 and 1:

$$R(x) = \frac{w \cdot f(x) - d_{min}}{d_{max} - d_{min}}, \tag{3}$$

where d_{max} and d_{min} are respectively the maximum and minimum values of the inner products $w \cdot f(x)$ among all the segments x in the first-pass retrieved list. The new relevance score $\hat{S}_Q(x)$ for re-ranking the segment x is then obtained by integrating the original relevance score $S_Q(x)$ in (1) with the confidence score $R(x)$ in (3) as

$$\hat{S}_Q(x) = S_Q(x)R(x)^\delta, \tag{4}$$

[1]If top and bottom N segments in the first-pass results are selected as the examples, then $n_t = n_f = N$.

where δ is a weight parameter. A new ranking list is thus generated based on the new relevance scores in (4) to be shown to the user.

III. Feature Vector Representations Based on Acoustic Information

In order to train an SVM model for each query term as mentioned above, each spoken segment needs to be represented by a feature vector. The basic idea here is that the MFCC vector sequences representing different occurrences of the same term should be similar in some way; while very different MFCC vector sequences very possibly imply different terms. It is therefore possible to discriminate relevant and irrelevant spoken segments by comparing the MFCC vector sequences with the pseudo-relevant and -irrelevant segments based on the signal parts hypothesized as the query. In this section, we show the method representing the MFCC vector sequences as a feature vector.

Here we first define the "hypothesized region" for a spoken segment x with respect to a query Q to be the part of the MFCC vector sequence for the segment corresponding to a word arc in the lattice whose word hypothesis is exactly the query term Q with the highest posterior probability, as shown in Fig. 2(a) at the upper left corner of the figure. Note that the hypothesized region is a sequence of MFCC vectors with variable length, but for SVM model training and testing, it is more convenient to represent different spoken segments by feature vectors with fixed dimensionality. Fig. 2(b), (c) and (d) illustrate three different ways to accomplish this goal as follows.

- **Term-based Average**: All MFCC vectors in the hypothesized region for the query term are averaged into a single feature vector, so the dimensionality of the feature vector is the same as that of each MFCC vector. The value of each component of this feature vector is the average of all the corresponding components of the MFCC vectors in the hypothesized region. This is denoted by $f_1(x)$ and is shown in Fig. 2(b) at the upper right corner of the figure.
- **Phone-based Average and Concatenation**: The hypothesized region is divided into a sequence of phone segments based on the phone boundaries obtained during the lattice construction. Each phone segment is then represented by the average of the MFCC vectors in the phone segment. The concatenation of these averaged MFCC vectors representing the phone segments then gives the feature vector for a spoken segment. Thus for a query term including m phones the dimensionality of the feature vector is m times of the dimensionality of a single MFCC vector. This is denoted by $f_2(x)$ and shown in Fig. 2(c) at the lower left corner of the figure.
- **State-based Average and Concatenation**: Each phone segment is further divided into a sequence of state segments according to the HMM state boundaries obtained during the recognition, each of which is again represented by the average of the MFCC vectors. All these averaged vectors for HMM states in a hypothesized region are then concatenated as a feature vector. Thus for l-state phone HMMs the dimensionality of such a feature vector is l times of the dimensionality of $f_2(x)$. This is denoted as $f_3(x)$ and is shown in Fig. 2(d) at the lower right corner of the figure.

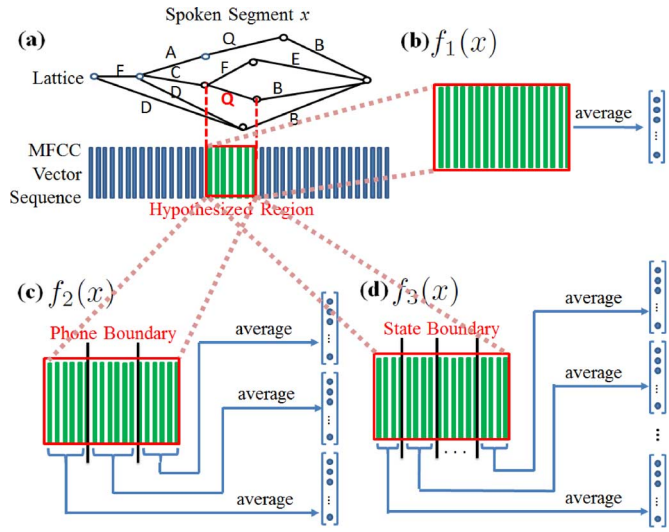

Fig. 2. Different forms of feature vector representations. (a): the definition of a "hypothesized region" in the lattice of segment x for the query term Q, where A, B, . . . , E are some other word hypotheses different from Q. (b), (c) and (d): the feature vectors $f_1(x)$, $f_2(x)$ and $f_3(x)$ respectively.

Although in the above we only mention MFCC vectors, and in the experiments below only results using MFCC vectors are reported, certainly many other representations for acoustic information of speech can be used as well. A good example may be the Gaussian posteriorgrams [52], [53] which may take better care of the speaker variability issue since the target spoken segments may be produced by many different speakers.

IV. Enhanced SVM

In conventional PRF scenario, the top/bottom N segments in the first-pass results are taken as positive/negative examples. However, in this way, it is unavoidable to include some incorrect examples (irrelevant segments taken as positive examples, and vice versa) in the training data especially when the recognition accuracy is relatively poor. To better handle this problem, in Section IV-A below, we propose that better sets of positive/negative examples not restricted to top and bottom N segments can be carefully selected, and the reliability for each selected example can be further estimated. In Section IV-B below, we further propose that the formulation of SVM can be modified to learn primarily from the presumably correct examples during training, while reduce the influence of those unreliable examples on the model learned.

A. Training Example Selection and Reliability Estimation

Because the same terms may exhibit similar acoustic characteristics, relevant segments may show certain degree of similarity to each other in terms of their feature vectors $f_1(x)$, $f_2(x)$ and $f_3(x)$ as defined in Section III, and the feature vectors for relevant and irrelevant segments may be far apart. Because the top/bottom N segments in the first-pass results usually have higher probabilities to be relevant/irrelevant, the relevance of each segment can be estimated to some extent based on the similarity of its feature vector with those of the top- and

bottom-ranked segments. Based on this concept, we can obtain positive/negative example sets not restricted to top and bottom segments. For each segment in the first-pass retrieved list, its similarity with respect to the top and bottom N segments is first computed based the distances between their feature vectors. If a spoken segment is similar to more top N segments than bottom N segments, it may be taken as a positive example, and the difference between its similarity to top and bottom N segments can be considered to be proportional to the reliability of this example. On the contrary, a spoken segment similar to more bottom N segments may be taken as a negative example, and its reliability can be estimated similarly. Such a concept can be realized by the procedure below.

The following procedure is derived to obtain a set of positive examples $\mathcal{X}_T$ and a set of negative examples $\mathcal{X}_F$ in which each example x has a value $C(x)$ proportional to its reliability.

(i) Each segment x in the first-pass result is first assigned an initial score $w_0(x)$, which is 1 for top N segments, -1 for bottom N segments, and 0 for the others.

(ii) Compute the similarity $s(x_i, x_j)$ between each segment pair (x_i, x_j) for all x_i and x_j in the first-pass results based on the Euclidean distance of their feature vectors,

$$s(x_i, x_j) = exp\left(-\frac{\|f(x_i) - f(x_j)\|_2}{\sigma}\right), \tag{5}$$

where $f(x_i)$ is the feature vector of segment x_i, which can be either $f_1(x_i)$, $f_2(x_i)$ or $f_3(x_i)$ in Section III, and σ is the variance of the values of $\|f(x_i) - f(x_j)\|_2$ for all segment pairs (x_i, x_j) in the first-pass retrieved list for the given query. Therefore, $s(x_i, x_j)$ is between 0 and 1, and smaller $\|f(x_i) - f(x_j)\|_2$ implies larger $s(x_i, x_j)$.

(iii) Find the K nearest neighbors for each segment x_i based on $s(x_i, x_j)$, or the K segments x_j with the highest similarity $s(x_i, x_j)$, which is denoted as $N(x_i)$.

(iv) Then a score $w(x_i)$ is computed for each segment x_i, which is to be used in the next step. $w(x_i)$ is the interpolation of x_i's initial score $w_0(x_i)$ obtained in step (i) and the initial scores $w_0(x_j)$ of its "nearest neighbors of both directions"[2] weighed by their similarities $s(x_i, x_j)$:

$$w(x_i) = (1-\alpha)w_0(x_i) + \alpha \sum_{\substack{x_j, \\ x_j \in N(x_i), \\ x_i \in N(x_j)}} s(x_i, x_j) w_0(x_j), \tag{6}$$

where the summation is over all segments x_j which is among the K nearest neighbors of x_i of both directions, and α is the interpolation weight. The first term on the right hand side of (6) implies only $1 - \alpha$ of the initial score $w_0(x_i)$ is kept with the segment x_i, while the second term implies α of the initial scores $w_0(x_j)$ of segments x_j who are nearest neighbors of x_i weighted by $s(x_i, x_j)$ are added to the score of x_i. If many of x_i's nearest neighbors are top N segments with initial scores of 1, $w(x_i)$ may be increased, and it can be positive with large value regardless of whether its initial score is 1, -1 or 0. Likewise, if most of x_i's nearest neighbors are bottom N segments with initial scores -1, $w(x_i)$ may be reduced and can be very negative.

(v) All segments x are taken as positive examples for SVM training if $w(x) > 0$, and taken as negative examples if $w(x) < 0$. We further define $C(x) = |w(x)|$, which is considered to be proportional to the reliability for example x, regardless of whether $w(x)$ is positive or negative.

If α in (6) is 0, the above procedure reduces to taking top and bottom N segments as training examples, and $C(x) = 1$ for all of these examples, or exactly the plain SVM in Section II-B.

B. SVM Enhancement

With the procedure in the above subsection, new sets of positive examples $\mathcal{X}_T = \{x_1^t, \ldots, x_i^t, \ldots, x_{n_t}^t\}$ and negative examples $\mathcal{X}_F = \{x_1^f, \ldots, x_j^f, \ldots, x_{n_f}^f\}$ are obtained, where n_t and n_f are the numbers of positive and negative examples respectively. For each example x, there is a positive real number $C(x)$ proportional to the reliability of the example. To learn better from these examples considering their reliability, the formulation of SVM in (2) can be slightly modified to include $C(x)$ obtained above. There are at least three possible modifications as listed below:

- *Slack Variable Scaling* [54]: In this approach, the slack variable corresponding to each example x is scaled by $C(x)$,

$$\min_{w, \epsilon_i^t, \epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}_T} C\left(x_i^t\right) \epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}_F} C\left(x_j^f\right) \epsilon_j^f \tag{7}$$

such that

$$\forall x_i^t \in \mathcal{X}_T, \quad w \cdot f\left(x_i^t\right) \geq 1 - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$
$$\forall x_j^f \in \mathcal{X}_F, \quad w \cdot f\left(x_j^f\right) \leq -1 + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

Because the slack variables ϵ_i^t and ϵ_j^f are scaled by $C(x_i^t)$ and $C(x_j^f)$ in (7), those examples with higher reliability, whether positive or negative, are given higher priority when training the SVM model to minimize (7), while less emphasis is given to examples with lower reliability. Therefore, the weight vector w learned naturally considers the reliability of the examples.

- *Margin Scaling*: Here the original margins $+1$ and -1 for SVM are replaced by $C(x)$ and $-C(x)$ obtained here for each example,

$$\min_{w, \epsilon_i^t, \epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}_T} \epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}_F} \epsilon_j^f, \tag{8}$$

such that

$$\forall x_i^t \in \mathcal{X}_T, \quad w \cdot f\left(x_i^t\right) \geq C\left(x_i^t\right) - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$
$$\forall x_j^f \in \mathcal{X}_F, \quad w \cdot f\left(x_j^f\right) \leq -C\left(x_j^f\right) + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

In the above constraints, clearly examples with higher reliability are required to have larger margins. Hence, to minimize (8), the weight vector w learned in this way should try to give positive examples with higher reliability larger values of $w \cdot f(x)$, or locate them farther away from the

[2] x_j is among the K nearest neighbors of x_i and vice versa

dividing hyperplane. For positive examples with lower reliability, on the other hand, smaller values of $w \cdot f(x)$ may be acceptable, or they can be closer to the dividing hyperplane. The negative examples are considered in the similar way. In other words, the model thus learned should better separate positive and negative examples with large reliability because of the larger margins, but pay less attention on discriminating the less reliable examples.

- *Combined Slack Variable & Margin Scaling*: Certainly, it is possible to scale both the slack variables and the margins simultaneously:

$$\min_{w,\epsilon_i^t,\epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}_T} C\left(x_i^t\right)\epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}_F} C\left(x_j^f\right)\epsilon_j^f \quad (9)$$

such that

$$\forall x_i^t \in \mathcal{X}_T, \quad w \cdot f\left(x_i^t\right) \geq C\left(x_i^t\right) - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$
$$\forall x_j^f \in \mathcal{X}_F, \quad w \cdot f\left(x_j^f\right) \leq -C\left(x_j^f\right) + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

In the above three cases, when $C(x) = 1$ for all positive and negative examples, the three enhanced SVMs are reduced to the plain SVM in (2).

V. Experimental Setup

We tested the proposed approaches on two different sets of spoken archives. The first was a set of recorded lectures for a course (**Lecture**), and the second a set of broadcast news (**News**). Mean average precision (MAP) [55] was used as the retrieval performance measure. Pair-wise t-test [56] with significance level at 0.05 was used to test the significance for the performance improvement. The package CVXOPT[3] was used for SVM optimization. γ and γ' in (2), (7), (8) and (9) were both set to be the inverse of the average of the norms for the feature vectors of all segments in the first-pass retrieved list, as was done previously.[4] The parameter δ in (4) was set to 10 in all the experiments. The setup for the experiments for the above two sets of testing spoken archives are described below.

A. Lecture

This is a corpus of 45 hours of recorded lectures for a course offered at National Taiwan University. We split the corpus into two parts: 12 hours for acoustic model training and 33 hours for testing. Retrieval tests were performed over the testing set. This spoken archive was produced by a single instructor, primarily in Mandarin Chinese but with many English words embedded in the Mandarin utterances. Another set of slides for the lectures were also available, which is completely in English. The lexicon used here was a combination of a Chinese dictionary with 10.7 K words plus 2 K English words taken from the slides. Because of the lack of corpora matched to the topic (technical content of the course) and the style (spontaneous monologue) for the spoken archive considered here, a Chinese trigram language model was first trained from the Mandarin Giga-word corpus[5] released by Linguistic Data Consortium. We also trained an English unigram language model from the slides, and then linearly interpolated it with the above Chinese trigram model. Each spoken segment in the corpus was transcribed into a lattice with beam width 50. 162 Chinese queries were manually selected as testing queries, each consisting of a single word.

TABLE I
1-Best Character Accuracies (for Mandarin Chinese Parts) of **Lecture** for Different Sets of Acoustic Models

	SI	ADP1	ADP2	SD
Character Accuracy	50.26%	62.55%	72.93%	84.08%

In order to evaluate the performance of the proposed approach under different recognition accuracies, we used four sets of acoustic models for generating the lattices:

- Speaker Independent Model (SI): trained by Maximum Likelihood criterion with 4602 state-tied triphones spanned from 35 Mandarin monophones, using a corpus of clean read speech in Mandarin including 24.6 hours of data produced by 100 males and 100 females.
- Speaker Adaptation Model 1 (ADP1): adapted from the above SI model with 500 utterances (about 20 minutes) taken from the training set of the **Lecture** corpus mentioned above. Only global MLLR was applied.
- Speaker Adaptation Model 2 (ADP2): adapted from the SI model with 500 utterances just as ADP1 above, but with MLLR with 256 classes followed by maximum a posterior estimation.
- Speaker Dependent Model (SD): trained on the 12-hour training set of the **Lecture** corpus mentioned above with 6620 state-tied triphones spanned from 35 Mandarin monophones and 39 English monophones.

In the first three cases, since the acoustic models (SI, ADP1 and ADP2) were based on Mandarin phonemes only, the English words embedded in the Chinese utterances were transcribed into Chinese word sequences with similar pronunciation, which made the retrieval task more challenging. In the last case, the speaker dependent models (SD) included triphones developed from the phoneme set including both Mandarin and English phonemes, it was therefore possible to transcribe the English words correctly. The character accuracies (for Chinese parts only) of the 1-best transcriptions with the four different sets of acoustic models are shown in Table I.

B. News

We used a broadcast news corpus in Mandarin Chinese[6] as the second spoken archive for testing. The news stories were recorded from TV stations in Taipei from 2001 to 2003, with a total length of 198 hours [57]. 174 Chinese queries were manually selected as testing queries, each consisting of a single word.

For the recognition of **News**, we used a 60 K-word lexicon, a tri-gram language model trained on 39 M words of Yahoo news, and a set of acoustic models with 64 Gaussian mixtures per state and 3 states per model trained on a corpus of 24.5 hours of broadcast news different from the archive tested here.

[3]http://abel.ee.ucla.edu/cvxopt/.

[4]SVM-light used the same strategy to derive the parameters if not given.

[5]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2005T14

[6]publicly available via Association for Computational Linguistics and Chinese Language Processing (ACLCLP)

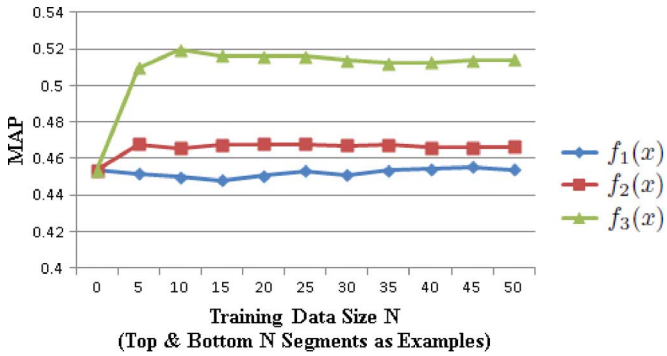

Fig. 3. MAP performance for **Lecture** with SI models yielded with feature vectors $f_1(x)$, $f_2(x)$ and $f_3(x)$ in Section III with plain SVM in Section II-B: top/bottom N segments in the first-pass results were selected as positive/negative examples. N is the size of the positive/negative example sets used in SVM training.

147 right context-dependent Initial models plus context-independent Final models[7] were used as the acoustic models, and 39-dimension MFCCs with cepstral mean and variance normalization (CMVN) applied were used as the acoustic features. Each spoken segment in the corpus was transcribed into a lattice with beam width 100. Since 48% and 31% of the speech in the corpus was produced by the reporters and respondents respectively in quite spontaneous form including relatively high background noise, and only 147 acoustic models were used here for simplicity, the character accuracy for the archive was only 54.43%.

VI. Experimental Results

A. Different Feature Vectors and Plain SVM

First of all, we tested the performance of feature vectors $f_1(x)$, $f_2(x)$ and $f_3(x)$ in Section III for plain SVM, that is, top and bottom N segments in the first-pass results were taken as positive/negative examples to train the SVM model. Fig. 3 shows the MAP performance for **Lecture** as functions of N,[8] or the size of positive/negative example sets used in SVM training. Only the results for the speaker independent (SI) models were shown in Fig. 3. The points for $N = 0$ represent the original first-pass results which are taken as the baseline.

We found that $f_1(x)$ yielded no improvement obviously because the query term usually included a sequence of phonemes, but the acoustic characteristics of the different phonemes in the MFCC vector sequence were averaged and smoothed in $f_1(x)$, which could not represent the hypothesized region. More sophisticated feature vector representations, $f_2(x)$ or $f_3(x)$, yielded improvements because the acoustic characteristics for each phoneme or even each HMM state were used, which therefore represented the hypothesized region much better. $f_3(x)$ obviously performed the best, or the HMM states were able to represent very well the acoustic characteristics within a hypothesized region. Similar results were obtained for other sets of acoustic models, ADP1, ADP2 and SD, but left out here.

[7]"Initial" is the initial consonant of a Mandarin syllable, and "Final" is the vowel or diphthong part but including the optional medial and nasal ending.

[8]If in the first-pass results there were fewer than $2N$ spoken segments, N was simply set to half of the number of retrieved segments.

TABLE II
MAP Performance Yielded With Feature Vector $f_3(x)$ in Section III for Plain SVM When Different Numbers of Top/Bottom Segments (N) in the First-Pass Results Were Selected as Positive/Negative Examples for the Two Testing Spoken Archives **Lecture** and **News**, and Four Different Acoustic Model Sets, SI, ADP1, ADP2, and SD, for **Lecture**. The First-Pass Results Obtained Before PRF are Taken as the Baselines, and the Superscript Labels * Indicate Significantly Better Than the Baselines in Terms of the Pair-Wise t-Test With Significance Level at 0.05

	Lecture				**News**
	SI	ADP1	ADP2	SD	
first pass (baseline)	0.4536	0.5539	0.7111	0.8041	0.6302
$N = 5$	0.5098*	0.6290*	0.7559*	0.8197*	0.6359
$N = 10$	0.5194*	0.6381*	0.7602*	0.8197*	0.6420*
$N = 15$	0.5161*	0.6393*	0.7584*	0.8179*	0.6471*
$N = 20$	0.5159*	0.6410*	0.7585*	0.8158*	0.6500*
$N = 25$	0.5159*	0.6387*	0.7544*	0.8150*	0.6507*
$N = 30$	0.5136*	0.6397*	0.7506*	0.8118	0.6522*
$N = 35$	0.5123*	0.6359*	0.7465*	0.8112	0.6515*
$N = 40$	0.5124*	0.6352*	0.7459*	0.8105	0.6514*
$N = 45$	0.5138*	0.6330*	0.7411*	0.8101	0.6520*
$N = 50$	0.5139*	0.6315*	0.7425*	0.8081	0.6521*

Table II shows the MAP performance yielded with the feature vector $f_3(x)$ with plain SVM when different numbers of top/bottom segments were taken as positive/negative examples. N in Table II is the number of top/bottom segments used in training. The results for the two testing spoken archives **Lecture** and **News** are listed here with four different sets of acoustic models for **Lecture**, SI, ADP1, ADP2 and SD. The first-pass results obtained before PRF are taken as the baselines, and the superscript labels * indicate significantly better than the baselines in terms of the pair-wise t-test with significance level at 0.05.

We can easily find that the plain SVM trained with feature vector $f_3(x)$ when taking top and bottom N segments as training examples always offered some improvements in all cases as compared to the baseline for both **Lecture** and **News** archives, all different sets of acoustic models for **Lecture**, and all values of N tested here. The improvements achieved were always significant except for SD models for **Lecture** with $N \geq 30$ and for **News** with $N = 5$. From Table II, we also observed that as the example set size N was raised the MAP first increased and then slightly decreased in most cases. This is reasonable because larger N implied more training data were used in training the SVM model, and the disturbances caused by the incorrect assumption about the relevance of the training segments (irrelevant segments assumed to be relevant and vice versa) can be diluted. However, when N was very large, since usually there were only limited number of relevant segments for each query, more irrelevant segments were inevitably included in the pseudo-relevant training set and taken as relevant, which caused the performance degradation. In the case of SD models for **Lecture**, the improvements became insignificant when $N \geq 30$. This is the case of acoustic models with very good quality, or the segments were reasonably well ranked in the first-pass retrieved list. Hence, we may assume most relevant segments were within the top 25 ($N = 25$) or less. For $N \geq 30$, some irrelevant segments were inevitably included and taken as positive examples, which is very probably the

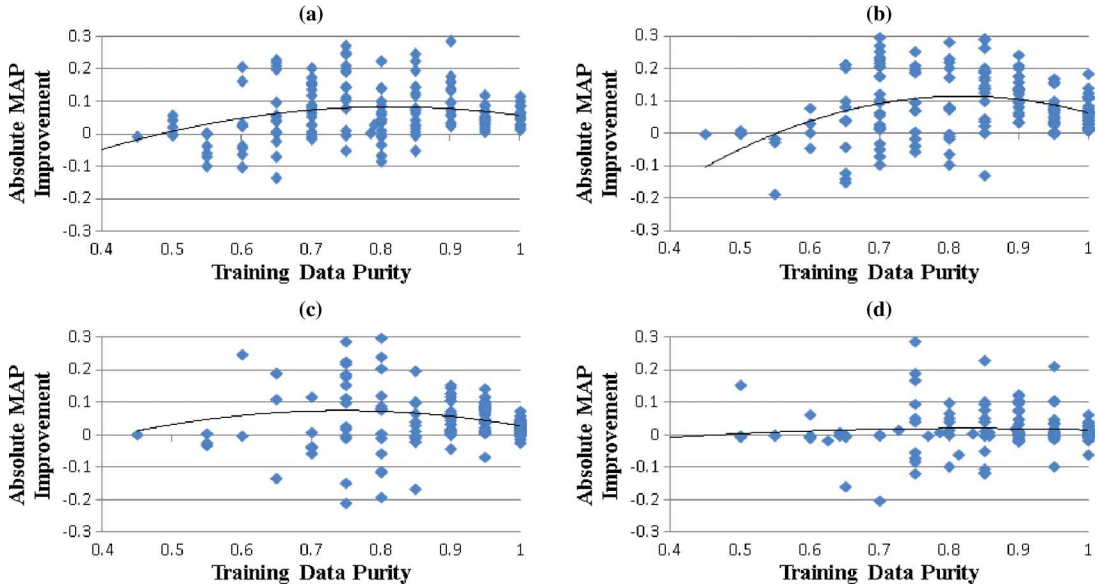

Fig. 4. Distribution of absolute MAP improvement versus the training data purity for each query for **Lecture** with feature vector $f_3(x)$ when taking top/bottom 10 segments as training examples ($N = 10$ in Table II). Training data purity is the average of the percentages of pseudo-relevant segments being relevant and pseudo-irrelevant segments being irrelevant. Each point in the figures represents a query. The curves in the figures are the quadratic trend lines. (a), (b), (c) and (d) are respectively for the results with different sets of acoustic models, SI, ADP1, ADP2, SD.

reason that the improvements became insignificant. For the results of **News**, the improvements were always significant except $N = 5$. This is probably because **News** contains speech produced by many different speakers under many different environments, so enough training data were necessary to cover enough acoustic variations.

Since we were never able to ensure all the pseudo-relevant/-irrelevant training data were correct, PRF was not supposed to improve the performance of every query. We may assume that PRF improved the performance of some queries but degraded that of the others. Our goal was simply that the former occurred much more than the latter. Here we are interested to see how SVM with $f_3(x)$ performed with such corrupted training data. We first define the training data purity for each query as the average of the percentages of pseudo-relevant segments being actually relevant and pseudo-irrelevant segments being actually irrelevant. Fig. 4 shows the distribution of the absolute MAP improvement achieved with each query versus the training data purity for that query with $f_3(x)$ for **Lecture** when taking top and bottom 10 segments as examples ($N = 10$ in Table II). Fig. 4(a), (b), (c) and (d) are respectively for the four different sets of acoustic models, SI, ADP1, ADP2 and SD. Each point in the figures represents the results for one query, with vertical scales being the absolute MAP improvement for the query, and the horizontal scales being the training data purity. Negative improvement means the MAP performance for the query was actually degraded after PRF. The curves in the figures are the quadratic trend lines.

At the first glance of Fig. 4, it seems surprising that higher training data purity did not always imply larger MAP improvement. This is probably because the queries with higher training data purity usually had higher MAP for the first-pass retrieved results, the space left for further improvement was therefore limited. On the other hand, we can see in all four cases in Fig. 4 the MAP improvements were positive for much more queries, although also inevitably negative for some smaller number of queries. The very corrupted training data really degraded the performance, but we also observed that even when the training data purity was less than 70%, MAP improvements were still achieved for some queries.

TABLE III
PERCENTAGE OF QUERIES WITH MAP PERFORMANCE DEGRADED AFTER PRF WITH FEATURE VECTOR $f_3(x)$ WHEN TAKING TOP/BOTTOM 10 SEGMENTS AS TRAINING EXAMPLES ($N = 10$ IN TABLE II)

	Lecture				News
	SI	ADP1	ADP2	SD	
Percentage of Queries Degraded (%)	16.05	14.81	14.20	27.78	35.00

Table III shows the percentage of queries whose MAP was degraded after PRF with feature vector $f_3(x)$ when taking top/bottom 10 segments as training examples ($N = 10$ in Table II). The results for **Lecture** with SI, ADP1, and ADP2 models look reasonable (around 15%), or roughly for 85% of queries the performances were improved after PRF. The situation was worse for SD of **Lecture** (27.78%) and the worst for **News** (35.00%). A possible reason for the situation of **News** is that it covered a wide variety of speakers and environments, which made the feature vector $f_3(x)$ based on MFCC vector sequences relatively inadequate. However, MAP performance of 65% of the queries for **News** was still improved by PRF in that case.

B. Enhanced SVM

In this section, the enhanced SVM described in Section IV is tested and analyzed. Because the feature vector $f_3(x)$ yielded the best results in Fig. 3 above, only $f_3(x)$ was used in the following experiments.

TABLE IV
MAP PERFORMANCE FOR **Lecture** WITH SI MODELS YIELDED BY THE ENHANCED SVM IN SECTION IV WITH FEATURE VECTOR $f_3(x)$. (A) FOR PLAIN SVM TAKING TOP/BOTTOM N ($5 \leq N \leq 50$) SEGMENTS AS TRAINING EXAMPLES (BASELINE), AND (B) FOR ENHANCED SVM WHERE THE TOP/BOTTOM N SEGMENTS WERE ASSIGNED SCORES $w_0(x) = 1$ OR -1 AT THE STEP (I) OF THE PROCEDURE IN SECTION IV-A, BUT THE SIZES OF POSITIVE/NEGATIVE EXAMPLE SETS DEPENDED ON THE VALUES OF $w(x)$ OBTAINED IN STEP (IV) OF THE PROCEDURE INCLUDING *Slack Variable Scaling* (COLUMN (B-1)), *Margin Scaling* (COLUMN (B-2)), AND *Combined Slack Variable & Margin Scaling* (COLUMN (B-3)). $\alpha = 0.8$ IN (6) AND $K = 5$ FOR STEP (III) IN SECTION IV-A. THE SUPERSCRIPT LABELS * INDICATE SIGNIFICANTLY BETTER THAN THE CORRESPONDING RESULTS IN COLUMN (A)

N	(a) top & bottom N segments as training examples	(b) Enhanced PRF in Section IV		
		(b-1) *Slack Variable Scaling*	(b-2) *Margin Scaling*	(b-3) *Combined Slack Variable & Margin Scaling*
5	0.5098	0.4979	0.5206*	0.5189*
10	0.5194	0.5088	0.5248	0.5327*
15	0.5161	0.5112	0.5189	0.5330*
20	0.5159	0.5115	0.5160	0.5335*
25	0.5159	0.5145	0.5178	0.5366*
30	0.5136	0.5143	0.5202	0.5337*
35	0.5123	0.5126	0.5175	0.5311*
40	0.5124	0.5142	0.5185	0.5316*
45	0.5138	0.5153	0.5200	0.5329*
50	0.5139	0.5156	0.5191	0.5320*

In Table IV, MAP performance for **Lecture** yielded by the enhanced SVM is presented. Only the results for SI models are reported. Column (a) is the results for plain SVM taking the top/bottom N segments as training examples with N ranged from 5 to 50, which was used as the baseline here. Therefore, the results in Column (a) were simply copied from the **Lecture** SI column of Table II. Section VI-B is for the enhanced SVM, in which the top/bottom N segments were first assigned initial scores $w_0(x) = 1$ or -1 at step (i) of the procedure in Section IV-A, but the sizes of positive/negative example sets finally depended on the values of $w(x)$ obtained in step (iv) of the procedure. α in (6) was set to 0.8, and K was set to 5 at the step (iii) of the procedure in Section IV-A. As in Section IV-B, the SVM can be enhanced in three ways, *Slack Variable Scaling*, *Margin Scaling*, and *Combined Slack Variable & Margin Scaling*, respectively corresponding to columns (b-1), (b-2) and (b-3) in Section VI-B. The superscript labels * indicate significantly better than the corresponding results in column (a) for the same N.

We can observe that the results using *Slack Variable Scaling* were close to but not able to surpass the baseline (columns (b-1) vs (a)), whereas *Margin Scaling* outperformed the baseline (columns (b-2) vs (a)) for all values of N, but the improvements were not significant except for $N = 5$. This is probably because *Margin Scaling* in (8) utilized the scores $C(x)$ more aggressively than *Slack Variable Scaling* in (7). In *Slack Variable Scaling*, the effects of $C(x_i^t)$ or $C(x_j^f)$ proportional to the reliability of the segments x_i^t or x_j^f in (7) were in fact deleted if ϵ_i^t or ϵ_j^f were zero, or x_i^t or x_j^f were outside of the margin, so the values of $C(x)$ for most examples x did not have any effect on the training results. On the other hand, in *Margin Scaling*, the values of $C(x)$ were used to define the margins of the constraints, or $C(x)$ of every example x influenced the model learned. It is clear that *Combined Slack Variable & Margin Scaling* offered the best improvements over the baseline, and the improvements were significant regardless of N (columns (b-3) vs (a)). Below only the results for the *Combined Slack Variable & Margin Scaling* were reported for further discussion.

Fig. 5 shows the MAP performance comparison for **Lecture** yielded by plain SVM in Section II-B and enhanced SVM in Section IV as functions of N with feature vector $f_3(x)$ very similar to Table IV. However here Fig. 5(a), (b), (c) and (d) respectively show the results for different sets of acoustic models, SI, ADP1, ADP2 and SD, and in each case for the enhanced SVM the results for different choices of the values of K, or the number of nearest neighbors considered, are shown. Therefore, the curves of plain SVM and enhanced SVM ($K = 5$) in Fig. 5(a) are respectively columns (a) and (b-3) of Table IV. Same as Table IV, α in (6) was set to 0.8.

First consider Fig. 5(a), (b) and (c). We can observe that enhanced SVM always offered improvements over the plain SVM with SI, ADP1 and ADP2 models in all cases regardless of the values of N and K, except $N = 5$. In other words, the enhanced SVM started with only top/bottom 5 segments ($N = 5$) may not work reasonably, but became very well as long as $N > 5$. This verified the effectiveness of the proposed approach. For the enhanced SVM with SI model, $K = 5$ offered slightly better performance than $K = 10$, and for ADP1 and ADP2 models, $K = 5$ and 10 were comparable in MAP performance. This implied that the proposed enhanced SVM was not very sensitive to the value of K as long as K was large enough to consider sufficient neighbors. When K was equal to 1, subtle improvements over the baselines were still observed. However, different situation occurred for SD models in Fig. 5(d). Since the SD models were of very high quality (1-best character accuracy of 84.08% in Table I), the first-pass retrieved results were in fact ranked very well. In that case, selecting top/bottom N segments may be sufficient to generate positive/negative example sets with very high training data purity. This is probably why the enhanced SVM may not offer too much benefit over the plain SVM.

Fig. 6 is exactly the same results as those in Fig. 5, except for **News** rather than for **Lecture**. The general trends observed on the results for **News** in Fig. 6 were consistent with those on **Lecture** in Fig. 5. The enhanced SVM was superior than plain SVM regardless of the values of N and K.

Table V presents the MAP performance yielded by the enhanced SVM but with different values of α in (6) ($\alpha = 0.8$ for all results reported above) with $K = 5$ and $N = 10$. $\alpha = 0.0$ (or the plain SVM) is taken as the baseline here. The superscript labels * indicate significantly better than the baseline. The highest MAP in each column was in bold. For the results of **Lecture**, we observed that with SI, ADP1 and ADP2 models better MAP were achieved with larger values of α. The peaks of the MAP performance were reached when α was 0.9, 0.8 and 0.9 for SI, ADP1 and ADP2 respectively. Note that α is the interpolation weight between the two terms in (6). A larger value of α implies a smaller weight of the first term in (6), or a smaller weight for the initial score $w_0(x)$ in evaluating the function $w(x)$. This implies the initial assignments for the positive/negative examples were less reliable, which should be compensated by the higher weighted initial scores $w_0(x)$ of nearest neighbors in the second term of (6). This is exactly the case for relatively poor acoustic

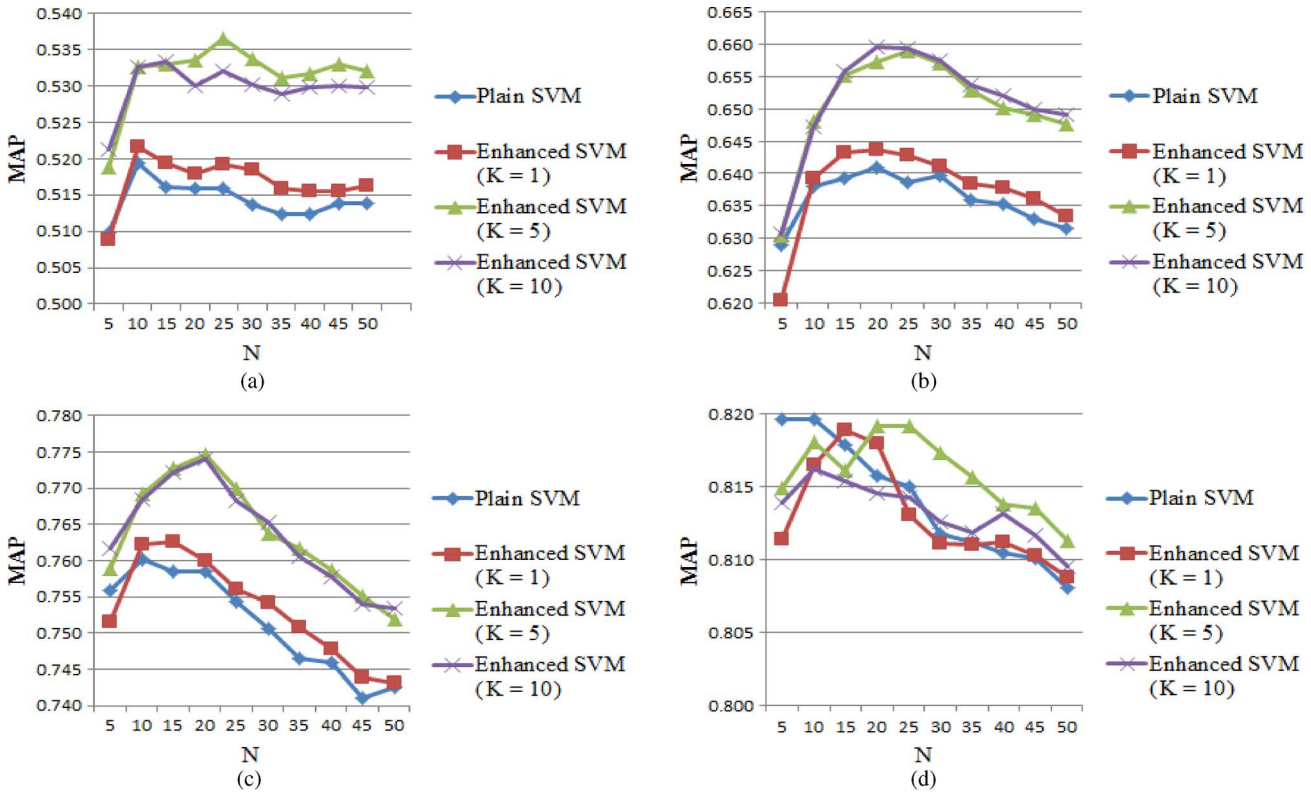

Fig. 5. MAP performance for **Lecture** respectively with four sets of acoustic models SI, ADP1, ADP2 and SD yielded by the plain SVM in Section II-B and the enhanced SVM in Section IV (*Combined Slack Variable and Margin Scaling* only) as functions of N with feature vector $f_3(x)$. Results of different choices of K are compared, where K is the number of nearest neighbors considered in step (iii) in Section IV-A. The plain SVM (blue curves) are the baselines. For plain SVM, N is the size of positive/negative example sets. For enhanced SVM, the initial scores $w_0(x) = 1$ or -1 were assigned to the top/bottom N segments at step (i) of the procedure in Section IV-A, but the sizes of positive and negative example sets depended on the values of $w(x)$ obtained in step (iv) of the procedure. (a) SI model; (b) ADP1 model; (c) ADP2 model; (d) SD model.

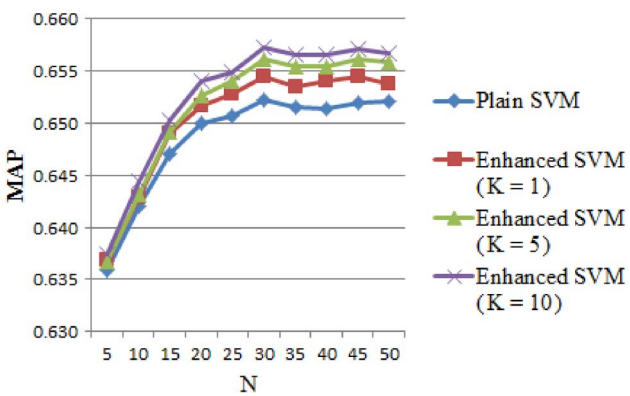

Fig. 6. MAP performance for **News** yielded by the plain SVM and enhanced SVM (*Combined Slack Variable and Margin Scaling*) as functions of N for different values of K with feature vector $f_3(x)$, very similar to those in Fig. 5, except for **News** here rather than for **Lecture**.

TABLE V
MAP PERFORMANCE YIELDED BY PLAIN SVM AND ENHANCED SVM BUT WITH DIFFERENT VALUES OF α IN (6) ($\alpha = 0.8$ FOR ALL RESULTS REPORTED ABOVE) WITH $K = 5$ AND $N = 10$. $\alpha = 0.0$ (OR THE PLAIN SVM) IS TAKEN AS THE BASELINE. THE SUPERSCRIPT * INDICATE SIGNIFICANTLY BETTER THAN THE BASELINE. THE HIGHEST MAP IN EACH COLUMN WAS IN BOLD

	Lecture				**News**
	SI	ADP1	ADP2	SD	
$\alpha = 0.0$ (baseline)	0.5194	0.6381	0.7602	0.8197	0.6423
$\alpha = 0.1$	0.5223*	0.6390	0.7616	0.8206	0.6424
$\alpha = 0.2$	0.5239*	0.6397	0.7635	0.8213	0.6426
$\alpha = 0.3$	0.5256*	0.6408*	0.7642*	**0.8217**	0.6427
$\alpha = 0.4$	0.5268*	0.6418*	0.7646	0.8214	0.6427
$\alpha = 0.5$	0.5289*	0.6432*	0.7656*	0.8215	0.6429
$\alpha = 0.6$	0.5309*	0.6456*	0.7672*	0.8211	0.6430*
$\alpha = 0.7$	0.5318*	0.6472*	0.7692*	0.8194	0.6431*
$\alpha = 0.8$	0.5327*	**0.6481***	0.7691*	0.8181	**0.6435**
$\alpha = 0.9$	**0.5359***	0.6459*	**0.7712***	0.8145	0.6365

models SI, ADP1 and ADP2, and explains why for SD models with very high recognition accuracy $\alpha = 0.3$ gave the highest MAP. For **News**, with relatively poor accuracy, we similarly observe that larger α led to better performance, and the best result was achieved at $\alpha = 0.8$.

C. Comparison With User Relevance Feedback

All approaches discussed above are based on SVM in the scenario of PRF. Another related scenario is user relevance feedback, and in this subsection we wish to compare the proposed approach with user relevance feedback. The scenario of user relevance feedback is very similar to PRF, except that the user provides the positive/negative examples for training, so we know they are exactly relevant/irrelevant, while in PRF these examples are only pseudo-relevant/-irrelevant derived by the system.

In the experiment below, in the scenario of user relevance feedback, we assume the user browsed the first-pass retrieved list from the top, and gave the system the relevance informa-

TABLE VI
THE COMPARISON OF THE PROPOSED APPROACH WITH PRF AND USER RELEVANCE FEEDBACK IN TERMS OF MAP. ROW (a) IS THE MAP PERFORMANCE OF THE FIRST-PASS RESULTS. SECTION VI-B IS THE RESULTS FOR THE PROPOSED APPROACH WITH PRF, INCLUDING ROW (b-1) FOR PLAIN SVM WITH $N = 10$ AND ROW (b-2) FOR ENHANCED SVM WITH $N = 10$, $K = 5$ AND $\alpha = 0.8$. SECTION VI-C IS THE RESULTS FOR USER RELEVANCE FEEDBACK WHEN THE CORRECT RELEVANCE INFORMATION OF THE TOP 20 SEGMENTS IN THE FIRST-PASS RETURNED LIST WAS GIVEN ($M = 20$), INCLUDING THE UPPER BOUND RESULTS IN ROW (c-1) ASSUMING THE RANKING OF ALL THE SEGMENTS INCLUDING THE TOP 20 ALREADY SEEN BY THE USER COULD BE RE-RANKED, AND ROW (c-2) IN WHICH THE RANKING OF THE TOP 20 SEGMENTS IN THE LIST WERE FROZEN

Approach		Lecture				News
		SI	ADP1	ADP2	SD	
(a) First-pass result		0.4536	0.5539	0.7111	0.8041	0.6302
(b) Pseudo-relevance Feedback (PRF)	(b-1) Plain SVM ($N = 10$)	0.5194	0.6381	0.7602	0.8197	0.6420
	(b-2) Enhanced SVM ($N = 10$, $K = 5$, $\alpha = 0.8$)	0.5359	0.6481	0.7712	0.8217	0.6435
(c) User Relevance Feedback	(c-1) Upper Bound	0.5544	0.6773	0.7894	0.8303	0.6659
	(c-2) Top M Frozen	0.4718	0.5753	0.7180	0.8026	0.6433

tion (relevant or irrelevant) for each of the top M spoken segments on the list.[9] These segments labelled by the user were then taken as positive and negative examples for training the SVM to be used for re-ranking the spoken segments in the first-pass retrieved list below the top M segments following exactly the same process as described in Section II-C. The sizes of positive/negative example sets were not N but depended on the user information.[10] Note that here the positive/negative examples were labelled by the user and known to be correct, so the enhancement processes described in Section IV considering the reliability of the pseudo-relevant/-irrelevant examples by the function $C(x)$ are not needed at all. Therefore, only the plain SVM was used here without any enhancement processes. Note that in evaluating MAP the order of the top M labelled spoken segments should be "frozen" [58], [59]. Because in practice these top M spoken segments have been seen by the user and given the relevance information, re-ranking them is meaningless. This concern does not exist for PRF.

Because the users usually browse the retrieved objects based on the ranking orders provided by the system, the assumption that the user labels the top M objects on the returned list in the experiments here is quite realistic [59], [60]. Nevertheless, this assumption is not always true. In the scenario of relevance feedback, for example, the system can actively learn more from the users by asking the users to label the most confused objects [61]. However, from the users' perspective, browsing the objects based on the ranking orders gives them the opportunities to see the most relevant objects (if the system performance is good enough), whereas in active learning their time can be wasted for labelling the objects not very relevant.

Table VI presents the comparison of MAP performance yielded by the proposed approach with PRF and by user relevance feedback. Row (a) is the MAP performance of the first-pass results as the baseline. Section VI-B is the results for the proposed approach with PRF, including row (b-1) for plain SVM with $N = 10$ ($N = 10$ in Table II), and row (b-2) for enhanced SVM with $N = 10$, $K = 5$ and $\alpha = 0.8$ ($\alpha = 0.8$ in Table V). Section VI-C is the results for user relevance feedback when the correct relevance information of the top 20 segments in the first-pass returned list was given ($M = 20$ as mentioned above). Row (c-1) is the upper bound assuming that the ranking of all the segments including the top 20 already seen by the user could be re-ranked. Although row (c-1) seems very good, it is not realistic. Furthermore, since the top 20 segments with relevance information provided by the user were used for training, the results were improved in any case because all segments labelled relevant were ranked on the top, and all labelled irrelevant at the bottom. This automatically improved the MAP significantly. Even if those segments below the top M not browsed by the user were re-ranked in wrong directions, the MAP degradation caused by the latter could be easily absorbed by the increase in MAP due to the former. The results in row (c-2) are realistic, since the ranking of the top 20 segments with relevance information given by the users were frozen.

We observed here that the proposed approaches based on PRF outperformed the user relevance feedback with the realistic frozen ranking assumption in terms of MAP in most cases (rows (b-1), (b-2) vs (c-2)), even though the user did not need to give any feedback during retrieval. In fact, because MAP values were primarily dominated by the top several items, in the case of user relevance feedback, the improvements in MAP scores were relatively limited since the top 20 items were frozen. In other words, although user relevance feedback may offer more reliable examples for training, and thus better SVM model may be learned, not too much space was left for MAP improvement. In this respect, the proposed approaches with PRF provide an effective way for improving the retrieval performance in terms of MAP without enlisting the help from the user.

In Table VI, it is surprising to find that with the SD models user relevance feedback with frozen ranking assumption was even lower than the first-pass results (rows (c-2) vs (a) under column SD). This implies that in this case the SVM model learned from user relevance information was not generalizable to the segments the user had not browsed. Since the SD model was already well matched to the testing archive, some irrelevant segments within top 20 labelled by the user very possibly had parts very close to the query phonetically or acoustically but actually irrelevant. Therefore, it is in fact very difficult to discriminate them from the relevant segments simply based on acoustic information. If this is the case, the model thus trained forced to discriminate the irrelevant segments from the relevant ones may be over-fitted to the training data, and therefore not generalizable to the segments not labelled by the users. This im-

[9]In the experiment here, we already knew the relevant/irrelevant spoken segments for all of the queries considered, so we simply simulated the user-labelled information to be the relevance information we had for the top M spoken segments for all queries.

[10]The number of positive examples plus negative ones is M.

plies other features in addition to the acoustic information may be needed. For the experiments here, on the other hand, PRF offered some improvements over the first-pass results with the SD models (rows (b-1), (b-2) vs (a) under column SD). Since in PRF those segments having parts very close to the query phonetically or acoustically were taken as positive examples, the SVM model learned was better generalizable to other segments.

VII. Conclusion

In this paper, pseudo-relevance feedback is used to automatically generate the training examples for training query-specific SVM for each query, which is then used to further re-rank the first-pass retrieved results. The feature vectors based on acoustic information were defined and used in training the SVM. The training examples are selected and weighted considering their reliability, and SVM is modified by rescaling slack variables and/or margins to consider the examples' weights. The proposed approaches were tested with two different sets of spoken archives with different speaking styles under different recognition accuracies. The results indicated simply taking top/bottom-ranked spoken segments as positive/negative examples already significantly improved the retrieval performance, and considering the weighted pseudo examples by modified SVM was even more helpful. We further showed that the proposed approach based on pseudo-relevance feedback may be able to yield retrieval performance better than those obtained in the scenario that correct information is provided by users.

References

[1] L.-S. Lee and B. Chen, "Spoken document understanding and organization," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 42–60, Sep. 2005.

[2] C. Chelba, T. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 39–49, May 2008.

[3] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance," in *Proc. HLT*, 2004.

[4] C. Chelba, J. Silva, and A. Acero, "Soft indexing of speech content for search in spoken documents," *Comput. Speech Lang.*, vol. 21, pp. 458–478, 2007.

[5] C. Chelba and A. Acero, "Position specific posterior lattices for indexing speech," in *Proc. ACL*, 2005.

[6] Y.-C. Pan, H.-L. Chang, and L.-S. Lee, "Analytical comparison between position specific posterior lattices and confusion networks based on words and subword units for spoken document indexing," in *Proc. ASRU*, 2007.

[7] T. Hori, I. Hetherington, T. Hazen, and J. Glass, "Open vocabulary spoken utterance retrieval using confusion networks," in *Proc. ICASSP*, 2007, pp. 73–76.

[8] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata: Application to spoken utterance retrieval," in *Proc. Workshop Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, 2004.

[9] B. Logan, P. Moreno, J.-M. van Thong, and E. Whittaker, "An experimental study of an audio indexing system for the web," in *Proc. ICSLP*, 2000.

[10] M. Akbacak, D. Vergyri, and A. Stolcke, "Open-vocabulary spoken term detection using graphone-based hybrid recognition systems," in *Proc. ICASSP*, 2008, pp. 5240–5243.

[11] Y.-C. Pan, H.-L. Chang, and L.-S. Lee, "Subword-based position specific posterior lattices (S-PSPL) for indexing speech information," in *Proc. Interspeech*, 2007.

[12] B. Logan, J.-M. Van Thong, and P. Moreno, "Approaches to reduce the effects of OOV queries on indexed spoken audio," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 899–906, Oct. 2005.

[13] R. Wallace, R. Vogt, and S. Sridharan, "A phonetic search approach to the 2006 NIST spoken term detection evaluation," in *Proc. Interspeech*, 2007.

[14] V. T. Turunen, "Reducing the effect of OOV query words by using morph-based spoken document retrieval," in *Proc. Interspeech*, 2008.

[15] V. T. Turunen and M. Kurimo, "Indexing confusion networks for morph-based spoken document retrieval," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. in Inf. Retrieval*, 2007, ser. SIGIR '07, pp. 631–638.

[16] D. Wang, J. Frankel, J. Tejedor, and S. King, "A comparison of phone and grapheme-based spoken term detection," in *Proc. ICASSP*, 2008, pp. 4969–4972.

[17] Y. Itoh, K. Iwata, K. Kojima, M. Ishigame, K. Tanaka, and S. Wook Lee, "An integration method of retrieval results using plural subword models for vocabulary-free spoken document retrieval," in *Proc. Interspeech*, 2007.

[18] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proc. ICASSP*, 2006, pp. 949–952.

[19] B. Logan, J.-M. Van Thong, and P. Moreno, "Approaches to reduce the effects of OOV queries on indexed spoken audio," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 899–906, Oct. 2005.

[20] K. Ng, "Subword-based approaches for spoken document retrieval," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, MA, USA, 2000.

[21] S. Wook Lee, K. Tanaka, and Y. Itoh, "Combining multiple subword representations for open-vocabulary spoken document retrieval," in *Proc. ICASSP*, 2005, pp. 505–508.

[22] S. Meng, P. Yu, J. Liu, and F. Seide, "Fusing multiple systems into a compact lattice index for Chinese spoken term detection," in *ICASSP*, 2008, pp. 4345–4348.

[23] Y.-C. Pan, H.-Y. Lee, and L.-S. Lee, "Interactive spoken document retrieval with suggested key terms ranked by a Markov decision process," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 632–645, Feb. 2012.

[24] S.-Y. Kong, M.-R. Wu, C.-K. Lin, Y.-S. Fu, and L.-S. Lee, "Learning on demand—Course lecture distillation by information extraction," in *Proc. ICASSP*, 2009, pp. 4709–4712.

[25] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proc. Interspeech*, 2007.

[26] M. Goto, J. Ogata, and K. Eto, "Podcastle: A web 2.0 approach to speech recognition research," in *Proc. Interspeech*, 2007.

[27] C. Alberti, M. Bacchiani, A. Bezman, C. Chelba, A. Drofa, H. Liao, P. Moreno, T. Power, A. Sahuguet, M. Shugrina, and O. Siohan, "An audio indexing system for election video material," in *Proc. ICASSP*, 2009, pp. 4873–4776.

[28] C.-H. Meng, H.-Y. Lee, and L.-S. Lee, "Improved lattice-based spoken document retrieval by directly learning from the evaluation measures," in *Proc. ICASSP*, 2009, pp. 4893–4896.

[29] J. Tejedor, D. T. Toledano, M. Bautista, S. King, D. Wang, and J. Colas, "Augmented set of features for confidence estimation in spoken term detection," in *Proc. Interspeech*, 2010.

[30] D. Wang, S. King, J. Frankel, and P. Bell, "Term-dependent confidence for out-of-vocabulary term detection," in *Proc. Interspeech*, 2009.

[31] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," in *Proc. Interspeech*, 2007.

[32] O. Kurland, L. Lee, and C. Domshlak, "Better than the real thing?: Iterative pseudo-query processing using cluster-based language models," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005.

[33] T. Tao and C. Zhai, "Regularized estimation of mixture models for robust pseudo-relevance feedback," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006.

[34] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008.

[35] Y. Lv and C. Zhai, "A comparative study of methods for estimating query language models with pseudo feedback," in *Proc. 18th ACM Conf. Inf. Knowl. Manag.*, 2009.

[36] Y. Lv and C. Zhai, "Positional relevance model for pseudo-relevance feedback," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010.

[37] W.-H. Lin, R. Jin, and A. Hauptmann, "Web image retrieval re-ranking with relevance model," in *Proc. IEEE/WIC Int. Conf. Web Intelligence WI '03*, 2003, pp. 242–248.

[38] R. Yan, A. G. Hauptmann, and R. Jin, "Negative pseudo-relevance feedback in content-based video retrieval," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003.
[39] S. Rudinac, M. Larson, and A. Hanjalic, "Exploiting visual reranking to improve pseudo-relevance feedback for spoken-content-based video retrieval," in *Proc. 10th Workshop Image Anal. Multimedia Interact. Services WIAMIS '09*, 2009.
[40] R. Yan, A. Hauptmann, and R. Jin, "Multimedia search with pseudo-relevance feedback," in *Proc. CIVR*, 2003.
[41] A. P. Natsev, M. R. Naphade, and J. TešiĆ, "Learning the semantics of multimedia queries and concepts from a small number of examples," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, ser. MULTIMEDIA '05, pp. 598–607.
[42] H.-Y. Lee, C.-P. Chen, and L.-S. Lee, "Integrating recognition and retrieval with relevance feedback for spoken term detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 7, pp. 2095–2110, Sep. 2012.
[43] H.-Y. Lee, C.-P. Chen, C.-F. Yeh, and L.-S. Lee, "A framework integrating different relevance feedback scenarios and approaches for spoken term detection," in *Proc. SLT*, 2010.
[44] Y.-N. Chen, C.-P. Chen, H.-Y. Lee, C.-A. Chan, and L.-S. Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 5644–5647.
[45] C.-P. Chen, H.-Y. Lee, C.-F. Yeh, and L.-S. Lee, "Improved spoken term detection by feature space pseudo-relevance feedback," in *Proc. Interspeech*, 2010.
[46] T.-W. Tu, H.-Y. Lee, and L.-S. Lee, "Improved spoken term detection using support vector machines with acoustic and context features from pseudo-relevance feedback," in *Proc. ASRU*, 2011.
[47] D. R. H. Miller, M. Kleber, C. Lin Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007.
[48] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007.
[49] P. Yu, K. Chen, L. Lu, and F. Seide, "Searching the audio notebook: Keyword search in recorded conversations," in *Proc. Conf. Human Lang. Technol. Empir. Meth. Natural Lang. Process.*, 2005.
[50] S. Parlak and M. Saraclar, "Spoken term detection for Turkish broadcast news," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '08)*, 2008.
[51] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowl. Disc.*, vol. 2, pp. 121–167, 1998.
[52] Y. Zhang and J. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Proc. ICASSP*, 2010, pp. 4366–4369.
[53] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on gaussian posteriorgrams," in *Proc. ASRU*, 2009.
[54] S. Hoi, M. Lyu, and R. Jin, "A unified log-based relevance feedback scheme for image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 4, pp. 509–524, Apr. 2006.
[55] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees, "The TREC spoken document retrieval track: A success story," in *Proc. Text Retrieval Conf. (TREC)*, 2000, vol. 8, pp. 16–19.
[56] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
[57] H. M. Wang, B. Chen, J. W. Kuo, and S. S. Cheng, "MATBN: A Mandarin Chinese broadcast news corpus," in *Comput. Linguist. Chinese Lang. Process*, 2005, pp. 219–236.
[58] I. Ruthven and M. Lalmas, "A survey on the use of relevance feedback for information access systems," in *Knowl. Eng. Rev.*, 2003.
[59] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templetes," in *Proc. ASRU*, 2009.
[60] T. Deselaers, R. Paredes, E. Vidal, and H. Ney, "Learning weighted distances for relevance feedback in image retrieval," in *Proc. ICPR*, 2008.
[61] B. Settles, "Active learning literature survey," 2010, Tech. Rep..

Hung-yi Lee , was born in 1986. He received the M.S. and Ph.D. degrees in communication engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2010 and 2012, respectively.

He is currently a postdoctoral fellow in Research Center for Information Technology Innovation, Academia Sinica. His research focuses on spoken content retrieval and spoken document summarization.

Lin-shan Lee (F'03) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He has been a Professor of electrical engineering and computer science at the National Taiwan University, Taipei, Taiwan, since 1982 and holds a joint appointment as a Research Fellow of Academia Sinica, Taipei. His research interests include digital communications and spoken language processing. He developed several of the earliest versions of Chinese spoken language processing systems in the world including text-to-speech systems, natural language analyzers, dictation systems, and voice information retrieval systems.

Dr. Lee was Vice President for International Affairs (1996–1997) and the Awards Committee chair (1998–1999) of the IEEE Communications Society. He was a member of the Board of International Speech Communication Association (ISCA 2002–2009), a Distinguished Lecture (2007–2008) and a member of the Overview Paper Editorial Board (since 2009) of the IEEE Signal Processing Society, and the general chair of ICASSP 2009 in Taipei. He is a fellow of ISCA since 2010, and received the Meritorious Service Award from IEEE Signal Processing Society in 2011.