ANALYTICAL COMPARISON BETWEEN POSITION SPECIFIC POSTERIOR LATTICES AND CONFUSION NETWORKS BASED ON WORDS AND SUBWORD UNITS FOR SPOKEN DOCUMENT INDEXING

Yi-cheng Pan, Hung-lin Chang and Lin-shan Lee

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

{thomas, komkon, lslee}@speech.ee.ntu.edu.tw

ABSTRACT

In this paper we analytically compare the two widely accepted approaches of spoken document indexing, Position Specific Posterior Lattices (PSPL) and Confusion Network (CN), in terms of retrieval accuracy and index size. The fundamental distinctions between these two approaches in terms of construction units, posterior probabilities, number of clusters, indexing coverage and space requirements are discussed in detail. A new approach to approximate subword posterior probability in a word lattice is also incorporated in PSPL/CN to handle OOV/rare word problems, which were unaddressed in original PSPL and CN approaches. Extensive experimental results on Chinese broadcast news segments indicate that PSPL offers higher accuracy than CN but requiring much larger disk space, while subword-based PSPL turns out to be very attractive because it lowers the storage cost while offers even higher accuracies.

Index Terms— PSPL, S-PSPL, Spoken Document Retrieval

1. INTRODUCTION

With the rapid increase of multimedia data on the Internet, it will be more and more important to retrieve spoken documents. Multimedia content usually carries speech, which usually tells the core information of the content and can be the key for retrieving multimedia content. In the last decade in the TREC (Text REtrieval Conference) Spoken Document Retrieval track [1], very good retrieval performances bared on ASR one-best results was obtained as compared to that on human reference transcripts, although using relatively long queries and target stories [2]. It was then realized that we have to utilize ASR alternates, such as lattices, for very short queries, short story segments or very bad recognition accuracies, which are more realistic [3, 4, 5]. In this paper, we thus focus on the problem of spotting short queries from short spoken segments based on lattices.

An efficient way for indexing the user query string $\mathcal{Q}$ given the lattice L generated by each spoken segment d in the archive was first proposed [6]. This approach can be considered as a direct inversion of the whole lattice and may produce the posterior probability of $\mathcal{Q}$ in a precise way, but costs huge storage space. Some other approaches were then developed to use the lattice information but in an approximate while space-economic way. An efficient approach was proposed to cluster the word arcs in a lattice according to their positions and then generate Position Specific Posterior Lattices (PSPL) [3]. PSPL is regarded as a form of clustering in this paper, as will be clear later on, and the words in the k^{th} cluster are those words in the k^{th} position of the PSPL structure. Such position knowledge is very useful for the proximity information and $P(\mathcal{Q} = W_1, ..., W_n|d)$ (where $W_1...W_n$ are words) can be easily approximated by each compositional substrings in $\mathcal{Q}$ with appropriate positions. Approaches based on Confusion Networks (CN) were also proposed for efficient lattice information utilization but at much less space requirement compared with a direct lattice inversion [4, 7]. The basic idea of using CN is quite similar to that of PSPL, and they both consider the position information for each word arc in the lattice. However, the position information in the two approaches are obtained quite differently. In PSPL, the position information of each word is acquired by the position of that word arc (i.e., the k^{th} word arc) in each possible path in the lattice. As a comparison, in CN, the position information is acquired by a clustering process in which we consider all word arcs in the lattice as a whole. We thus may say that the position information in PSPL is obtained more locally, while that in CN more globally.

In this paper we perform analytical comparison for these two methods. In addition, to cover the OOV/rare word problems, we use the approach proposed recently [8] to incorporate subword posterior probabilities in both PSPL and CN to produce subword-based PSPL (S-PSPL) and CN (S-CN).

In the following, we first give a brief summary about PSPL and CN in Sec. 2, followed by the fundamental distinctions between them in Sec. 3. The approach for incorporating subword units in PSPL/CN is then presented in Sec. 4, followed by a summary of ranking algorithm for all these approaches in Sec. 5. Experiments are discussed in Sec. 6, with concluding remarks in Sec. 7.

2. WORD-BASED INDEXING APPROACHES

In this section, we briefly describe PSPL and CN [3, 4, 7, 9], which similarly group the word arcs in the lattice into several strictly linear *clusters*, but in different ways. Each *cluster* includes several word arcs along with the corresponding posterior probabilities. Both approaches may produce proper soft-hit indices for each word in all spoken segments as (segment id, cluster number, posterior probability).

2.1. 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 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 each word's position. 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:

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

where π is a partial path in the lattice. The backward probability $\beta(W,t)$ retains the original definition [10].

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

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

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 l^{th} 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 [10]. In this paper, we regard the tuples $(W,d,pos,prob)$ for a specific spoken segment d and position pos as a *cluster*, which in turn includes several words along with their posterior probabilities.

2.2. Confusion Network (CN)

Another approach was proposed to cluster the word arcs in a word lattice into several strictly linear and simple lists of word alternatives, or the Confusion Network (CN) [9]. We refer to these lists as *clusters* in this paper. 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 [9, 11]. In SDR tasks, however, we may consider CN as a compact structure representing the original lattice, and it can also give us the proximity information of each word arc [4, 7].

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 [9].

3. FUNDAMENTAL DISTINCTIONS BETWEEN PSPL AND CN

From Secs. 2.1 and 2.2 we may induce several fundamental distinctions between PSPL and CN in terms of the basic principles and structures. They are summarized in this section.

3.1. Basic Construction Units

The construction of PSPL is based on paths in a lattice. This is clear in Fig. 1(a)(b)(c). We first enumerate all the paths in the lattice, each with its own length (counted in 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. 2.1 is an efficient way to accomplish this. We thus regard the words in each position as a cluster as in Fig.'1(c). It is clear that the reason for the words being in the k^{th} cluster is that there exist some paths carrying those words as the k^{th} word in the paths.

In CN, on the contrary, the construction unit 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 non-overlapped arcs are never in the same cluster). The basic procedures of intra/inter-word clustering in Sec. 2.2 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 k^{th} cluster, as in Fig.1(d), 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 k^{th} one.

3.2. Posterior Probabilities

In PSPL we assign a posterior probability $prob$ to a word W in the k^{th} cluster as the ratio of the sum of weights of those paths carrying W as the k^{th} word to the sum of all path weights in the lattice. In CN, the posterior probability $prob$ assigned to a word W in the k^{th} cluster represents not only the paths carrying W as the k^{th} word, but also possibly those as the $(k-1)^{\text{th}}$, $(k+1)^{\text{th}}$ 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.

3.3. Number of Clusters

The result of CN gives a rough idea about the number of words in a reasonable recognition result at a global view. For example, if the final CN has K clusters, very possibly the utterance has around K words. This is quite different for 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.

3.4. Coverage and Space Requirement

All word n-grams appearing in the lattice also appear in some n consecutive clusters of PSPL. But this is not necessarily true for CN. As depicted in Fig. 1, 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 4^{th} cluster while W_3, W_4 in the first two clusters. This is very possible for CN and implies CN is slightly less complete in covering all possible word sequences for indexing than PSPL.

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 k^{th} word while in others it is the $(k+1)^{\text{th}}$or $(k+2)^{\text{th}}$. So the word W may simultaneously appear in the k,$(k+1)^{\text{th}}$, $(k+2)^{\text{th}}$ 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 and 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 [3, 7]. For example in Fig.1 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.

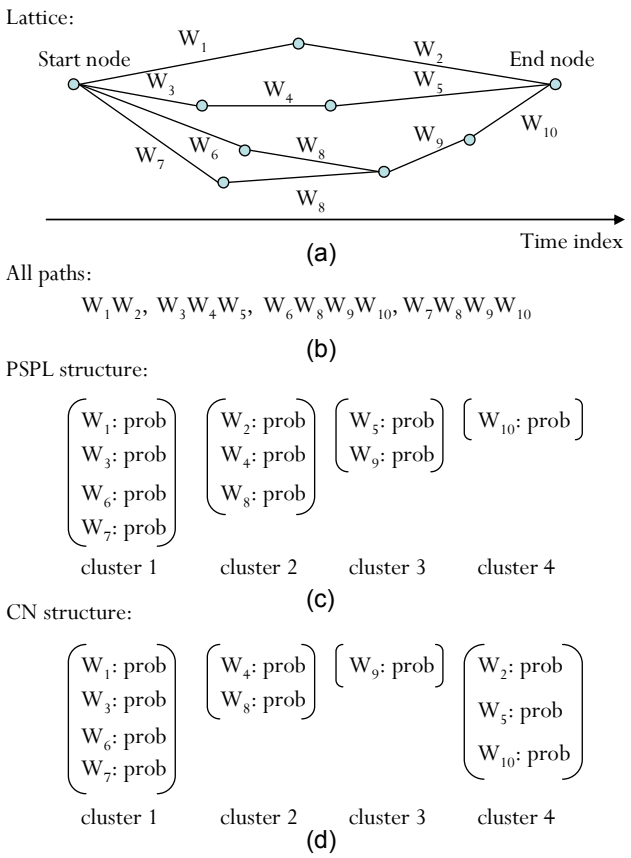

Fig. 1. (a) The ASR lattice, (b) all paths in (a), (c) the constructed PSPL structure, (d) the constructed CN structure, where $W_1, W_2, \ldots$ are words and by W_1 : prob we mean W_1 and its posterior probability in a specific cluster.

4. SUBWORD-BASED INDEXING APPROACHES

Subword-based indexing approaches have been shown to be very helpful in SDR, specially for OOV and rare words [12]. In this section we introduce a new approach to incorporate subword information in both PSPL and CN structures. We first summarize our approach to approximate subword posterior probabilities in a word lattice [8], which saves the need of ASR on the subword level. We then present the ways of incorporating this method into PSPL and CN to construct subword-based PSPL (S-PSPL) and CN (S-CN).

4.1. Subword Posterior Probability

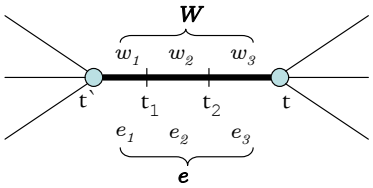

Fig. 2. Word edge W with subword units $w_1w_2w_3$ starts at time t' and ends at time t.

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.2. 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 [10], we may calculate the posterior probability of the edge e given the ASR lattice L, $P(e|L)$, as:

$$P(e|L) = \frac{\alpha(t') \cdot P(x_{t'}^{t}|W) \cdot P_{LM}(W) \cdot \beta(t)}{\beta_{start}}, \qquad (3)$$

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 of W, say w_1, we may write $P(e_1|L)$ as:

$$P(e_1|L) = \frac{\alpha(t) \cdot P(x_{t'}^{t_1}|w_1) \cdot P_{LM}(w_1) \cdot \beta(t_1)}{\beta_{start}}. \qquad (4)$$

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 in [13] 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 especially 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, via w_2 and w_3, to time t. Of course the path to go from t_1 is *at least one* but by making it "one" we may rewrite $\beta(t_1)$ as $\beta(t_1) = P(x_{t_1}^{t}|w_1w_2) \cdot \beta(t)$. We can now substitute $P_{LM}(W)$ and $P(x_{t_1}^{t}|w_1w_2) \cdot \beta(t)$ 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. 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. 2.1, 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 [10].

The elementary forward step is very similar to Eq. (1)

$$\alpha(W,t,l) = \sum_{W'} \sum_{\substack{t'\text{:}\exists \text{ edge } e \text{ starting at time } t', \text{ ending at time } t, \text{ and with } word(e)=W}} \left[\alpha(W',t',l') \cdot P_{AM}(W) \cdot P_{LM}(W)\right], \quad (5)$$

where $l = l' + Sub(W)$; $Sub(W)$ is the number of subword units in W. $P_{AM}(W)$ and $P_{LM}(W)$ are the same as in Eq. (1).

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):

$$P(W,b,b+Sub(W)-1|L) = \sum_t \frac{\alpha(W,t,b+Sub(W)-1)\cdot\beta(W,t)}{\beta_{start}} \cdot Adj(W,t), \quad (6)$$

where $Adj(W,t)$ and β_{start} are the same as Eq. (2). Following the assumptions made in section 4.1, 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:

$$P(w,k|L) = \sum_{\substack{W,b\text{: } w \text{ is the } r^{\text{th}} \text{ subword in } W \text{ and}\\ b+r-1=k}} P(W,b,b+Sub(W)-1|L). \quad (7)$$

4.3. Subword-based Confusion Network (S-CN)

It is straightforward to construct a subword-based CN (S-CN) given the approximations in Sec. 4.1. During ASR, we may record the start and end time for the subword units in each word arc. We then follow Sec. 4.1 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.

5. RELEVANCE RANKING OF SPOKEN SEGMENTS GIVEN PSPL/S-PSPL OR CN/S-CN

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

$$S(d,W_i...W_{i+N-1}) = \log\left[1+\sum_k \prod_{l=0}^{N-1} P(W_{i+l},k+l|L)\right], \quad (8)$$

$$S_{N\text{-}gram}(d,\mathcal{Q}) = \sum_{i=1}^{Q-N+1} S(d,W_i...W_{i+N-1}), \quad (9)$$

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 $S(d,\mathcal{Q})$,

$$S(d,\mathcal{Q}) = \frac{\sum_{N=1}^{Q} t_N \cdot S_{N\text{-}gram}(d,\mathcal{Q})}{\sum_{N=1}^{Q} t_N}, \quad (10)$$

where we assign the weights t_N exponentially with the N-gram order in this research. Better weight assignments may be possible.

For S-PSPL and S-CN, the above procedures remain unchanged except we decompose $\mathcal{Q}$ into a sequence of subword units instead and the allowed N-gram of $\mathcal{Q}$ is also based on subword units.

6. EXPERIMENTS

6.1. 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 segmented these stories into 5034 segments, each with one to three utterances. We used the TTK decoder [14] developed at National Taiwan University to generate the bigram lattices for these segments. From the bigram lattices, we generated the corresponding 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 different sizes of PSPL/CN and S-PSPL/S-CN were generated. Four lattices — L_1, L_2, L_3 and L_4 — were generated, 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 for the approach [6] was 19.2, 29.1, 44.5, 69.3 MB 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 [15]. The acoustic models included a total of 151 right-context-dependent intra-syllable Initial-Final (I-F) 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 1–3 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 (1.36 words long in average), while the remaining 120 were in-vocabulary (IV) queries (1.34 words long in average).

PSPL/CN and S-PSPL/S-CN, along with the 1-best ASR result baseline, result in a total of 7 experiments using the conditions: (a) word-based 1-best hypotheses; (b)(c) word-based PSPL and CN; (d)(e) character-based S-PSPL and S-CN; and (f)(g) syllable-based S-PSPL and S-CN.

6.2. Indexing Coverage Analysis

We first took the manual transcription of each of the 5034 segments as the query and computed the relevance score with respect to the

Experimental Conditions		word-based		character-based		syllable-based	
Lattice	Edges per spoken word	(b) PSPL	(c) CN	(d) PSPL	(e) CN	(f) PSPL	(g) CN
L_1	19.89	0.11	0.06	0.33	0.19	0.40	0.21
L_2	30.27	0.12	0.05	0.36	0.17	0.43	0.19
L_3	46.75	0.12	0.04	0.38	0.14	0.46	0.15
L_4	72.77	0.13	0.04	0.41	0.12	0.48	0.13

Table 1. Indexing coverage for the six experimental conditions (b)-(g).

spoken document itself as described in Sec. 5. The results averaged over the 5034 segments are listed in Table. 1. This number indicates how completely the partial and full sequences of the spoken words are covered when a lattice is produced by ASR uncertainly, which is in turn approximated by a reduced structure such as PSPL or CN. Larger values imply better coverage. We may observe that as far as indexing coverage goes, PSPL outperforms CN and S-PSPL/S-CN outperforms PSPL/CN. This is consistent with the discussion in Sec. 3.

6.3. Comparison by Retrieval Accuracy

All retrieval results presented here are in terms of Mean Average Precision (MAP) and Recall-Precision average (R-P), both evaluated by the standard trec_eval package used by the TREC evaluations. The results for in-vocabulary (IV), OOV and all queries are respectively shown in Table 2. The results for (a) was from a trigram one-pass decoding procedure (character accuracy 75.27%). The results of (b)-(g) were produced from the bigram lattice L_3 (average 46.75 edges per spoken word).

From these results we have some observations. First, word-based PSPL/CN ((b) and (c)) improved significantly from that for 1-best ((a)). This verified that lattice information is quite beneficial for the task of spotting short queries from short spoken segments. However, the weakness for word-based approaches in handling OOV queries is also clear.

By incorporating subword information, acquired from word lattices, we see significant improvements for S-PSPL/S-CN as compared to PSPL/CN((d)-(g) vs (b)(c)), not only for OOV queries but for IV queries as well. This is quite reasonable due to the fact that the concept of the language model training data used here was not very well matched with the 5034 spoken segments. Some new popular terms in the spoken segments may thus be very hard to be recognized due to very low language model scores, even if they existed in the lexicon. Subword information therefore helped.

The use of syllables deserves some discussions. Syllables carry more confusing information in Chinese (a syllable is shared by many homonym characters with different meanings), but with recognition accuracy significantly higher than that for words or characters. As a result, conditions (f)/(g) offered great advantages over (b)/(c) for all cases and even (d)/(e) for some cases on OOV queries, because it is difficult (or even impossible) to recognize OOV words as correct words or characters, but relatively easier to recognize them as correct syllables. On the other hand, for IV queries, the improvements brought by (f)/(g) were less than (d)/(e) due to the confusing information of syllables. However, when comparing conditions (f)/(g) with (b)/(c), we see considerable improvements in all cases. This also demonstrates the superiority of S-PSPL/S-CN.

Comparing PSPL and CN directly, we see that under all cases PSPL/S-PSPL gave significantly better performance over CN/S-CN. This is consistent with our discussions in Sec. 3 and the indexing coverage analysis in Sec. 6.2. But behind the higher accuracies, PSPL/S-PSPL requires more storage capacity than CN/S-CN, as also mentioned in Sec. 3 and this will be analyzed in the next section.

6.4. Comparison by Storage and Accuracy

In Table 3 we demonstrate clear tradeoffs (for structures from lattices L_1 to L_4) between index size and retrieval accuracies (MAP for all queries, IV+OOV). Generally larger lattices offered higher accuracies. An important point is that PSPL/S-PSPL takes much more space than CN/S-CN, while both of them take much less space than the original lattices [6]. It is also interesting to note that the index size gap between S-PSPL and S-CN for subword units ((f) vs (g), (d) vs (e)) is much less than that between PSPL and CN for words ((b) vs (c)). We also note the size shrinking from word-based PSPL to character-based S-PSPL ((b) to (d)) and the slight size expansion from word-based CN to character-based CN. This can be explained as follows. A word may be duplicated many times in different PSPL clusters, but a character is duplicated much less times in different S-PSPL clusters. This explains the decrease of index size from (b) to (d). But the situation is different for CN since it has much fewer duplicated words in different clusters, as discussed in Sec. 3. Decomposing the word arcs into character arcs thus expand the size. On the other hand, the size of syllable-based S-PSPL/S-CN is much less than character-based S-PSPL/S-CN ((f)(g) vs (d)(e)) since different characters are simply combined and represented by the same syllable.

The results in Table 3 are plotted in Fig. 3 where size and MAP are the two dimensions. We have six curves for the six approaches (b)-(g) considered to show the tradeoff between size and MAP. In general those approaches at the upper left corner of Fig. 3 are more attractive, because higher MAP is obtained at smaller index size. So S-PSPL looks quite attractive. The MAP degradation of S-CN for larger lattices can be also observed in Fig. 3, which is somehow consistent with the results in Table. 1. In larger lattices many arcs with very low posterior probabilities are included, which may introduce problems for the clustering approaches used by S-CN, for example discontinuities among connected arcs may be produced just as shown in Fig. 1 ($W_3W_4W_5$ are connected in the lattice but not in the CN, except here subword units instead of words are considered). Pruning techniques [9] will play important roles in such situations.

7. CONCLUSION

In this paper we perform analytical comparison including extensive experiments on PSPL and CN, as well their subword-based versions, S-PSPL and S-CN. It was found that PSPL/S-PSPL always gives better performance than CN/S-CN, but takes more space for the indices. Also, the gap in the index size between S-PSPL and S-CN is much smaller than that between PSPL and CN. Moreover, S-PSPL/S-CN always performs better than word-based PSPL/CN for both IV

Experimental Conditions		word-based			character-based		syllable-based	
		(a) 1-best	(b) PSPL	(c) CN	(d) S-PSPL	(e) S-CN	(f) S-PSPL	(g) S-CN
IV queries	MAP	0.5853	0.7445	0.7369	0.8846	0.8369	0.8419	0.8104
	R-P	0.6005	0.7228	0.6952	0.8540	0.8010	0.8164	0.7815
OOV queries	MAP	0.0747	0.1046	0.1020	0.7077	0.6759	0.7124	0.6938
	R-P	0.0732	0.0938	0.0938	0.6938	0.6583	0.6649	0.6662
IV+OOV (all queries)	MAP	0.4577	0.5906	0.5766	0.8420	0.7982	0.8107	0.7823
	R-P	0.4314	0.5715	0.5505	0.8155	0.7667	0.7799	0.7538

Table 2. Mean Average Precision (MAP) and Recall-Precision average (R-P) for in-vocabulary (IV), out-of-vocabulary (OOV) and all (IV+OOV) queries, from lattice L_3 with average 46.75 arcs per spoken word.

Experimental Conditions		word-based				character-based				syllable-based			
		(b) PSPL		(c) CN		(d) PSPL		(e) CN		(f) PSPL		(g) CN	
Lattice	Lattice Size	Size	MAP	Size	MAP	Size	MAP	Size	MAP	Size	MAP	Size	MAP
L_1	19.2	5.9	0.5643	2.1	0.5542	3.5	0.8125	2.4	0.7920	2.1	0.8015	1.5	0.7887
L_2	29.1	8.8	0.5768	2.8	0.5639	4.8	0.8309	3.0	0.7983	2.6	0.8072	1.7	0.7860
L_3	44.5	12.9	0.5906	3.6	0.5766	6.6	0.8420	3.8	0.7982	3.2	0.8107	2.0	0.7823
L_4	69.3	18.7	0.5984	4.7	0.5820	9.0	0.8492	4.7	0.7942	4.0	0.8059	2.3	0.7721

Table 3. The MAP results (for all queries, IV+OOV) with their corresponding index size (in MB) for different lattices with different sizes.

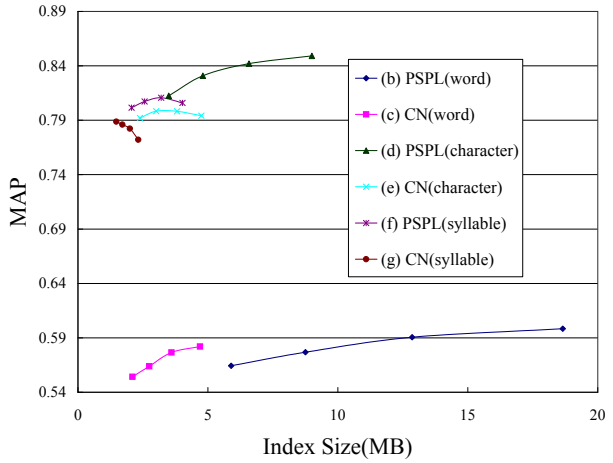

Fig. 3. The tradeoff between MAP and index size for the different approaches considered.

and OOV queries. S-PSPL turns out to be very attractive considering both accuracy and the index size.

8. REFERENCES

[1] 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.

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

[3] 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.

[4] J. Mamou, D.Carmel, and R. Hoory, "Spoken document retrieval from call-center conversations," in *SIGIR*, 2006, pp. 51–58.

[5] 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.

[6] M. Saracler and R.Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT*, 2004.

[7] 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.

[8] Y.-C. Pan, H.-L. Chang, and L.-S. Lee, "Subword-based position specific posterior lattices (S-PSPL) for indexing speech information," in *Interspeech*, 2007, pp. 318–321.

[9] 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.

[10] 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.

[11] Y.-S. Fu, Y.-C. Pan, and L.-S. Lee, "Improved large vocabulary continuous Chinese speech recognition by character-based consensus networks," in *ISCSLP*, 2006, pp. 422–434.

[12] K. Ng, *Subword-based Approaches for Spoken Document Retrieval*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[13] 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.

[14] 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.

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