Voice-based Information Retrieval — how far are we from the text-based information retrieval ?

Lin-shan Lee and Yi-cheng Pan
National Taiwan University
Taipei, Taiwan, ROC
lslee@gate.sinica.edu.tw
thomashughPan@gmail.com

Abstract — **Although network content access is primarily text-based today, almost all roles of text can be accomplished by voice. Voice-based information retrieval refers to the situation that the user query and/or the content to be retried are in form of voice. This paper tries to compare the voice-based information retrieval with the currently very successful text-based information retrieval, and identifies two major issues in which voice-based information retrieval is far behind: retrieval accuracy and user-system interaction. These two issues are reviewed, analyzed and discussed in detail. It is found that very good approaches have been proposed and very good improvements have been achieved, although there is still a very long way to go. A few successful prototype systems, among many others are presented at the end.**

Index Terms: Information Retrieval, Voice, Accuracy, User-system Interaction, Lattice, Dialogue

I. Introduction

With the rapid increase of web content, text-based information retrieval has become a very popular technology with many successful applications, which in turn generated a very successful industry today. However, this is not the end of the story, but only the beginning. The ever-increasing Internet bandwidth, the ever-decreasing storage costs, and the fast development of multimedia technologies have paved the road for more and more multimedia network content. Multimedia content usually carries speech information, and such speech information usually tells the topics and concepts relevant to the multimedia content. As a result, speech information becomes the key for indexing and retrieving such content [1], [2], [3], [4], [5], [6], [7]. In fact, although network content access is primarily text-based today, almost all roles of texts can be accomplished by voice. Not only the speech information can be used to index and retrieve multimedia content, but the user instructions and queries can also be entered in form of voice. With the many hand-held devices with multimedia functionalities commercially available and the fast increasing quantities of multimedia content over the Internet, this area is apparently getting more and more important today. This leads to the concept of voice-based information retrieval as shown in Fig. 1. In addition to using text instructions/queries to retrieve text documents as has been very popular today, either the instructions/queries or the content to be retrieved, or both of them, can be in spoken form. This actually includes three different tasks: (1)using text queries to retrieve spoken documents, (2)using spoken queries to retrieve text documents, and (3)using spoken queries to retrieve spoken documents.

In this paper we try to offer an overview of the area of voice-based information retrieval by comparing it with the currently very successful text-based counterpart. The discussions are primarily focused on two key issues, the retrieval accuracy and the user-system interaction, which are identified as the two major areas where much more technology advances are still needed, if we wish voice-based information retrieval in the future can be as convenient and attractive as text-based information retrieval today. We also present three successful prototype systems as application examples: a broadcast news browser, a course lecture browser, and a personal photo browser.

Below, after a brief introduction of the three different tasks of voiced-based information retrieval and the comparison between voice-based and text-based information retrieval in sections 2 and 3, the issues of retrieval accuracy and user-system interaction are discussed in detail in sections 4 and 5 respectively. The application examples are finally presented in section 6, and the concluding remarks are made in section 7.

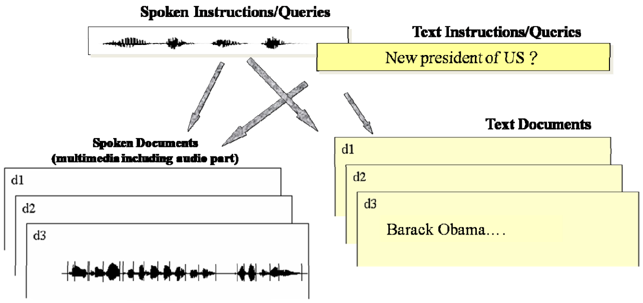

Spoken Instructions/Queries
Text Instructions/Queries
New president of US ?
Text Documents
Spoken Documents
(multimedia including audio part)
d1
d2
d3
d1
d2
d3
Barack Obama….

Fig. 1. Voice-based Information Retrieval: User instructions/queries in spoken form and/or documents to be retrieved in spoken form.

II. Three Different Tasks of Voice-based Information Retrieval

Here we very briefly explain the three different tasks of voice-based information retrieval.

2.1 Using Text Queries to Retrieve Spoken Documents

This has usually been referred to as Spoken Document Retrieval, and has been considered and studied for long. For example, in the last decade in the TREC (Text REtrieval Conference) Spoken Document Retrieval track [8], very good retrieval performance based on ASR one-best results for the spoken documents was obtained as compared to that on human reference transcripts, although using relatively long queries and relatively long target documents [9]. It was then realized that considering much shorter queries and much shorter spoken segments with much poorer recognition accuracies should be a more realistic scenario [3], [10], [11], [12], [13], [14]. For such cases, the problem turned out to be much more difficult and most efforts were concentrated on detecting a certain term in the very short spoken segments, usually referred to as Spoken Term Detection. This task looks similar to the traditional task of keyword spotting, while the major difference is that here the query set is open and very often includes out-of-vocabulary (OOV) words. In such task people have found it necessary to consider the relatively poor recognition accuracies in various ways, such as using lattices to include multiple recognition hypotheses, using confusion matrices or fuzzy matching to consider possible recognition errors, etc. These will be discussed in more details later on.

2.2 Using Spoken Queries to Retrieve Text Documents

This has usually been referred to as Voice Search [15], [16], [17]. It is very possibly the part of voice-based information retrieval closest to realistic applications, and thus has attracted very high attention in recent years. The information to be retrieved is usually an existing text database such as those in directory assistance applications, although with lexical variations and so on but primarily without recognition uncertainty. It is the user query and therefore user intention which is uncertain. A popular approach to handle this problem is via the well developed technologies of spoken dialogues as shown in Fig. 2. The user query is recognized and the output with uncertainty used for search. The uncertainty in user intention and the high degree of lexical variation between the target documents and the user query may lead to many retrieved results, which may be disambiguated by spoken dialogue loops.

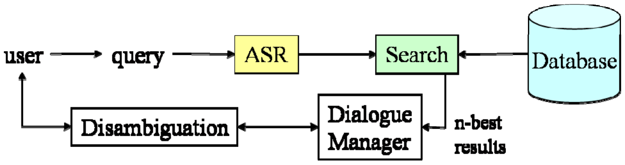


Fig. 2. Voice Search accomplished by spoken dialogue loops

Some people also considered similar problems but in slightly different directions. For example, some represented the queries as more complete lattices, and some performed more semantic analysis during retrieval. In such cases the task may also be referred to as Spoken Query Processing [18], [19].

2.3 Using Spoken Queries to Retrieve Spoken Documents

In this case the speech recognition uncertainly exists on both sides of the queries and the documents, and therefore naturally this is a more difficult task this. Much less work was reported for this task, although some of them was performed very early. In an example effort, the task was considered as a problem of query-by-example [20]. In another example effort, the lattices of the query and the documents were aligned and compared using the graphical model [21]. People also tried to directly match the query and the content on the signal level [22], [23], as other examples. But clearly more work is needed for this task.

III. Comparison between Voice-based and Text-based Information Retrieval

Table 1 lists the comparison between voice-based and text-based information retrieval in terms of three aspects: (1)Resources, (2)Accuracy and (3)User-system Interaction.

TABLE 1 COMPARISON BETWEEN VOICE-BASED AND TEXT-BASED INFORMATION RETRIEVAL

	Text-based	Voice-based
Resources	• Rich resources—huge quantities of text documents available over the Internet • Quantity continues to increase exponentially due to convenient access	• Spoken/multimedia content are the new trend • Can be realized even sooner given mature technologies
Accuracy	• Retrieval accuracy acceptable to users • Retrieved documents properly ranked and filtered	• Problems with speech recognition errors, especially for spontaneous speech under adverse environments
User-System Interaction	• Retrieved documents easily summarized on-screen, thus easily scanned and selected by user • Users may easily select query terms suggested for next iteration retrieval in an interactive process	• Spoken/multimedia documents not easily summarized on-screen, thus difficult to scan and select • Lacks efficient user-system interaction

First consider the resources. Text-based information retrieval is so useful and attractive because huge quantities of text documents are available over the Internet, and the quantity continues to increase exponentially due to the convenient access. For voice-based information retrieval, definitely multimedia and spoken content are the new trend, and such resources as rich as text-based resources can be realized even sooner given mature technologies. So this is not a problem at all.

Next consider the retrieval accuracy. Clearly the accuracy for text-based information is acceptable to users and users even like it very much. In fact, the retrieval engines usually can properly rank and filter the retrieved documents which improve the perceived precision to a good extent. On the other hand, there are still serious problems with the accuracy of voice-based information retrieval, especially for spontaneous speech under adverse environments in queries and/or target documents which give very poor ASR accuracies. In fact, memory and computation requirements for voice-based information retrieval technologies also cause serious problems if a satisfactory accuracy has to be achieved. So the cost for memory and computation requirements is another problem coming together with the accuracy.

Finally, consider the user-system interaction. For text-based information retrieval the retrieved documents are easily summarized on-screen, thus easily scanned and selected by the user. The user can also select query terms suggested by the search engines for next iteration retrieval in an interactive process. Such convenient user-system interaction is actually a very important key which makes text-based information retrieval very attractive. For voice-based information retrieval, however, the situation is completely different. The multimedia/spoken documents are not easily summarized on-screen, thus difficult to scan and select. In fact, an efficient user-system interaction scenario still doesn't exist.

IV. Retrieval Accuracy for Voice-based Information Retrieval

If the recognition of the spoken queries and/or spoken target segments (in the target documents) can be 100% accurate, the voice-based information is naturally reduced to text-based information retrieval. Unfortunately this is never true. Recognition errors are inevitable, and the recognition accuracy is even not predictable or controllable. Many approaches have been considered to handle the recognition errors here. Use of confusion matrices or fuzzy matching techniques to tolerate recognition errors to a certain extent [24], [25], [26], use of lattices rather than 1-best output to consider multiple recognition hypotheses so as to include more correct results, and use of subword units to try to handle the out-of-vocabulary (OOV) words to some degree are good examples. Below we very briefly explain two major approaches here: Lattice-based Approaches and Subword Units.

4.1. Lattice-based Approaches

If all utterances in the spoken segments are represented as lattices with multiple alternatives rather than 1-best output, certainly the probability that the correct words are included and considered can be higher. However, much more noisy words are also included which cause some trouble, although they can be discriminated with some scores such as posterior probabilities, while some important words (e.g. OOV words) may still be missing.

Word/Subword-based lattice information was converted into a weighted finite state machine (WFSM) in an earlier work [7]. The query word/subword sequence was then located in the WFSM using exact-matching. A two-stage approach was used in another work [27]: audio documents were first selected by approximated term frequencies, and then a detailed lattice search was performed to determine the exact locations.

Another important issue to be considered for such lattices is that the memory and computation requirements may become prohibitively huge, especially if we assume all spoken documents in the very large archives need to be represented as lattices. Great efforts were therefore made to reduce such lattices into simplified forms, referred to as indexing structures here. One example is in Fig. 3, in which the simplified indexing structure is a linear sequence of clusters, each of which includes a number of word hypotheses with some scores such as posterior probabilities. In this way the memory and computation requirements can be significantly reduced (but in fact still huge) with the primary indexing functionalities preserved. In fact, in such structures even more possible paths may be generated. For example, in Fig. 3(a) the word W_3 can not be followed by the word W_8, but this becomes possible in Fig. 3(b). N-gram matching can then be performed over such structures of the spoken queries and/or segments. Many such indexing structures have been proposed and shown to be useful, good examples include Position Specific Posterior Lattices (PSPL) [1], Confusion Networks (CN) [3], [12], Time-based Merging for Indexing (TMI) [11], [28], Time-anchored Lattice Expansion (TALE) [28], etc. Below we use PSPL and CN as two illustrative examples.

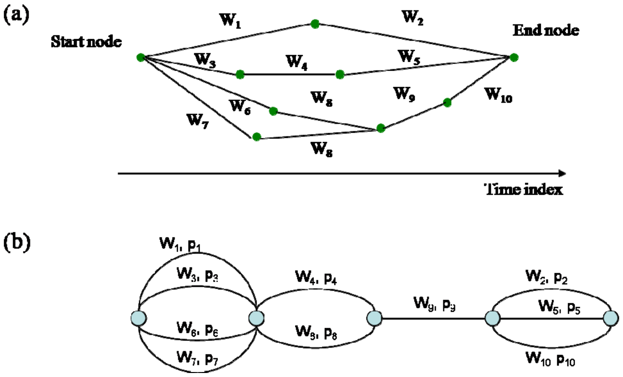


Fig. 3. Lattice-based Indexing Structure (a)An original lattice and (b)an example of the corresponding simplified indexing structure

4.1.1 Position-Specific Posterior Lattices (PSPL)

The basic idea of PSPL is to calculate the posterior probability *prob* of a word *W* at a specific position *pos* (actually the sequence ordering in a path including the word *W*) in a lattice for a spoken segment d as a tuple (*W, d, pos, prob*). Such information is actually hidden in the lattice *L* of *d* since in each path of *L* we clearly know the position (or sequence ordering) of each word. Since it is very likely that more than one path includes the same word in the same position, we need to aggregate over all possible paths in a lattice that include a given word at a given position.

A variation of the standard forward-backward algorithm can be employed for this computation. The forward probability mass $\alpha(W, t)$ accumulated up to a given time t at the last word W needs to be split according to the length l measured in the number of words:

$$\alpha(W,t,l) \doteq \sum_{\substack{\pi:\ \pi \text{ ends at time } t, \text{ has the} \\ \text{last word } W, \text{ and includes } l \\ \text{words}}} P(\pi),$$

where π is a partial path in the lattice. The backward probability β(W, t) retains the original definition [29].

The elementary forward step in the forward pass can now be carried out as follows:

$$\alpha(W,t,l'+1) = \sum_{W'} \sum_{\substack{t':\exists \text{ arc } e \text{ starting at time } t', \\ \text{ending at time } t, \text{ and with} \\ word(e) = W}} [\alpha(W',t',l') \cdot P_{AM}(W) \cdot P_{LM}(W)], \quad (1)$$

where $P_{AM}(W)$ and $P_{LM}(W)$ denote the acoustic and language model scores of W respectively; e is a word arc in the lattice and $word(e)$ means the word entity of arc e.

The position specific posterior probability for the word W being the lth word in the lattice is then:

$$P(W,l|L) = \sum_t \frac{\alpha(W,t,l) \cdot \beta(W,t)}{\beta_{start}} \cdot Adj(W,t), \quad (2)$$

where β_{start} is the sum of all path scores in the lattice, and $Adj(W, t)$ consists of some necessary terms for probability adjustment, such as the removal of the duplicated acoustic model scores on W and the addition of missing language model scores around W [29]. We can regard the tuples (W, d, pos, $prob$) for a specific spoken segment d, a specific position pos but different words W as a *cluster* of words in the indexing structure as mentioned here, which includes a number of words along with their posterior probabilities.

4.1.2 Confusion Network (CN)

This approach was proposed earlier to cluster the word arcs in a lattice into several strictly linear clusters of word alternatives, referred to as the Confusion Network (CN) [30]. In each cluster, posterior probabilities for the word alternatives are also obtained. The original goal of CN was focused on the WER minimization for ASR, since it was shown that this structure gives better expected word accuracy [30], [31]. In the retrieval task here, however, we consider CN as a compact structure representing the original lattice, giving us the proximity information of each word arc [3], [12].

This approach includes a bottom-up clustering algorithm to construct a CN from a lattice. We follow the standard forward-backward algorithm to compute the posterior probability of each word arc as preprocessing before clustering. Each word arc is then regarded as a cluster at the beginning of clustering. Then we run two steps of clustering to produce the final strictly linear *clusters*, the *intra-word clustering* and *inter-word clustering*. After clustering, the posterior probabilities of those word arcs in the same cluster representing the same word W are summed up to be a single posterior probability for a single W in that cluster [30].

4.1.3 Fundamental Distinctions Between PSPL and CN

From the above we may induce several fundamental distinctions between PSPL and CN in terms of the basic principles and structures. They are summarized here.

(a) Basic Construction Principles

The construction of PSPL is based on paths in a lattice. This is clear in Fig. 4(a)(b)(d). We first enumerate all the paths in the lattice, each with its own length (counted in number of words) and path weights as combined language and acoustic model scores. The posterior probability of a given word at a given position is then computed by aggregating all the path weights, where the paths include the given word at the given position, as the numerator and then divided by the sum of all the path weights in the lattice. The algorithm presented in Sec. 4.1.1 is an efficient way to accomplish this. We thus regard the words in each position as a cluster as in Fig. 4(d). It is clear that the reason for the words being in the kth cluster is that there exist some paths carrying those words as the kth word in the paths.

In CN, on the contrary, the construction is based on word arcs instead of paths in the lattice. All word arcs that overlap in time will be clustered together in one or several clusters (while nonoverlapped arcs are never in the same cluster). The basic procedures of intra/inter-word clustering in Sec. 4.1.2 provide a means to ensure that arcs with higher probabilities, more similar pronunciations and/or more overlaps in time will be clustered first. The reason for a word to be in the kth cluster, as in Fig. 4(e), is not as straightforward as that for PSPL. By following the priorities as constrained by the clustering algorithm, those words having similar time spans and usually similar pronunciations are finally clustered together. All the clusters are then sorted by time, and a specific cluster appears to be the kth one. These principles are summarized in Fig. 4(c).

(b) Posterior Probabilities

In PSPL we assign a posterior probability *prob* to a word W in the kth cluster as the ratio of the sum of weights of those paths carrying W as the kth word to the sum of all path weights in the lattice. In CN, the posterior probability *prob* assigned to a word W in the kth cluster represents not only the paths carrying W as the kth word, but also possibly those as the (k−1)th, (k+1)th word and so on, due to the clustering approach of CN. The clustering algorithm tries its best to cluster the word arcs together as long as their time spans overlap, regardless of the exact positions of these word arcs in their respective paths, though sometimes those word arcs appearing in similar time spans also occur in similar positions in their respective paths.

(c) Number of Clusters

The CN gives a rough idea about the number of words in a reasonable recognition result at a global view. For example, if the CN of an utterance has K clusters, very possibly the utterance has around K words. This is quite different from PSPL. If we have K clusters in the PSPL structure, all we can say is that the longest paths (counted in words) in the lattice have K words, thus usually K is much larger than the real number of words.

(d) Coverage and Space Requirement

Each word N-gram appearing in the lattice also appear in n consecutive clusters of PSPL. But this is not necessarily true for CN. As depicted in Fig. 4, while the trigram $W_3W_4W_5$ appearing in the lattice also appears in the PSPL's first to third clusters, we can't find consecutive clusters for it in the CN structure, since W_5 is in the 4th cluster while W_3,W_4 in the first two clusters. This is very possible for CN and implies CN is slightly less complete than PSPL in covering all possible word sequences for indexing purposes.

On the other hand, the same word arc usually duplicate many times in different clusters in PSPL, because the word lengths of different paths usually differ. A word W may appear as several arcs with similar time spans in more than one paths, and in some paths it is the kth word while in others it is the $(k+1)$th or $(k+2)$th. So the word W may simultaneously appear in the k,$(k+1)$th, $(k+2)$th clusters of PSPL. But this rarely happens for CN since the first step in constructing CN is to cluster the word arcs representing the same word with similar time spans together. This also implies for PSPL we need much more space to store the indices than CN. Note that both PSPL and CN generate extra paths than the original lattices [10], [12]. For example in Fig. 4 the word sequences $W_1W_4W_5$ in PSPL and $W_3W_8W_9$ in CN (both from the first to the third cluster) do not appear in the original lattice.

4.1.4 Relevance Ranking of Spoken Segments Given PSPL or CN

Given the strictly linear clusters in word-based PSPL or CN structures as in Fig. 4 for all the spoken segments, assuming a spoken document retrieval task, we may use them to evaluate the relevance scores between the segments and a query Q, which is a sequence of words, $\{W_j , j = 1, 2.., Q\}$ [10]. We first calculate the expected tapered-count for each N-gram $\{Wi...Wi+N-1\}$ within the query in a spoken segment d, $S(d,Wi...Wi+N-1)$ as in Eq (3) below, and aggregate the results to produce a score $S_{N\text{-}gram}$ (d, Q) for each order N as in Eq (4) below[10]:

$$S(d, W_i...W_{i+N-1}) = \log\left[1 + \sum_{k}\prod_{l=0}^{N-1} P(W_{i+l}, k+l|L)\right], \qquad (3)$$

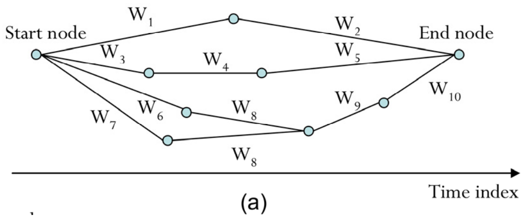
$$S_{N\text{-}gram}(d, Q) = \sum_{i=1}^{Q-N+1} S(d, W_i...W_{i+N-1}), \qquad (4)$$

where L is the lattice obtained from d and k is the cluster number in PSPL or CN structures. The different proximity types, one for each N-gram order allowed by the query length Q, are finally combined by a weighted sum to give the final relevance score $S(d, Q)$,

$$S(d, Q) = \frac{\sum_{N=1}^{Q} t_N \cdot S_{N\text{-}gram}(d, Q)}{\sum_{N=1}^{Q} t_N}, \qquad (5)$$

where different weights t_N may be possible.

Lattice:



(a)

All paths:

W_1W_2, $W_3W_4W_5$, $W_6W_8W_9W_{10}$, $W_7W_8W_9W_{10}$

(b)

PSPL:

Locating a word in a segment according to the position (or sequence ordering) of the word in a path

CN:

Clustering several words in a segment according to similar time spans and word pronunciation

(c)

PSPL structure:

W_1: p_1 W_3: p_3 W_6: p_6 W_7: p_7	W_2: p_2 W_4: p_4 W_8: p_8	W_5: p_5 W_9: p_9	W_{10}: p_{10}
Cluster 1	Cluster 2	Cluster 3	Cluster 4

(d)

CN structure:

W_1: p_1 W_3: p_3 W_6: p_6 W_7: p_7	W_4: p_4 W_8: p_8	W_9: p_9	W_2: p_2 W_5: p_5 W_{10}: p_{10}
Cluster 1	Cluster 2	Cluster 3	Cluster 4

(e)

Fig. 4. (a) The ASR lattice, (b) all paths in (a), (c) basic principles in constructing PSPL and CN, (d) the constructed PSPL structure, (e) the constructed CN structure, where W_1, W_2, … are words and by W_1:p_1 we mean W_1 and its posterior probability p_1 in a specific cluster.

4.2 Subword Units

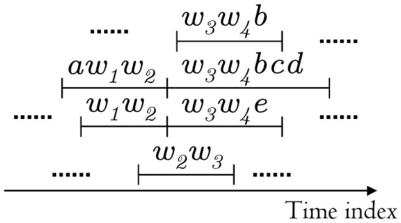


Fig. 5. A partial lattice with several word arcs denoted by their constituent subword units.

With PSPL and CN as discussed above, we can index the *soft-hits* for each word in the lattice as a tuple: (W, d, pos, $prob$) [1], [3]. Then for the query Q composed of several words $\{W_j, j = 1..., Q\}$, we may check the soft-hits for each of these words W_j and find out the relevant documents ranked by their similarities with the query considering the posterior probabilities and the proximity information, which is much more powerful than the conventional approach based on one-best search. However, PSPL or CN is not able to handle queries with rare or OOV words. The lower N-gram probabilities of rare words or the absence in the lexicon for OOV words simply makes it impossible for these words to appear in the lattice. This is an important issue because rare or OOV words are very often the keywords used in queries, because people usually care about new events rather than those that are well known [32].

Consider an example. Assuming a spoken document *d* contains a rare or OOV word *W* with the subword units $\{w_1w_2w_3w_4\}$; the ASR lattice for *d* is shown in Fig. 5. The word *W* never appears as a word arc in the lattice, but is replaced by many other words including similar subword units such as w_3w_4b, aw_1w_2, for instance, where *a*, *b*, *c*, *d*, . . . are other subword units. With PSPL or CN constructed based on words the *soft-hits* for the word *W* do not include *d* due to the absense of the word arc *W* in the lattice of *d*. If the PSPL or CN are constructed based on subword units, on the other hand, indexing can be based on the subword units. Each subword unit has its *soft-hits* also as the tuples mentioned above. Thus for a query Q containing the same rare/OOV word *W*, we simply decompose it into subword units $\{...w_1w_2w_3w_4...\}$ and find hits for *d* in the sequences w_1w_2, $w_2w_3w_4$, $w_1w_2w_3w_4$, and so on. As a result, with PSPL and CN constructed based on subword units, *d* has a fair rank under this query and can be retrieved accordingly, even if *W* is not in the lexicon.

The above leads to the concept of constructing lattice-based indexing structures using subword units [33]. Note that in order to do that we need to be able to estimate the posterior probabilities for subword units. The estimation of posterior probabilities for such subword units is not trivial, but several different points of view have been proposed for an efficient approximation [33, 34, 35]. One of them is briefly summarized below in section 4.2.1.

4.2.1 Subword Posterior Probability

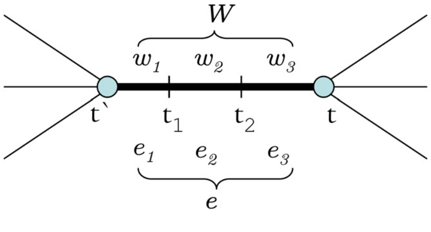


Fig. 6. A word edge *W* with subword units $w_1w_2w_3$ starting at time *t'* and ending at time *t*.

Consider a word *W* with subword units $w_1w_2w_3$ corresponding to an edge *e* starting at time *t'* and ending at time *t* in a word lattice as shown in Fig. 6. During ASR we may record the boundaries between w_1, w_2, and w_3, which are t_1 and t_2. Following the previously proposed approach [29], we may calculate the posterior probability of the edge e given the ASR lattice *L*, *P*(*e*|*L*), as:

$$P(e|L) = \frac{\alpha(t') \cdot P(x_{t'}^{t}|W) \cdot P_{LM}(W) \cdot \beta(t)}{\beta_{start}}, \qquad (6)$$

where $\alpha(t')$ and $\beta(t)$ denotes the forward and backward probability mass accumulation up to time *t'* and t as in the standard forward-backward algorithm. β_{start} is the same as in Eq. (2). To extend the same approach to compute the posterior probability of a subword unit of *W*, say w_1, we may write $P(e_1|L)$ as:

$$P(e_1|L) = \frac{\alpha(t_1) \cdot P(x_{t'}^{t_1}|w_1) \cdot P_{LM}(w_1) \cdot \beta(t_1)}{\beta_{start}}. \qquad (7)$$

Here we have two new values to be estimated, $P_{LM}(w_1)$ and $\beta(t_1)$.

It may be possible to train a new language model which mix words and subwords for estimating $P_{LM}(w_1)$. However, it was shown [36] that subword-based language model has less predicting ability than word-based one, and the way to use subwords and words in a single language model is not clear. The value of $\beta(t_1)$ is even more difficult to estimate. We simply don't have the node corresponding to t_1 in the word lattice, and even if we specially generate a node for t_1, the transitions at this new node is not as free as the other nodes due to the lexicon constraints.

Here we made some simplifications and assumptions to have effective and easy estimations of $P_{LM}(w_1)$ and $\beta(t_1)$. First, we assume $P_{LM}(W) \approx P_{LM}(w_1)$. Of course this is a very rough assumption and we know that $P_{LM}(W) \leq P_{LM}(w_1)$ since the event of w_1 for some history includes the event of W for the history. Secondly, we assume that w_1 has only one path to go from t_1 to time *t*, via w_2 and w_3. Of course there is at least one path to go from t_1 to *t* via w_2 and w_3, but by making it "the only one" we may rewrite $\beta(t_1)$ as $\beta(t_1)= P\left(x_{t_1}^{t}\middle|w_1w_2\right)\cdot\beta(t)$. We can now substitute $P_{LM}(W)$ and $P\left(x_{t_1}^{t}\middle|w_1w_2\right)\cdot\beta(t)$ for $P_{LM}(w_1)$ and $\beta(t_1)$ in Eq.(4). Now the result is very simple and we have $P(e_1|L) \approx P(e|L)$. Similar assumptions can be made on the subword edges e_2 and e_3 and we can have $P(e_2|L) \approx P(e_3|L) \approx P(e|L)$.

4.2.2 Subword-based Position Specific Posterior Lattices (S-PSPL)

With the posterior probability for subword units properly estimated, we are now able to construct the Subword-based Position Specific Posterior Lattices (S-PSPL). Similar to PSPL, we begin with the computation for the position specific probabilities for words, except here the position is based on subword units. Similar to those in Sec. 4.1.1, with a variation of the standard forward-backward algorithm, the forward probability mass $\alpha(W, t)$ accumulated up to a given time t with the last word being *W* needs to be split according to the length *l*, measured in number of subword units instead of words:

$$\alpha(W,t,l) \doteq \sum_{\substack{\pi\text{: a partial path ends at time } t,\\ \text{has last word } W\text{, and includes}\\ l \text{ subword units}}} P(\pi).$$

The backward probability $\beta(W, t)$ retains the original definition [29].

The elementary forward step is very similar to Eq. (1) where $l = l' + Sub(W)$; $Sub(W)$ is the number of subword units in W. $P_{AM}(W)$ and *PLM*(*W*) are the same as in Eq. (1).

$$\alpha(W,t,l) = \sum_{W'} \sum_{\substack{t':\exists \text{ edge } e \\ \text{starting at} \\ \text{time } t'\text{, end-} \\ \text{ing at time} \\ t\text{, and with} \\ word(e) = W}} \left[\alpha(W',t',l') \cdot P_{AM}(W) \cdot P_{LM}(W)\right], \quad (8)$$

On the other hand, the position specific posterior probability for the word W being the b^{th} to the $(b+Sub(W) - 1)^{\text{th}}$ subword units in the lattice is very similar to Eq. (2):

$$P(W,b,b+Sub(W)-1|L) = \sum_{t} \frac{\alpha(W,t,b+Sub(W)-1)\cdot\beta(W,t)}{\beta_{start}} \cdot Adj(W,t), \quad (9)$$

where $Adj(W, t)$ and β_{start} are the same as Eq. (2). Following the assumptions made in the above, the probability of a subword w being the k^{th} subword unit in the lattice is then simply the sum of the position specific posterior probabilities for the appropriate words W:

$$P(w,k|L) = \sum_{\substack{W,b\text{: } w \text{ is the } r^{\text{th}} \\ \text{subword in } W \text{ and} \\ b+r-1=k}} P(W,b,b+Sub(W)-1|L). \quad (10)$$

Note that it is possible to recognize subword units directly and produce subword-based lattices, from which PSPL for subword units can be constructed for indexing. But this approach does not in fact yield satisfactory results, since the recognition accuracy of plain subword units is generally much worse than that of words [7], [36]. On the other hand, directly converting the word-level representation of a lattice into a subword-level lattice is easy, but that representation can include only subword unit strings which are substrings of in-vocabulary word string pronunciations [7].

But with the approaches presented here, we start with word lattices, breaking word arcs into subword unit arcs, estimating their posterior probabilities, and then follow exactly the same way of constructing PSPL to construct S-PSPL as proposed here. In this way we can keep the high accuracy of word-based recognition, while at the same time use subword units to handle OOV words. In S-PSPL as proposed here, by recording only the position information and posterior probabilities for subword units, strings of subword units are not constrained by in-vocabulary words any longer.

4.2.3 Subword-based Confusion Network (S-CN)

It is straightforward to construct a subword-based CN (S-CN) given the approximations in section 4.2.1 During ASR, we may record the start and end time for the subword units in each word arc. We then follow section 4.2.1 to assign the posterior probabilities for subword units. We then regard these subword units as subword arcs and run the clustering algorithm as we do for original CN to construct S-CN. In each cluster of S-CN, we also sum up the posterior probabilities of subword arcs representing the same subword unit, as we do for CN.

4.2.4 Relevance Ranking of Spoken Segments Given S-PSPL or S-CN

For S-PSPL and S-CN, the procedures in section 4.1.4 remain unchanged, except now we decompose Q into a sequence of subword units instead and the allowed N-gram of Q is also based on subword units.

4.2.5 Examples of Frequently Used Subword Units

Different choices of subword units have been used by different research groups in many different works. It seems the choice of subword units has to do with the characteristics of the specific languages. For example, phonemes (or phoneme N-grams in similar approaches) have been popularly used for many alphabetic languages such as English [37], [38]. Graphemes [39] and graphones [40], [41] were also used in some works. People also used word fragments, or sometime referred to as particles, which are groups of phonemes very often appear together and can be derived by data-driven approaches [42], [43]. Morphs or morph-like units [44], [45], on the other hand, have been found very useful for morph-based languages such as Turkish and Finnish. On the other hand, Mandarin Chinese is monosyllable-based, for which there is a special mapping relation between monosyllables and characters. It has been found very early that syllables, characters (and N-grams of them in similar approaches) are very useful subword units for Mandarin Chinese [46], [47], [33].

4.3 Some Example Test Results

Here we very briefly summarize some example test results to see the performance of the various approaches mentioned above [47].

4.3.1 Experimental Setup

The corpora used in the experiments to be retrieved are Mandarin broadcast news stories collected daily from local radio stations in Taiwan from August to September 2001. We manually divided segmented these stories into 5034 segments, each with one to three utterances. We used the TTK decoder [48] developed at National Taiwan University to generate the bigram lattices for these segments. From the bigram lattices, we generated the corresponding word-based PSPL/CN and S-PSPL/S-CN structures, with which we recorded the tuple (segment id, position, posterior probability) for each word (subword) unit in the respective segment's lattice.

By altering the beam width in generating the bigram lattice, we obtained different lattice depths and sizes and in turn word-based PSPL/CN and S-PSPL/S-CN of different sizes were generated. Four lattices — L_1, L_2, L_3 and L_4 —were created, each with averaged 19.89, 30.27, 46.75, and 72.77 edges per spoken word respectively. The disk size needed to store the four lattices was 19.2, 29.1, 44.5, 69.3MB respectively.

A trigram language model estimated from a 40M news corpus collected in 1999 was used. The lexicon of the decoder consisted of 62K words selected automatically from the above training data based on the PAT tree algorithm [49]. The acoustic models included a total of 151 right-context-dependent intra-syllable Initial-Final (IF) models, trained using 8 hrs of broadcast news stories collected in 2000. The recognition character accuracy obtained for the 5034 segments was 75.27% (under trigram one-pass decoding). As the corpus was in Mandarin Chinese, the subword units used in S-PSPL and S-CN were characters and syllables.

159 text test queries were generated by manual selection from a set of automatically generated candidates, each including 1 to 3 words. The candidates were high-frequency N-grams with length 1 to 3 words which appeared at least 8 times in the 5034 segments. 39 of the 159 queries included OOV words and were thus categorized as OOV queries, while the remaining 120 were in-vocabulary (IV) queries.

Word-based PSPL/CN and S-PSPL/S-CN resulted in a total of 6 experiments: (a)(b) word-based PSPL and CN; (c)(d) character-based S-PSPL and S-CN; and (e)(f) syllable-based S-PSPL and S-CN.

4.3.2 Test Results

The results are plotted in Fig. 7 where memory size and MAP (Mean Average Precision, computed by the standard trec_eval package used by the TREC evaluations, evaluated for all queries, IV plus OOV) are the two dimensions, which demonstrates clear tradeoffs between index size and retrieval accuracies. We have six curves for the six approaches (a)-(f) considered to show this tradeoff. Each curve has 4 points, representing the results for the 4 lattice sizes, L_1, L_2, L_3 and L_4. The distinctions between PSPL and CN discussed in section 4.1.3 can be verified here. For example, as mentioned in section 4.1.3 that CN is less complete in indexing as compared to PSPL, but PSPL uses much more memory space.

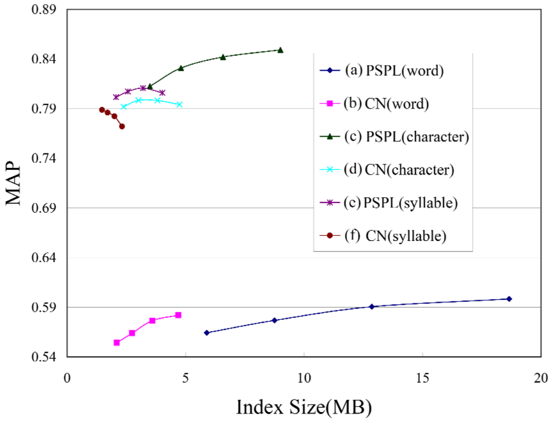


Fig. 7. The tradeoff between MAP and index size for the different approaches considered [47].

In principle those approaches at the upper left corner of Fig. 7 are more attractive, because higher MAP is obtained at smaller index size. So we see subword-based approaches use less memory space but achieves much higher accuracy, for example S-PSPL looks quite attractive. More detailed discussions about the data here are also available [47].

V. User-System Interaction for Voice based Information Retrieval

As mentioned above, for text-based information retrieval the retrieved documents can be easily summarized on the screen, thus easily scanned and selected by the user. Very often the user can also select some query terms suggested by the search engine for next iteration retrieval in an iterative process. In other words, interactive or dialogue functionalities are not only popularly used for text-based information retrieval, but an important reason why the text-based information retrieval is useful and attractive.

For voice-based information retrieval, on the other hand, we don't have such interactive or dialogue scenario yet. Unlike the written documents with well structured paragraphs and titles, the multimedia and spoken documents are both very difficult or browse, since they are just audio/video signals, very difficult to be shown on the screen, and the user can not go through each of them from the beginning to the end during browsing.

A few examples explaining the need of interactive or dialogue functionalities in voice-based information retrieval are listed here. First, the user's query during retrieval is usually very short which inevitably includes ambiguity and therefore results in too many outputs. For example, given the query "*George Bush*", the user may be interested in the *Iraq* or *China* issue. The system can tell the difference only with following-up interactions. Second, OOV word problem is often handled by subword-based indexing and retrieval techniques as mentioned previously, but such techniques also naturally lead to many irrelevant retrieved documents and thus low precision. Following-up interactions or dialogues are therefore very helpful for the user to identify and select the desired information. Third, the gap between the system and the user in such scenarios is usually huge. It is difficult for the user to formulate his queries precisely describing his information needs to retrieve efficiently. The system also needs a good mechanism to probe the user's needs. As a result, a series of follow-up questions and interactions is certainly very helpful. Dialogues are certainly a good solution to such problems [50]. This is why use-system interaction is an important issue here.

5.1 Proposed Approaches

In order to address this issue, one possible solution is to use multi-modal dialogues to help the user to "navigate" across spoken documents archives and find the desired spoken documents. In this concept, for a query given by the user, the retrieval system produces a topic hierarchy constructed from the retrieved spoken documents to be shown on the screen. Each node on the hierarchy represents a cluster of retrieved documents and is labeled by a key term, or a topic. The spoken documents can also be presented in forms of automatically generated summaries or titles in addition. The user can then expand his query easily by choosing or deleting the key terms within the topic hierarchy by a simple click or a second spoken query to specify more clearly what he is looking for, while at the same time browse through the titles and summaries (if available) when needed. This is a multi-modal dialogue process because the system response is in form of a topic hierarchy displayed on the screen, and the user input may be given by clicks or spoken queries. With a few dialogue turns, the small set of spoken documents desired by the user can be found by a more specific query precisely expanded during the dialogue process. This is the way the system guides the user to "navigate" across the spoken archives to find the desired documents, as shown in Fig. 8. [51]. In such approaches, the following key elements seems to be needed: information extraction (to extract key information such as key terms or topics, titles and summaries from the spoken documents), document structuring (to organize the set of retrieved spoken documents into some form of hierarchical structures) and query-based (able to respond to the series of user queries to offer the information about system output for the request) [6]. A set of technologies has been proposed for such purposes and is referred to as multimedia or spoken document understanding and organization [5], with which it is possible to construct a convenient user-system interaction interface for the purpose here. This will be briefly explained below.

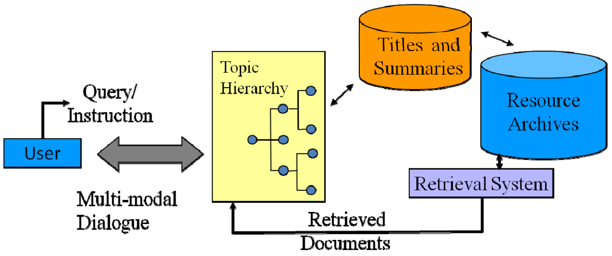


Fig. 8. The multi-modal dialogue scenario for convenient user-system interaction.

5.2 Semantic Analysis of Spoken Documents

Today many machine learning tools have been available for semantic analysis of documents. Most of them can be used for spoken documents, but we have to bear in mind that spoken documents include inevitable ASR errors. Here we very briefly summarize the Probabilistic Latent Semantic Analysis (PLSA) [52] as an example.

Latent Semantic Analysis (LSA) has been widely used in analyzing the content of documents by exploring the relationships between a set of terms and a corpus of documents considering a set of latent topics. In recent years, efforts have been made to establish a probabilistic framework for the above latent topical approaches, including improved model training algorithms, of which PLSA or aspect model [52] is a popularly used example. In PLSA, a set of latent topic variables is defined, T_k, $k = 1, 2, \ldots, K$, to characterize the "term-document" co-occurrence relationships, as shown in Figure 9. Both the document d_i and a term t_j are assumed to be independently conditioned on an associated latent topic T_k. The conditional probability of a document d_i generating a term t_j thus can be parameterized by

$$P(t_j|d_i) = \sum_{k=1}^{K} P(t_j|T_k)P(T_k|d_i). \qquad (11)$$

Notice that this probability is not obtained directly from the frequency of the term t_j occurring in d_i, but instead through $P(t_j\,|T_k)$, the probability that the term t_j is used in the latent topic T_k, as well as $P(T_k|d_i)$, the likelihood that d_i addresses the latent topic T_k. The PLSA model can be optimized with EM algorithm by maximizing a carefully defined likelihood function [52].

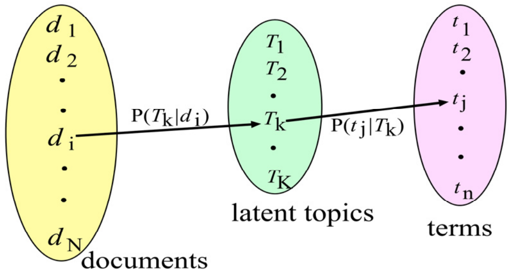


Fig. 9. Graphical representation of the Probabilistic Latent Semantic Analysis (PLSA) model.

5.3 Key Term Extraction from Spoken Documents

Key terms have long been used to identify the semantic content of documents. The only difference here is that the key terms need to be extracted automatically from spoken documents which are dynamically generated and updated from time to time. In fact, key terms have also been found useful in constructing retrieval models. The retrieval can naturally become more precise if the key terms in queries/documents can be known [53]. In addition to many other approaches being used for key term extraction, PLSA mentioned above offers very useful parameters for key term extraction. Two example parameters are briefly explained below [54].

(a) Latent Topic Significance

The latent topic significance score of a term t_j with respect to a topic $T_k, S_{t_j}(T_k)$, is defined as:

$$S_{t_j}(T_k) = \frac{\sum\limits_{d_i \in D} n(t_j, d_i) \times P(T_k|d_i)}{\sum\limits_{d_i \in D} n(t_j, d_i) \times [1 - P(T_k|d_i)]} \qquad (12)$$

where $n(t_j,\ d_i)$ is the occurrence count of the term t_j in a document d_i, and $P(T_k|d_i)$ is obtained from a PLSA model trained with a large corpus. In equation (12) the term frequency of t_j in a document d_i, $n(t_j, d_i)$, is further weighted by a ratio which has to do with how the document d_i is focused on the topic T_k, since the denominator of the ratio is the probabilities that the document d_i is addressing all other topics different from T_k. After summation over all documents d_i, a higher $S_{t_j}(T_k)$ obtained in equation (12) implies the term t_j has a higher frequency in the latent topics T_k than other latent topics, and is thus more important in the latent topic T_k.

(b) Latent Topic Entropy

The latent topic entropy of a term t_j is defined as

$$H(t_j) = -\sum_{k=1}^{K} P(T_k|t_j) log P(T_k|t_j) \tag{13}$$

Apparently higher entropy here implies the term is frequently observed in many different latent topics, or is less specific semantically. Lower entropy, on the other hand, indicates that the term is focused on very few latent topics, and thus possibly is a key term for these few latent topics.

5.4 Automatic Generation of Summaries and Titles for Spoken Documents

Automatic summarization of text or spoken documents has been actively investigated for long time [55]. Many approaches for automatic summarization of documents, among others, have attempted to select a number of indicative sentences or passages from the original document according to a target summarization ratio, and sequence them to form a summary. Different approaches have been used to identify sentences carrying concepts closer to the complete documents [56]. The spoken documents bring extra difficulties such as the recognition errors, problems with spontaneous speech, and lack of correct sentence or paragraph boundaries. In order to avoid the redundant or incorrect parts while selecting the important and correct information in spoken documents, multiple recognition hypotheses, confidence scores, acoustic and language model scores and other forms of grammatical knowledge have been utilized [54], [57], [58], [59], [60], [61], [62], [63]. In recent years, a general approach have been found to be very successful [57], in which each sentence in the document, $S = t_1 t_2 \ldots t_j \ldots t_n$, represented as a sequence of terms t_j , is given a score:

$$\begin{aligned} I(S) =& \frac{1}{n}\sum_{j=1}^{n}[\lambda_1 s(t_j) + \lambda_2 l(t_j) + \\ & \lambda_3 c(t_j) + \lambda_4 g(t_j)] + \lambda_5 b(S), \end{aligned} \tag{14}$$

where some statistical measure $s(t_j)$ (such as TF/IDF or the like) and linguistic measure $l(t_j)$ (e.g., named entities and different parts-of-speech (POSs) are given different weights, function words not included) are defined for each term t_j . $c(t_j)$ and $g(t_j)$ are calculated from the confidence score and N-gram score for the term t_j , $b(S)$ is calculated from the grammatical structure of the sentence S, and λ_1, λ_2, λ_3, λ_4 and λ_5 are weighting parameters. In this framework, a Significance Score specially selected for this purpose for $s(t_j)$ in equation (14) has been found specially useful [57], while other parameters such as the Latent Topic Significance and Latent Topic Entropy in equations (12)(13) were also successfully used [54].

The titles exactly complement the summaries for the user during browsing and retrieval. The user can easily select the desired document with a glance at the list of titles. He can then looks through or listen to the summaries in text or speech form for the titles he selected. Automatic generation of titles for spoken documents was less reported as compared to automatic generation of summaries, probably because the task is even more difficult. A title has to be very brief and readable, in addition to being able to tell the key information of the document [64], [65], [66], [67], [68]. An example work reported recently is briefly summarized below [69]. It includes two parts: the training part and the testing part, as shown in Fig. 10. In the training part, three sets of carefully designed models are trained with a training corpus of text documents with human-generated titles: term selection model tells the most suitable terms to be included in the title, term ordering model gives the best ordering of the terms to make the title readable, and title length model tells the reasonable length of the title. In the testing part, the input testing documents are first transcribed into texts with errors using ASR techniques, and then text summaries are obtained. In this way the least important utterances can be removed and important terms can be better collected and used to construct the title. A delicate Viterbi algorithm is then performed on the summaries with scores obtained from the above three sets of models, which gives the output title.

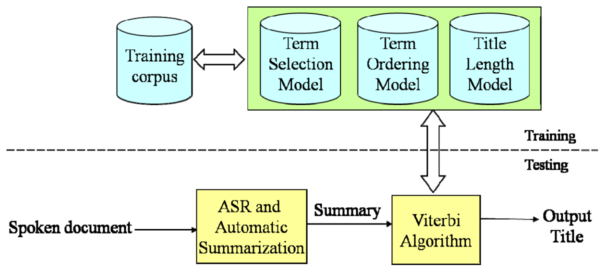


Fig. 10. The block diagram of the recently proposed approach of automatically generating titles for spoken documents

5.5 Semantic Clustering and Structuring of Spoken Documents

As mentioned above, multimedia or spoken documents are difficult to show on the screen and difficult to browse. It is therefore a reasonable approach to try to cluster them into some structures based on the semantic concepts or topics of the documents, so as to help the user to "navigate" across the retrieved documents as well as the entire document archive to find out what he needs. This may include two different parts: query-based local semantic structuring and global semantic structuring.

5.5.1 Query-based Local Semantic Structuring

For each query entered by the user, a Topic Hierarchy can be constructed from the many spoken documents retrieved to be shown on the screen for the user, on which each node represents a set of spoken documents with similar semantic concepts and is labeled by a key term or a topic. This topic hierarchy offers a user interface for multi-modal dialogue between the user and the system. This is shown in Fig. 8 above, and the construction of the topic hierarchy is referred to as query-based local semantic structuring [6, 51].

The topic hierarchy can be constructed in various ways. A most straightforward approach is to develop a feature vector V_d for each retrieved spoken document d in terms of the key terms it includes, and then build a feature vector v_t for each key term t by averaging all those document feature vectors V_d for documents including the key term t weighted by the term frequencies. The Hierarchical Agglomerative Clustering and Partitioning (HAC+P) algorithm [70] can then be performed on-line in real time using these feature vectors v_t to cluster all the key terms into a balanced hierarchy. Every node on the hierarchy is therefore a key term or a topic, which actually includes all retrieved documents including this key term. This HAC+P algorithm consists of two phases: an HAC-based clustering to construct a binary-tree hierarchy and a partitioning (P) algorithm to transform the binary-tree hierarchy to a balanced and comprehensive m-ary hierarchy, where m can be different integers at different splitting nodes. An example partial list of such a topic hierarchy obtained for an archive of broadcast news is shown Figure 11.

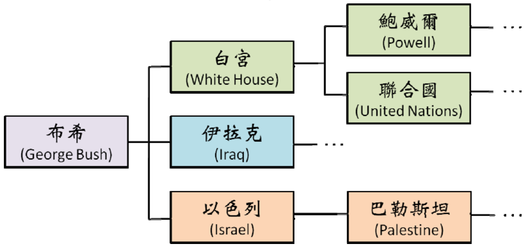


Fig. 11. An example topic hierarchy constructed for the retrieved broadcast news stories obtained with the query: "US and Middle East"

The first phase of HAC algorithm is based on the similarity between two clusters C_i and C_j of key terms, $S(C_i, C_j)$,

$$S(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{v_t \in C_i} \sum_{v_s \in C_j} c(v_t, v_s), \qquad (15)$$

where $c(v_t, v_s)$ is the cosine measure of the angle between the vectors v_t and v_s for key terms t and s. The HAC algorithm is performed bottom-up. Assume there are n key terms in the retrieved documents, the initial clusters, C_1, C_2, ...,C_n, are exactly the n key terms. Let C_{n+i} be the new cluster created at the *i-th* step by merging two clusters. The output binary tree can be expressed as a list, C_1, ...,C_n, C_{n+1}, ...,C_{2n-1}. An example is in Figure 12(a), where n = 5, and C_6, ...,C_9 are created by HAC [51], [70].

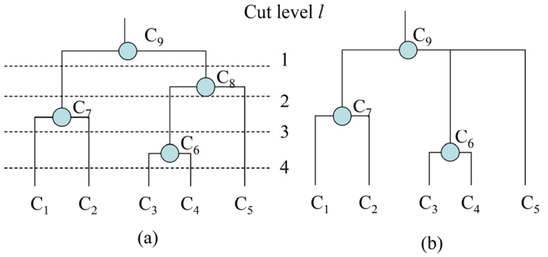


Fig. 12. An illustrative example for the HAC+P algorithm

The second phase of partitioning is top-down. The binary tree is partitioned into several sub-hierarchies first, and then this procedure is applied recursively to each sub-hierarchy. The point is that in each partitioning procedure the best level at which the binary-tree hierarchy should be cut in order to create the best set of sub-hierarchies has to be determined based on the balance of two parameters: the cluster set quality and a preferred number of splitting branches obtained at the level chosen. As shown in Figure 12(a), partitioning can be performed on 4 possible levels by a cut through the binary tree, l = 1, 2, 3, and 4. If a cut is performed at the level l = 2, the result will be three sub-hierarchies, C_5, C_6, and C_7 as shown in Figure 12(b) [51], [70].

5.5.2 Global Semantic Structuring

In addition to the topic hierarchy as mentioned above, sometimes it will be very helpful to have a semantic structure constructed for a wider coverage of the documents even before a query is entered, telling the user the global semantic structure of the document archive to help the user browse or navigate across the archive. This is referred to as global semantic structuring here. One way to achieve this purpose is to use PLSA to analyze the topics T_k for the archive (or a subset of it), and then cluster the documents d_i using the probabilities $P(T_k|d_i)$. These clusters can be labeled by a set of key terms with highest scores for documents in the clusters, and organized in a two-dimensional tree structure, or a multi-layered map, as shown in Fig. 13, for the convenience of the user. In this multi-layer map, documents addressing similar topics are grouped in the same cluster. On each layer, distance between clusters on the map has to do with the relationships between the topics for the documents. Also, a cluster with many documents can be expanded into another map in the next layer [71].

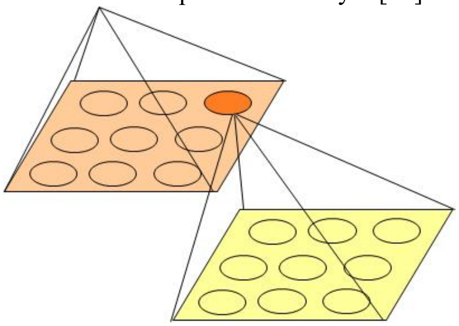


Fig. 13. The two-dimensional tree structure or multi-layered map for global semantic structuring

5.6 Interactive Retrieval in a Dialogue Loop

With all the supporting materials as presented above in sections 5.1-5.5, it is now possible to construct the user-system interaction interface for voice-based information retrieval as a dialogue loop [50], [72], [73], [74], very similar to the very successful spoken dialogue systems developed by many research groups in recent decades.

5.6.1 Dialogue Systems in Voice-based Information Retrieval

Clearly we wish to learn all successful approaches in the very successful spoken dialogue systems [75], [76], [77], [78], [79], [80]. However, we have to realize the dialogue systems we are considering here are somewhat different. The distinct feature for the dialogue systems discussed here is that instead of with a well-organized or relational database at the back-end, the knowledge source to be explored by the dialogue process is usually an unstructured archive of documents, in either text, speech or multimedia form. Under such environment, we are actually faced with many new challenges. First, without a well-organized database at the back-end, the goal of Spoken Language Understanding (SLU) becomes difficult to define. Semantic slots and frames may still be useful, but they are not transformed into an SQL query for a relational database. Second, a much wider spectrum with unknown scope and scale of the back-end knowledge source also implies a much higher degree of variations in the user input utterances, which usually include very short queries, homonyms words, polysemous words, and even OOV words. Different from the mainstream spoken dialogue systems, for which the SLU component relies heavily on the ASR output words, here the user's intention may be difficult to correctly identify even with correct ASR output words (e.g. very short queries or polysemous words), not mentioning that ASR can never recognize OOV words. Third, different from the mainstream spoken dialogue systems in which the user is usually very clear about what kind of information is available and accessible; here the user is usually not aware of the content and structure of the back-end knowledge source. As a result efficient interaction and proper guidance by the system during the dialogue process become necessary. Finally, as has been mentioned repeatedly, the back-end knowledge very often includes multimedia or spoken documents. Therefore the system outputs are usually difficult to be explained in short speech utterances or shown on the screen, and difficult to be browsed by the user, and the problem becomes even worse when the user tries to access the information via small hand-held clients with very small screen. Therefore the system output presented to the user in a more compact, comprehensive, and structural way becomes an important requirement for efficient interaction. This is why we need the techniques mentioned in sections 5.1-5.5 [50].

The block diagram of the proposed dialogue loop for voice-based information retrieval is shown in Fig. 14 [50]. This block diagram is for spoken document retrieval by spoken queries. Some modification is needed for other tasks. There are three major building blocks in Fig. 14: spoken language based information access (primarily using the techniques presented in section 4), dialogue modeling, and multi-modal user interface. The block of multi-modal user interface may includes the presentation of system output in terms of key terms, titles and summaries, and the topic hierarchy discussed above in sections 5.1-5.5. The block of dialogue modeling will be discussed below.

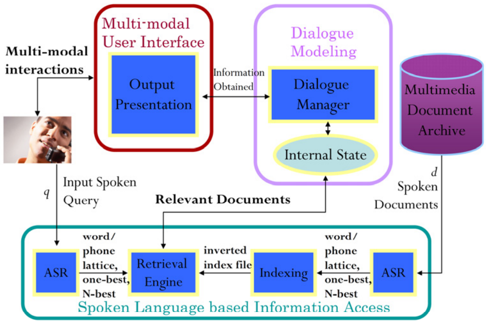


Fig. 14. Structure of a type-II dialogue system

5.6.2 Dialogue Modeling for Voice-based Information Retrieval

Because not too much work on this dialogue modeling has been reported, probably because the concept of considering the necessary interactions between the user and the system for voice-based information retrieval purposes as a dialogue loop is still new. Below, we present one example approach for dialogue modeling with such purposes based on Markov Decision Process (MDP) [81].

The approach is based on the assumption that a topic hierarchy with nodes labeled by key terms or topics is constructed for system output, as mentioned above. At the early stage of the dialogue, because the user doesn't know what can be found from the back-end archive and how to enter the query efficiently, very often he only enters very short queries. With such very short queries, the retrieved documents can be many, a large number of key terms can be extracted, and as a result the topic hierarchy constructed can be very large. The purpose of dialogue modeling here is therefore to rank the key terms before constructing the topic hierarchy. The goal of ranking here is to minimize the number of key terms the user needs to enter before his information needs are satisfied, assuming he chooses the first key term on the topic hierarchy from the top which is relevant to his needs. In this way, the key terms ranked the highest will appear on the top of the constructed topic hierarchy, so the user may spend only minimum time in navigating across the hierarchy, and the system may use only limited space in the screen of hand-held clients to show the most important topics first. This is the basic scenario for dialogue modeling discussed here.

First of all, we define an internal state S_i for the dialogue as the *And*-combinations of all the query terms the user has entered from the beginning, and the machine action A_m as the change in the internal state when an extra query term is entered by the user to further expand the query. For example, in the state $S_2 = [t_i, t_j]$ (t_i, t_j are two query terms entered), if a new term t_k is entered, this automatically leads to a new state $S_5 = [t_i, t_j, t_k]$. The goal of dialogue modeling here is to minimize the number of query terms a user has to enter before his information needs are satisfied, very similar to minimizing the number of dialogue turns in mainstream spoken dialogue systems. We thus define the total reward R_0, to be maximized in the MDP framework, as the above number of query terms the user has to enter. But the latter number should be minimized rather than maximized, or the total reward R_0 should be actually negative of the above number. We therefore define the reward function, $r(S_i, A_m)$, as negative one if the action A_m leads the state S_i to a new state S_j and the documents retrieved by S_j doesn't satisfy the user, and zero otherwise.

With the above definitions we can see that the reward function is determined by each specific user rather than a predefined function. The learning process can then be represented as a state transition tree structure as shown in Fig. 15, in which each node is an internal state, or a series of query terms entered. The tree in Fig. 15 is for a specific user, in which the leaf nodes represented by double circles are those states where the user is satisfied. Each of these leaf nodes are labeled by a score $m(\cdot)$, which is the negative of the number of the query terms successively entered in order to arrive at the state, or the total reward R_0 to be maximized. We then give the score u to each intermediate state as shown in Fig. 15, which is the maximum score $m(\cdot)$ for all child leaf nodes of the intermediate state, $u = \max_i[m(S_i)]$, where the maximization is performed over all child leaf nodes of the state. Such a learning process can be performed with a huge number of training simulated users to obtain the dynamics of the reward function and a balanced view of how efficient a query term entered at each state can satisfy the user. The scores u for all the states averaged over a huge number of training users is then used to rank the key terms. The query term ranking and the internal states then determines the operations of the dialogue manager, including the construction of the topic hierarchy [81].

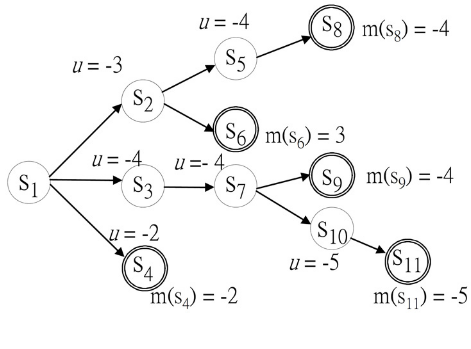


Fig. 15. A typical learning tree constructed for the retrieval states for a specific user

5.7 Some Example Test Results

In prelimary tests with the dialogue systems as presented above for voice-based information retrieval, an archive of 10,000 broadcast news stories in Mandarin Chinese served as the back-end unstructured knowledge source. The topic hierarchy presenting the system output for the block of Multi-modal User Interface in Fig. 14 and the term ranking approach for the block of Dialogue Modeling in Fig. 14 were both implemented [6], [50], [51], [81]. In the test, 5,000,000 users were simulated in training the dialogue modeling module, while another 1,000 users were simulated for test. All entered key terms were automatically extracted from the archive of news stories. We evaluated the performance of this *dialogue system* in terms of task success rate and the average number of query terms needed for a successful retrieval. The task was defined to be successful if the user is satisfied or the recall is above a given threshold [81]. Recognition errors for queries and documents were simulated by generating feature vectors according to the Hidden Markov Models with increased Gaussian mixture variances, and then recognized normally [81]. The dialogue modeling discussed above is compared against two previously proposed term selection algorithms, the *wpq* method [82] and the *tf-idf* method.

Fig. 16 (a) shows the detailed numbers of failure trials and successful trials completed in different number of query terms out of the 1000 simulated testing users. The queries was assumed to be 100% correct, and 1000 out of the 10,000 news stories were assumed to be spoken with character accuracy of 71% (the rest in text form and completely correct). It can be found that with the *tf-idf* method, 746 out of the 1000 trials failed; all successful trials were finished within 7 query terms. Much better performance was obtained for the *wpq* method. However, when the proposed dialogue modeling was used, only 120 trials failed, and all trials were completed within 4 query terms. Similar plots can be seen in Fig. 16 (b), in which query recognition accuracy was reduced to 74% and 1700 out of the 10,000 news stories were spoken with character recognition accuracy of 77%.

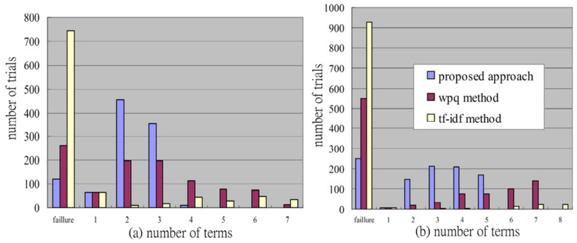


Fig. 16. Number of failure trials and successful trials completed in different number of query terms for the proposed dialogue modeling approach compared to the *wpq* and *tf-idf* methods for two different cases [81].

In Fig. 17 (a)(b) we plot the task success rate and the average number of query terms needed in successful trials for the same three methods as discussed above as functions of the query recognition accuracies, where in case (1) all the 10,000 news stories were 100% correct, and in case (2) 1700 of them has accuracy of 77%. It can be found that the performance of the dialogue modeling was very well, and quite robust with respect to recognition errors.

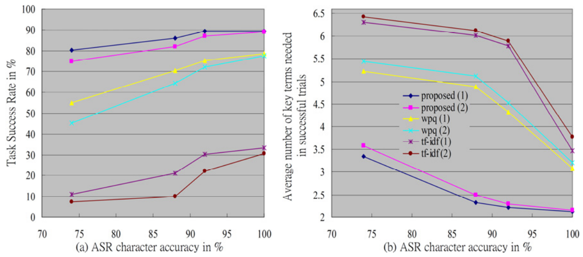


Fig. 17. (a) The task success rates and (b) the average numbers of query terms needed in successful trials for different query recognition accuracies for two different cases [81].

VI. Successful Prototype Systems for Typical Application Examples

In this section we very briefly present a few successful prototype systems for typical application examples, among many others, for voice-based information retrieval. As mentioned previously, retrieval of multimedia information based on the included voice information is a very attractive application direction. The examples presented here are all multimedia retrieval systems driven by voice.

6.1 A Broadcast News Browser

In this system [6], the broadcast news were taken as the example spoken/multimedia documents. The broadcast news archive to be summarized includes a total of 5800 news stories with a total length of 110 hours, all in Mandarin Chinese. The block diagram of the system is in Fig. 18, which includes not only the retrieval function, but the semantic analysis, automatic generation of titles and summaries, semantic structuring and dialogue loop as described in section 5.1-5.6.

For those news stories with video signals, the video signals were also summarized using video technologies, for example, video frames for human faces, moving objects and scene changes are more important, and the length of the video summary is based on the length of the speech summary. For the global semantic structure, a total of six two-dimensional tree structures were obtained for six categories of news stories, e.g. world news, business news, sports news, etc. A 3x3 small map on the second layer of the two-dimensional tree for world news overlaid with the video signal is shown in Fig. 19. This is a map expanded from a cluster in the first layer covering all disasters happening worldwide. As can be found that on this map one small cluster is for airplane crash (墜機) and similar, one for earthquake (地震) and similar, one for hurricane (颶風) and similar, and so on. All news stories belonging to each node of the two-dimensional tree are listed under the node by their automatically generated titles. The user can easily browse through the titles or click to view either the summaries or the complete stories. With this structure it is much more easier for the user to browse the news stories either top-down or bottom-up. For the query-based local semantic structuring, the topic hierarchy constructed in real-time from the news stories retrieved by a query, "White House of United States (美國白宮)," is shown on the left lower corner of Fig 20, in which the three topics on the first layer are respectively Iraq (伊拉克), US (美國) and Iran (伊朗), and one of the node in the second layer below US is President George Bush (布希). When the user clicks the node of President George Bush, the relevant news stories are listed on the right lower corner by their automatically generated titles. The user can then click the "summary" button to view the summary, or click the titles to view the complete stories. Such information are overlaid with the news retrieved with the highest score.

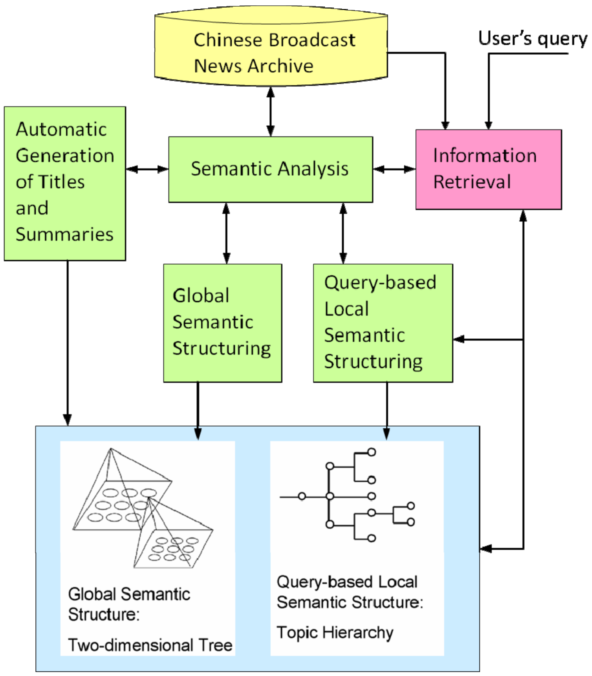
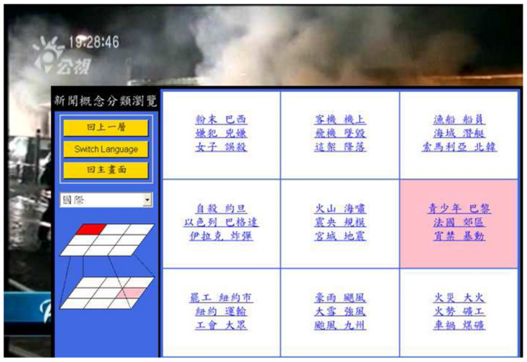

Chinese Broadcast News Archive
User's query
Automatic Generation of Titles and Summaries
Semantic Analysis
Information Retrieval
Global Semantic Structuring
Query-based Local Semantic Structuring
Global Semantic Structure:
Two-dimensional Tree
Query-based Local Semantic Structure:
Topic Hierarchy

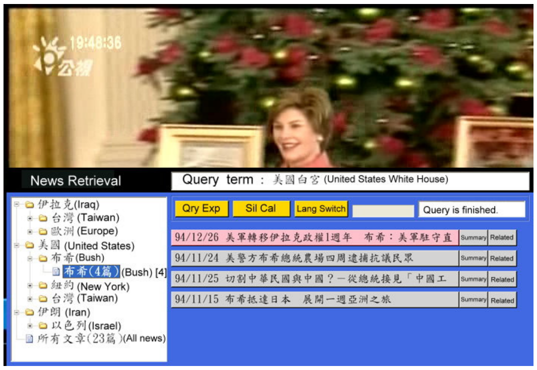
Fig. 18. The block diagram of the initial prototype system



新聞概念分類瀏覽

粉末 巴西 嫌犯 兇嫌 女子 誤殺	客機 機上 飛機 墜毀 這架 降落	漁船 船員 海域 潛艇 索馬利亞 北韓
自殺 約旦 以色列 巴格達 伊拉克 炸彈	火山 海嘯 震央 規模 宮城 地震	青少年 巴黎 法國 郊區 宵禁 暴動
罷工 紐約市 紐約 運輸 工會 大眾	豪雨 颶風 大雪 強風 颱風 九州	火災 大火 火勢 礦工 車禍 煤礦

Fig. 19. A 3x3 map on the second layer expanded from a cluster on the first layer of the global semantic structure for world news



Query term ： 美國白宮 (United States White House)

- 伊拉克(Iraq)
  - 台灣 (Taiwan)
  - 歐洲 (Europe)
- 美國 (United States)
  - 布希(Bush)
    - 布希(4篇)(Bush) [4]
  - 紐約 (New York)
  - 台灣 (Taiwan)
- 伊朗 (Iran)
  - 以色列(Israel)
- 所有文章(23篇)(All news)

Query is finished.

94/12/26 美軍轉移伊拉克政權1週年　布希：美軍駐守直

94/11/24 美警方布希總統農場四周逮捕抗議民眾

94/11/25 切割中華民國與中國？—從總統接見「中國工

94/11/15 布希抵達日本　展開一週亞洲之旅

Fig. 20. The result of query-based local semantic structuring for a query of "White House of United States"

6.2 A Course Lecture Browser

Although there have been many course lectures available over the Internet, a major difficulty of efficiently utilizing the many complete course lectures available over the network is that it takes quite long time to listen to a complete course (e.g. a complete course may include 45 hours), and it may not be easy for leaders or researchers working in the industry to spend so much time to learn a complete course. On the other hand, the content of a course is usually well structured; the learner cannot understand an advanced subject without knowing related fundamentals of the background. As a result, direct retrieval of the course content for some advanced subjects is usually not helpful to the learner, simply because the retrieved results are difficult to understand. Also, after learning a subject the learner usually doesn't know the related subjects which should be learned next.

This system divided the course lectures into "major segments" based on the slides used; key term extraction, hierarchical summarization, and semantic structuring were then performed for the "major segments". More importantly, a key term graph was constructed which serves not only as the global semantic structure, but as the query-based local semantic structure too, since the key term naturally links all the "major segments" both globally for the whole course and locally for retrieved segments. All these offer the necessary materials and scenario for the user-system interaction needed for retrieval. The user can thus enter queries to the system and learn what he needs in his own way [83].

A single course lecture corpus was used for the prototype system, which is a course on Digital Speech Processing with a total length of 45 hours. It was offered in National Taiwan University (NTU) in 2006 by a single instructor in Mandarin Chinese, while all the terminologies were produced directly in English. This system was given a name of "NTU Virtual Instructor". The block diagram of the system is shown in Fig. 21 [83]. On the upper left corner the course lectures includes 3 parts: audio, video and slides. Audio and video signals have synchronized time indices, but the slides are not synchronized with the signals in general. The core of the proposed approach is on the lower left part of Fig. 21, content organization and retrieval. Topic segmentation is to try to collect a number of "short segments" which discuss the same subject topic together into "major segments", primarily based on the transcribed short segments and the slides. Semantic analysis and summarization is then performed and key terms extracted. Semantic structuring is primarily based on a key term graph constructed for all the key terms extracted with an example partial list shown in Fig. 22, which is used to link all the major segments semantically. The basic unit for voice-based information retrieval is the short segments.

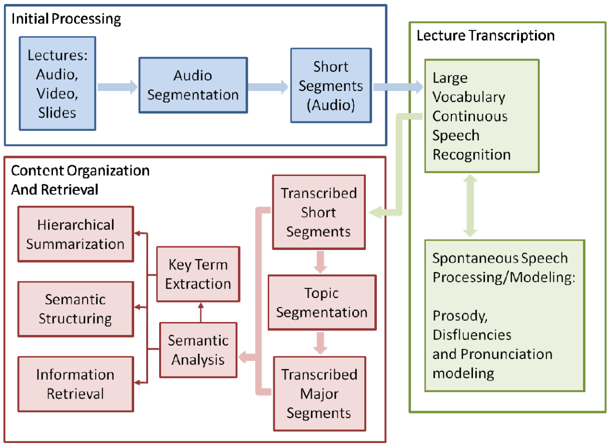

Initial Processing
Lectures: Audio, Video, Slides
Audio Segmentation
Short Segments (Audio)
Lecture Transcription
Large Vocabulary Continuous Speech Recognition
Spontaneous Speech Processing/Modeling:
Prosody, Disfluencies and Pronunciation modeling
Content Organization And Retrieval
Transcribed Short Segments
Topic Segmentation
Transcribed Major Segments
Semantic Analysis
Key Term Extraction
Hierarchical Summarization
Semantic Structuring
Information Retrieval

Fig. 21. Block diagram of the course lecture browser

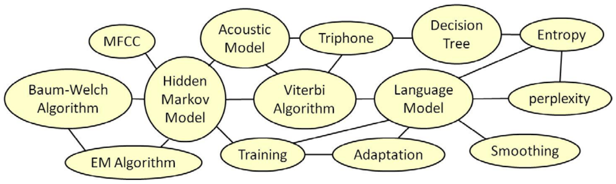


Fig. 22. An example partial list of the Key Term Graph

6.3 A Personal Photo Browser

Content-based image retrieval using image features is very successful [84] but not satisfactory for personal photos, because users prefer high-level semantic descriptions of photos that use words as indices or queries, such as who, where, when, what (objective/events) and so on. This desired scenario can be in general achieved by tags entered via networks. But such tags are usually freely entered and not associated with any type of ontology or categorization, therefore often inaccurate or ambiguous [85]. In addition, for personal photos many tags are personal and thus have to be annotated by the users themselves, such as "mammy and catty", or "my little house". Tagging by the users themselves, on the other hand, is time-consuming. Automatic tagging such as semantic concept detection [86] is promising, but still suffers from relatively low accuracy currently [87]. It will be highly desired if only very few photos need to be tagged, and if the users can tag the photos with spontaneous speech when the photo is taken. In addition, very often a given query results in a large number of photos. For efficient browsing it will be very convenient for the user if the large number of retrieved photos can be classified based on meaningful groups (or "topics" in semantic analysis as mentioned in 5.2 above) rather than just sorted by scores. Since different queries result in different sets of retrieved photos, pre-defined ontology cannot be used here. Unsupervised clustering for search results is therefore attractive.

Here we present a user-friendly latent semantic retrieval and clustering system for personal photos with sparse spontaneous speech annotation using fused speech and image features [88]. We used image features to derive the relationships among photos, since these features are the universal language describing photos. We trained semantic models with Probabilistic Latent Semantic Analysis (PLSA) as described in 5.2 above using fused speech and image features to analyze the "topics" of these photos. Only 10% of the photos need to be annotated by spontaneous speech of a few words regarding one or two semantic categories (e.g. what or where), while all photos can be effectively retrieved using high-level semantic queries in words (e.g. who, what, where, when) and clustered by the semantics as well.

As shown in Fig. 23, the proposed approach includes a preparation phase (left part) and a retrieval and clustering phase (right part). Visual words and audio words are first generated for each photo (Blocks (B) and (C), lower left of the figure) in the photo archive (Block (A), upper left corner). These two types of words are then fused to construct a "document" for each photo (Block (D), middle). These "documents" and their "words" are then used to train a PLSA topic model (Blocks (E)(F)(G), upper middle). The user query then includes only very few semantic words in text form. PLSA retrieval gives the desired photos (Block(H), right middle), which are then further clustered into "query-based local semantic structure" as described in 5.5.1 based on the PLSA topics (Block (I), lower right corner).

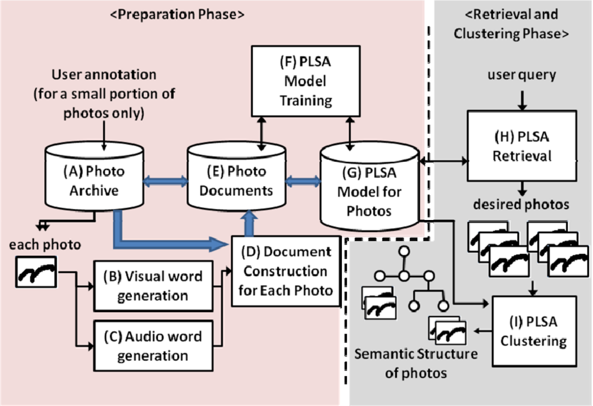

<Preparation Phase>
<Retrieval and Clustering Phase>
User annotation (for a small portion of photos only)
(F) PLSA Model Training
user query
(A) Photo Archive
(E) Photo Documents
(G) PLSA Model for Photos
(H) PLSA Retrieval
desired photos
each photo
(B) Visual word generation
(C) Audio word generation
(D) Document Construction for Each Photo
(I) PLSA Clustering
Semantic Structure of photos

Fig. 23. The proposed approach: preparation phase (document construction for photos and PLSA model training) and retrieval and clustering phase (based on PLSA)

Fig. 24 shows the user interface and an example output for an input query "大家(all together)". The three rows of photos here show the top three photos in each of the first three clusters automatically generated after retrieval. They are all photos for all people, but respectively "in the opera house", "in restaurants", and "on the street" as the three clusters here. Among the nine photos only the second of the first cluster has a speech tag of "大家在歌劇院 (all in opera house together)". All other eight photos are not tagged at all, but can be properly retrieved and clustered.

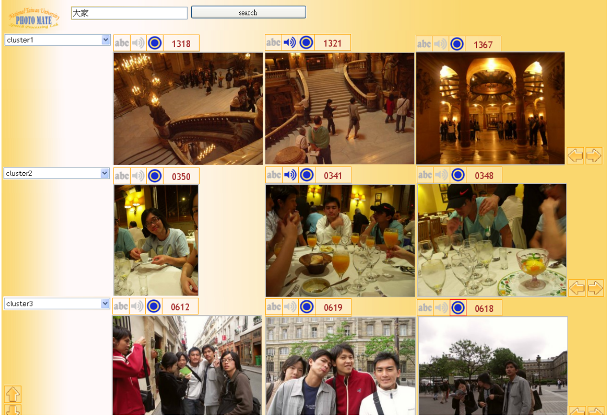


Fig. 24. The user interface and an example output for the personal photo browser

VII. Concluding Remarks

Text-based information retrieval has been extremely successful, and all roles of texts can be accomplished by voice. Multimedia content including audio information has been growing exponentially over the web. The concept of voice-based information retrieval has thus been very attractive. Substantial effort has been made and very significant progress has been obtained. However, compared to text-based information retrieval, we notice that voice-based information retrieval still has a very long way to go.

Relatively poor accuracy due to uncertainty in speech recognition, specially for spontaneous speech in adverse environments and for OOV words, is clearly the major problem. Along this direction, lattices and efficient indexing structures are very useful example approaches. Use of subword units seems to be very attractive, because it can cover many OOV words, may be used across different languages, and possibly uses much less space. Methods for reducing computation and memory requirements are definitely highly desired, since the computation and memory requirements may be prohibitively huge. In addition, many techniques useful in text-based retrieval, such as query expansion, semantic concept matching, page ranking, etc., may all be considered here.

On the other hand, more efficient user-system interaction interface is also a key issue for practical applications. Multi-modal dialogues with topic hierarchy constructed from retrieved documents whose nodes are labeled by key terms or topics may be a possible approach, and automatically generated summaries and titles may be helpful. Much more work in this area is still needed.

Speech recognition technology has been difficult for many applications, probably because the user very often expects the technology can replace human beings. Voice-based information retrieval may be an area slightly different in this aspect, since it may handle massive quantities of content which is impossible for human beings. For this reason it can be a good application area for speech recognition, although there is still a very long way to go.

REFERENCES

[1] Ciprian Chelba and Alex Acero, “Position specific posterior lattices for indexing speech”, in ACL, Ann Arbor, 2005, pp. 443-450.

[2] M. Gilbert, J. Feng, “Speech and Language Processing over the Web”, IEEE Signal Processing Magazine, May 2008.

[3] Jonathan Mamou, David Carmel, and Ron Hoory, “Spoken document retrieval from call-center conversations”, in SIGIR, 2006, pp. 51-58.

[4] Peng Yu, Kaijiang Chen, Lie Lu, and Frank Seide, “Searching the audio notebook: Keyword search in recorded conversations”, in HLT, 2005, pp. 947-954.

[5] Lin-shan Lee and Berlin Chen, “Spoken Document Understanding and Organization”, IEEE Signal Processing Magazine, Special Issue on Speech Technology in Human-machine Communication, Vol. 22, No.5, Sept. 2005, pp.42-60.

[6] Lin-shan Lee, Sheng-Yi Kong, Yi-Cheng Pan, Yi-Sheng Fu, Yu-Tsun Huang, “Multi-layered Summarization of Spoken Document Archive by Information Extraction and Semantic Structuring”, International Conference on Spoken Language Processing, Pittsburgh, USA, Sept 2006.

[7] M. Saraclar and R. Sproat, “Lattice-based search for spoken utterance retrieval”, in HLT 2004.

[8] J. Garofolo, G. Auzanne, and E. Voorhees, “The trec spoken document retrieval track: A success story”, in Recherched Informations Assiste par Ordinateur: ContentBased Multimedia Information Access Conference, 2000.

[9] http://trec.nist.gov/

[10] C. Chelba, J. Silva, and A. Acero, “Soft indexing of speech content for search in spoken documents computer speech and language”, Computer Speech and Language, vol. 21, no. 3, pp.458-478, July 2007.

[11] Z.-Y. Zhou, P. Yu, C. Chelba, and F. Seide, “Towards spoken-document retrieval for the internet: Lattice indexing for large-scale web-search architectures”, in HLT, 2006, pp. 415–422.

[12] T. Hori, I.L. Hetherington, T.J. Hazen, and J.R. Glass, “Open-vocabulary spoken utterance retrieval using confusion networks”, in ICASSP, 2007, pp. 73–76.

[13] C. Chelba, T. J. Hazen, M. Saraclar, “Retrieval and Browsing of Spoken content”, IEEE Signal Processing Magazine, May 2008, pp.39-49.

[14] D. Vergyri, et al., “The SRI/OGI 2006 Spoken Term Detection System”, Interspeech 2007, pp. 2393-2396.

[15] Y. Wang, D. Yu, Y.-C. Ju, A. Acero, “An Introduction to Voice Search”, IEEE Signal Processing Magazine, May 2008, pp. 29-38.

[16] A. Acero, et. al., “Live Search for Mobile: Web Services by Voice on the Cellphone”, ICASSP 2008, pp. 5256-5259.

[17] A. Moreno-Daniel, B.-H. Juang, J. Wilpon, "A scalable method for voice search to nationwide business listings," icassp, pp.3945-3948, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.

[18] A. Moreno-Daniel, J. Wilpon, B. H. Juang, S. Parthasarathy, “Towards the Integration of Automatic Speech Recognition and Information Retrieval for Spoken Query Processing”, Interspeech 2008, pp. 2154-2157.

[19] A. Moreno-Daniel, S. Parthasarathy, B. H. Juang, J. G. Wilpon, “Spoken Query Processing for Information Retrieval”, ICASSP 2007, pp. IV-121-IV-124.

[20] T. K. Chia, K. C. Sim, H. Li, H. T. Ng, “A Lattice-based Approach to Query-by-Example Spoken Term Retrieval”, SIGIR 2008, pp. 363-370.

[21] H. Lin, A. Stupakov, J. Bilmes, “Spoken Keyword Spotting via Multi-lattice alignment”, Interspeech 2008, pp. 2191-2194.

[22] V. Athitsos, P. Papapetrou, M. Potamias, G. Kollios, D. Gunopulos, “Approximate Embedding-Based Sequence Matching of Time Series”, SIGMOD 2008, ACM, Vancouver, Canada, June 2008.

[23] Y. Yaguchi, Y. Watanabe, K. Naruse, R. Oka, “Speech and Sound Search on the Web: System Design and Implementation”, IEEE International Conference on Computer and Information Technology 2007.

[24] J. Mamou, B. Ramabhadran, “Phonetic Query Expansion for Spoken Document Retrieval”, Interspeech 2008, pp. 2106-2109.

[25] S. Srinivasan, D. Petkovic, “Phonetic Confusion Matrix Based Spoken Document Retrieval”, SIGIR 2000, pp.81-87.

[26] K. Ng, “Towards Robust Methods for Spoken Document Retrieval”, ICSLP 1998.

[27] P. Yu, K. J. Chen, C. Y. Ma, and F. Seide, “Vocabulary-independent indexing of spontaneous speech”, IEEE Trans. Speech Audio Process., vol. 13, no. 5, pp. 635.643, 2005.

[28] F. Seide, P. Yu, Y. Shi, “Towards Spoken Document Retrieval for the Enterprise: Approximate Word-Lattice Indexing with Text Indexers”, ASRU 2007, pp. 629-634.

[29] F. Wessel, R. Schluter, K. Macherey, and H. Ney, “Confidence measures for large vocabulary continuous speech recognition”, SAP, vol. 9, no. 3, pp. 288-298, Mar 2001.

[30] L. Mangu, E. Brill, and A. Stolcke, “Finding consensus in speech recognition: Word error minimization and other applications of confusion networks”, Computer Speech and Language, vol. 14, no. 4, pp. 373-400, Oct 2000.

[31] Y.-S. Fu, Y.-C. Pan, and L.-S. Lee, “Improved large vocabulary continuous Chinese speech recognition by character-based consensus networks”, in International Symposium on Chinese Spoken Language Processing, 2006, pp. 422-434.

[32] Bernard J. Jansen, Amanda Spink, Judy Bateman, and Tefko Saracevic, “Real life information retrieval: a study of user queries on the web”, SIGIR Forum, vol. 32, no. 1, pp. 5-17, 1998.

[33] Yi-Cheng Pan, Hung-Lin Chang, Berlin Chen, Lin-shan Lee, “Subword-Based Position Specific Posterior Lattices (S-PSPL) for Indexing Speech Information”, Interspeech, Antwerp, Belgium, August 2007, pp.318-321.

[34] W.-K. Lo, F. Soong, and S. Nakamura, “Generalized posterior probability for minimizing verification errors at subword, word and sentence levels”, in ISCSLP, 2004, pp. 13-16.

[35] Q. Yao, F. K. Soong, and T. Lee, “Tone-enhanced generalized character posterior probability (GCPP) for Cantonese LVCSR”, in ICASSP, 2006, pp. 133-136.

[36] K.-C. Yang, T.-H. Ho, L.-F. Chien, and L.-S. Lee, “Statistics-based segment pattern lexicon: A new direction for Chinese language modeling”, in ICASSP, 1998, pp. 169-172.

[37] R. Wallace, R. Vogt, S. Sridharan, “A Phonetic Search Approach to the 2006 NIST Spoken Term Detection Evaluation”, Interspeech 2007, pp. 2385-2388.

[38] K. Ng, “Subword-based approaches for spoken document retrieval”, Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[39] D. Wang, J. Frankel, J. Tejedor, S. King, “A comparison of Phone and Grapheme-based Spoken Term Detection”, ICASSP 2008, pp. 4969-4972.

[40] M. Bisiani, H. Ney, “Open Vocabulary Speech Recognition with Flat Hybrid Models”, Interspeech 2005.

[41] M. Akbacak, D. Vergyri, A. Stolcke, “Open-Vocabulary Spoken Term Detection Using Graphone-based Hybrid Recognition Systems”, ICASSP 2008, pp. 5240-5243.

[42] B. Logan, J. M. V. Thong, and P. J. Moreno, “Approaches to reduce the effects of OOV queries on indexed spoken audio”, IEEE Trans. Multimedia, vol. 7, no. 5, pp. 899-906, 2005.

[43] P. Yu, K. Chen, L. Lu, and F. Seide, “Searching the audio notebook: Keyword search in recorded conversations”, in HLT 2005, pp. 947-954.

[44] S. Parlak, M. Saraclar, “Spoken Term Detection for Turkish Broadcast News”, ICASSP 2008, pp. 5244-5247.

[45] V. T. Turunen, M. Kurimo, “Indexing Confusion Networks for Morph-based Spoken Document Retrieval”, SIGIR 2007, pp.631-638.

[46] Berlin Chen, Hsin-Min Wang and Lin-shan Lee, “Discriminating Capabilities of Syllable-based Features and Approaches of Utilizing Them for Voice Retrieval of Speech Information in Mandarin Chinese”, IEEE Transactions on Speech and Audio Processing, Vol.10, No.5, July 2002, pp.303-314.

[47] Yi-Cheng Pan, Hun-Lin Chang, Lin-shan Lee, “Analytical Comparison between Position Specific Posterior Lattices and

Confusion Networks Based on Words and Subword Units for Spoken Document Indexing", IEEE Automatic Speech Recognition and Understanding Workshop, Kyoto, Japan, December 2007, pp. 677-682.

[48] Y.-C. Pan, "One-pass and word-graph-based search algorithms for large vocabulary continuous mandarin speech recognition", M.S. thesis, National Taiwan University, 2001.

[49] L.-F. Chien, "Pat-tree-based keyword extraction for Chinese information retrieval", in SIGIR, 1997, pp. 50–58.

[50] Yi-Cheng Pan, Lin-shan Lee, "Type- Ⅱ Dialogue Systems for Information Access from Unstructured Knowledge Sources", IEEE Automatic Speech Recognition and Understanding Workshop, Kyoto, Japan, Dec 2007, pp. 544-549.

[51] Yi-Cheng Pan, Chien-Chih Wang, Ya-Chao Hsieh, Te-Hsuan Lee, Yen-shin Lee, Yi-Sheng Fu, Yu-Tsun Huang and Lin-shan Lee, "A Multi-Modal Dialogue System for Information Navigation and Retrieval across Spoken Document Archives with Topic Hierarchies", Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop, San Juan, Nov-Dec 2005, pp.375-380.

[52] T. Hofmann, "Probabilistic latent semantic analysis", Uncertainty in Artificial Intelligence, 1999.

[53] Ya-chao Hsieh, Yu-tsun Huang, Chien-chih Wang and Lin-shan Lee "Improved Spoken Document Retrieval with Dynamic Key Term Lexicon and Probabilistic Latent Semantic Analysis (PLSA)", International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, May 2006, pp. I961-964.

[54] Sheng-Yi Kong and Lin-shan Lee "Improved Spoken Document Summarization Using Probabilistic Latent Semantic Analysis (PLSA)", International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, May 2006, pp. I941-944.

[55] I. Mani and M.T. Maybury, "Advances in Automatic Text Summarization", Cambridge, MA:MIT Press, 1999.

[56] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis", in Proc. ACM SIGIR Conference on R&D in Information Retrieval, 2001, pp. 19-25.

[57] S. Furui, T. Kikuchi, Y. Shinnaka, and C. Hori, "Speech-to-text and speech-to-speech summarization of spontaneous speech", IEEE Trans. on Speech and Audio Processing, vol. 12, no. 4, pp. 401-408, 2004.

[58] Sameer Maskey, Julia Hirschberg, "Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization", In Proc. of Eurospeech 2005, Lisbon, Portugal, 2005.

[59] Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore, "Incorporating speaker and discourse features into speech summarization", In Proceedings of the Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Meeting (HLT-NAACL) 2006, New York City, USA, June 2006.

[60] T.Kitade, H.Nanjo, and T.Kawahara, "Automatic extraction of key sentences from oral presentations using statistical measure based on discourse markers", In Proc. ICSLP, pp.2169--2172, 2004.

[61] S. Togashi, M. Yamaguchi, S. Nakagawa, "Summarization of spoken lectures based on linguistic surface and prosodic information", Spoken Language Technology Workshop, pp.34-37, 2006.

[62] Xiaodan Zhu and Gerald Penn, "Summarization of Spontaneous Conversations", In Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech 2006), Pittsburgh, 2006, pp. 1531-1534, September 2006.

[63] Pascale Fung, Ho Yin Chan and Jian Zhang, "Rhetorical-state Hidden Markov Models for Extractive Speech Summarization" in ICASSP 2008, 2008.

[64] Michael J. Witbrock and Vibhu O. Mittal, "Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries", in Proc. of ACMSIGIR, 1999, pp. 315-316.

[65] Michele Banko, Michael J. Witbrock, and Vibhu O. Mittal, "Headline generation based on statistical translation", in Proc. of ACL, 2000.

[66] Stephen Wan, Mark Dras, Cecile Paris, and Robert Dale, "Using thematic information in statistical headline generation", in Proc. of ACL, 2003.

[67] R. Jin and A. Hauptmann, "Automatic title generation for spoken broadcase news", in Proc. of HLT, 2001, pp. 1-3.

[68] Shun-Chuan Chen and Lin-shan Lee, "Automatic title generation for chinese spoken documents using an adaptive k nearest-neighbor approach," in Proc. in EUROSPEECH, 2003, pp. 2813-2816.

[69] Sheng-Yi Kong, Chien-Chi Wang, Ko-Chien Kuo, Lin-shan Lee, "Automatic Title Generation for Spoken Documents with a Delicate Scored Viterbi Algorithm", 2nd IEEE Workshop on Spoken Language Technology, Goa, India, Dec 2008, pp.165-168.

[70] S.-L Chuang and L.-F. Chien, "A practical web-based approach to generating topic hierarchy for text segments", in ACM SIGIR, 2004, pp. 127-136.

[71] Te-Hsuan Li, Ming-Han Lee, Berlin Chen, Lin-shan Lee, "Hierarchical Topic Organization and Visual Presentation of Spoken Documents Using Probabilistic Latent Semantic Analysis (PLSA) for Efficient Retrieval/Browsing Applications", European Conference on Speech Communication and Technology, Lisbon, Sept. 2005, pp.625-628.

[72] R. W. White, "Evaluating implicit feedback models using searcher simulations", ACM Transactions on Information Systems, vol. 23, no. 3, pp. 325-361, 2005.

[73] T. Misu and T. Kawahara, "Speech-based interactive information guidance system using question-answering technique", in ICASSP, 2007, pp. 145-148.

[74] A. Grunstein, J. Orszulak, S. Liu, S. Roberts, J. Zabel, B. Reimer, B. Mehler, S. Seneff, J. Glass, J. Coughlin, "City Browser: Developing A Conversational Automotive HMI", Proc. CHI, pp. 4291-4296, Boston, April 2009.

[75] V.W. Zue and J.R. Glass, "Conversational interfaces: Advances and challenges", Proc. of IEEE, vol. 88, no. 8, pp. 1166-1180, 2000.

[76] S. Young, "Talking to machines (statistically speaking)", in ICSLP, 2002.

[77] S. Seneff V.W. Zue, J.R. Glass, J. Polifroni, C. Pao, T.J. Hazen, and L. Hetherington, "Jupiter: A telephone-based conversational interface for weather information", IEEE Trans. on Speech and Audio Processing, vol. 8, no. 1, pp. 85-96, 2000.

[78] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding", IEEE Signal Processing Magazine, vol. 22, no. 5, pp. 16-31, 2005.

[79] J. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems", Computer Speech and Language, vol. 21, no. 2, pp. 393-242, 2007.

[80] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialogue strategies", IEEE Trans. on Speech and Audio Processing, vol. 8, no. 1, pp. 11-23, 2000.

[81] Yi-cheng Pan, Jia-yu Chen, Yen-shin Lee, Yi-sheng Fu, Lin-shan Lee, "Efficient Interactive Retrieval of Spoken Documents with Key Terms Ranked by Reinforcement Learning", International Conference on Spoken Language Processing, Pittsburgh, USA, Sept 2006, pp.333-336.

[82] S. E. Robertson, "On term selection for query expansion," Journal of Documentation, vol. 46, pp. 129–146, 1990.

[83] Sheng-Yi Kong, Miao-Ru Wu, Che-Kuang Lin, Yi-Sheng Fu, Lin-shan Lee, "Learning on Demand-Course Lecture Distillation by Information Extraction and Semantic Structuring for Spoken Documents", International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, ROC, April 2009, pp. 4709-4712.

[84] John R. Smith, Shih-Fu Chang, "VisualSEEk: a fully automated content-based image query system," ACM Multimedia 1996.

[85] L. Kennedy et al, "How Flickr helps us make sense of the world: Context and Content in community-contributed media collections," ACM Multimedia, pp. 631-640, 2007.

[86] Milind Naphade et al, "Large-Scale Concept Ontology for Multimedia," IEEE Multimedia Magazine, 2006.

[87] Rong Yan, et al, "A Learning-based Hybrid Tagging and Browsing Approach for Efficient Manual Image Annotation," CVPR 2009.

[88] Yi-Sheng Fu, Chia-Yu Wan, Lin-shan Lee, "Latent Semantic Retrieval of Personal Photos with Sparse User Annotation by Fused Image/Speech/Text Features", International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, ROC, April 2009, pp.1969-1972.