Subword-based Position Specific Posterior Lattices (S-PSPL) for Indexing Speech Information

Yi-cheng Pan[1], *Hung-lin Chang*[1], *Berlin Chen*[2], *Lin-shan Lee*[1]

[1]Graduate Institute of Computer Science and Information Engineering, National Taiwan University
[2]Graduate Institute of Computer Science and Information Engineering, National Taiwan Normal University
[1]{thomas, komkon, lslee}@speech.ee.ntu.edu.tw, [2]berlin@csie.ntnu.edu.tw

10.21437/Interspeech.2007-175

Abstract

Position Specific Posterior Lattices (PSPL) have been recently proposed as very powerful, compact structures for indexing speech. In this paper, we take PSPL one step further to Subword-based Position Specific Posterior Lattices (S-PSPL). As with PSPL, we include posterior probabilities and proximity information, but we base this information on subword units rather than words. The advantages of S-PSPL over PSPL mainly come from rare and/or OOV words, which may be included in S-PSPL but generally are not in PSPL. Experiments on Mandarin Chinese broadcast news showed significant improvements from S-PSPL as compared to PSPL. Such advantages are believed to be language independent.

Index Terms: search, spoken document retrieval, subword posterior probability

1. Introduction

With the rapid increase of web content, retrieval of text information has become a very popular technology with many successful applications such as Google, Yahoo and MSN. The ever-increasing Internet bandwidth, the ever-decreasing storage costs, and the fast development of multimedia technologies have paved the way for more and more multimedia content. Multimedia content usually carries speech information, and such speech information usually contains the topics and concepts relevant to the multimedia content. Speech information thus becomes key when indexing and retrieving such content [1, 2, 3, 4]. The retrieval of speech information presents more difficulties than that of text information. For example, speech recognition errors and out-of-vocabulary(OOV) terms are well-known problems.

A very successful new approach of indexing speech information with a very compact structure, PSPL, has been recently proposed [1]. This approach efficiently considers all possible paths in the recognized lattice, as well as word proximity information within the lattice. In this way, we can not only exploit proximity information, which is very useful in text retrieval, but also take the entire lattice into account, thus minimizing the impact of recognition errors. However, the OOV problem is still left unaddressed in PSPL; that is, OOV words generally do not appear in the recognized lattice.

In this paper, we propose a new approach of subword-based PSPL (S-PSPL) motivated primarily by PSPL, but based on subword units instead of words. In this S-PSPL approach we encode the word lattice posterior probabilities and proximity information as in PSPL, except that in S-PSPL this information is based on subword units. Preliminary experiments show that S-PSPL offers significant improvements over PSPL. The improvements are clearly due to the fact that S-PSPL is able to properly handle to a good extent OOV words as well as rarely-used words, which are included in neither the lattice nor in the PSPL. This paper's contribution is to take PSPL one step further into S-PSPL and to consider a more intrinsic component of the word lattices: subword units.

2. Subword-based Position Specific Posterior Lattices (S-PSPL)

In this section, we first introduce the basic idea of Subword-based PSPL (S-PSPL), and then infer the subword posterior probability based on the word posterior probability, and finally formulate the S-PSPL.

2.1. Extending PSPL to Subword-based PSPL (S-PSPL)

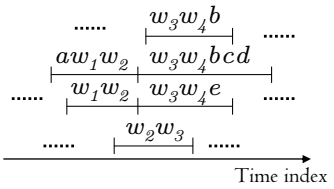


Figure 1: A partial lattice with several word arcs denoted by their constituent subword units.

With PSPL, we can index the *soft-hits* for each word in the lattice as a tuple: `(document id, position, posterior probability)` [1]. Then for the query $\mathcal{Q}$ composed of several words $\{W_i, i = 1, w.., n\}$, we may check the *soft-hits* for each of these words W_i 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 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. But 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 [5]. As an example, 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 figure 1. 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_2w_3, aw_1w_2, for instance, where $a, b, c, d, \ldots$ are subword units as

well. With PSPL the *soft-hits* for the word W do not include D due to the absense of the word arc W in D's lattice. With subword-based PSPL(S-PSPL), on the other hand, indexing is based on subword units. Each subword unit has its *soft-hits* as the tuples mentioned above. Thus for a query $\mathcal{Q}$ 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 S-PSPL D has a fair rank under this query and can be retrieved accordingly.

2.2. Subword Posterior Probabilities

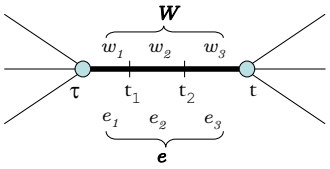


Figure 2: The word edge W with subword units $\{w_1w_2w_3\}$ starts at time τ and ends at time t.

Although the concept of S-PSPL is straightforward, the implementation requires the detailed formulation of quite a few new probabilities defined based on subword units. Consider a word W as shown in figure 2 with subword units $\{w_1w_2w_3\}$ corresponding to the edge e starting at time τ and ending at time t in the word lattice L. During the decoding process the boundaries between w_1 and w_2, and w_2 and w_3, are also recorded respectively as t_1 and t_2. The posterior probability of e given L, $P(e|L)$, is given below [6]:

$$P(e|L) = \frac{\alpha(\tau) \cdot P(x_\tau^t|W) \cdot P_{LM}(W) \cdot \beta(t)}{\beta_{start}}, \qquad (1)$$

where $\alpha(\tau)$ and $\beta(t)$ denote the forward and backward probabilities accumulated up to time τ and t as in the standard forward-backward algorithm[1], $P(x_\tau^t|W)$ is the acoustic likelihood function, $P_{LM}(W)$ the language model score, and β_{start} denotes the sum of all path scores in the lattice accumulated from the end time to the start time of the lattice. Equation (1) can be extended to the posterior probability of a subword of W, say w_1 with edge e_1 as in figure 2, as:

$$P(e_1|L) = \frac{\alpha(\tau) \cdot P(x_\tau^{t_1}|w_1) \cdot P_{LM}(w_1) \cdot \beta(t_1)}{\beta_{start}}. \qquad (2)$$

Here we need to calculate the two new probabilities $P_{LM}(w_1)$ and $\beta(t_1)$. Since neither is easy to estimate, we make some assumptions to obtain effective estimates. First, we assume $P_{LM}(w_1) \approx P_{LM}(W)$. Of course, this is not true, the actual relation being $P_{LM}(w_1) \geq P_{LM}(W)$, since the set of events having w_1 given its history includes the set of events having W given the same history (the inverse is not necessarily true). However, this is an approximation for easier implementation. Second, we assume that after w_1 there is only one path from t_1 to t: that through w_2 and w_3. Again, this is clearly not true. There may be other paths from t_1 to t through other subword units. This is again a simplifying assumption. With this assumption we have the approximation $\beta(t_1) = P(x_{t_1}^t|w_2w_3) \cdot \beta(t)$. We can now substitute these two approximate values for $P_{LM}(w_1)$ and $\beta(t_1)$ in equation (2), and the result turns out to be very simple: $P(e_1|L) \approx P(e|L)$. With similar assumptions for the subword edges e_2 and e_3, we have $P(e_2|L) \approx P(e_3|L) \approx P(e|L)$. Similar results can be observed in [7] from a different point of view.

[1] We adopt the step described in [6] to merge all nodes with identical associated times into a single node. Thus the index of the forward or backward probability mass here is specified by time rather than node number.

2.3. Position Specific Posterior Probabilities for Subword Units

Now we are ready to compute the position specific posterior probabilities for each subword unit in the word lattice. Similar to PSPL, we begin with the computation for the position specific probabilities for words, except that in this case, the position is based on subword units. A variation of the standard forward-backward algorithm can be employed for this computation. The computation for the backward pass remains unchanged, whereas 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:

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

The backward probability $\beta(W,t)$ retains the original definition [6].

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

$$\alpha(W,t,l) = \sum_{W'} \sum_{\substack{t'\text{:}\exists \text{ edge } e \\ \text{starting at time} \\ t', \text{ ending at} \\ \text{time } t \text{ and with} \\ word(e) = W}} \alpha(W',t',l') \cdot P_{AM}(W) \cdot P_{LM}(W), \qquad (3)$$

where $l' = l - Sub(W)$; $Sub(W)$ is the number of subword units in W. $P_{AM}(W)$ and $P_{LM}(W)$ denote the acoustic and language model scores of W, respectively.

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:

$$P(W,b,b+Sub(W)-1|L) = \sum_t \frac{\alpha(W,t,b+Sub(W)-1) \cdot \beta(W,t)}{\beta_{Start}} \cdot Adj(W,t), \qquad (4)$$

where $Adj(W,t)$ 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 [6]. Following the assumptions made in section 2.2, the probability of the subword w being the k^{th} subword unit in the lattice is then simply the summation of the position specific posterior probabilities for the appropriate words W:

$$P(w,k|LAT) = \sum_{\substack{W,\, b\text{: } w \text{ is the } r^{\text{th}} \\ \text{subword in } W \text{ and} \\ b+r-1=k}} P(W,b,b+Sub(W)-1|L), \qquad (5)$$

which finalizes the formulation of S-PSPL.

2.4. Relevance Ranking of Spoken Documents Based on S-PSPL

Given a query $\mathcal{Q}$ we decompose it into a sequence of subword units, $\{w_j, j = 1, 2.., Q\}$. The relevance ranking of the spoken documents based on S-PSPL proposed here is exactly the same as that based on PSPL [1] except that every word W_j is replaced by its corresponding subword units. Thus we calculate an expected tapered-count for each N-gram $\{w_i...w_{i+N-1}\}$ of the query in a spoken document D, $S(D, w_i...w_{i+N-1})$, and aggregate the results to produce a score $S_{\textit{N-gram}}(D, \mathcal{Q})$ for each order N [1]:

$$S(D, w_i...w_{i+N-1}) = \log\left[1 + \sum_k \prod_{l=0}^{N-1} P(w_{i+l}, k + l|L)\right], \quad (6)$$

$$S_{\textit{N-gram}}(D, \mathcal{Q}) = \sum_{i=1}^{Q-N+1} S(D, q_i...q_{i+N-1}), \quad (7)$$

where L is the lattice obtained from D. The different proximity types, one for each N-gram order allowed by the query length, are finally combined by a weighted sum to give the final relevance score $S(D, \mathcal{Q})$,

$$S(D, \mathcal{Q}) = \frac{\sum_{N=1}^{Q} t_N \cdot S_{\textit{N-gram}}(D, \mathcal{Q})}{\sum_{N=1}^{Q} t_N}, \quad (8)$$

here we assign the weights t_N exponentially with the N-gram order. Clearly, better weight assignments may be obtained with more principled methods.

3. Experiments

3.1. Experimental Setup

The corpus used in the experiments are the 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 taken as a document. We used the TTK decoder developed in National Taiwan University to generate the bigram lattices for these segments. From the bigram lattices we generated the corresponding PSPL/S-PSPL structures, with which we recorded the tuple `(segment id, position, posterior probability)` for each word/subword unit in the respective bigram lattice. A trigram language model trained from a 40M news corpus collected in 1999 was used. The lexicon of the decoder consisted of 62K words selected from the above training data automatically based on the PAT tree algorithm [8]. The acoustic models included a total of 151 right-context-dependent intra-syllable Initial-Final (I-F) models and it was trained by 8 hrs of broadcast news stories collected in 2000. The recognition character accuracy obtained for the 5034 segments was 75.27%. Since the corpus is in Mandarin Chinese, the subword units used in S-PSPL can be either characters, syllables, or I-F models. Here we only reported the results on characters and syllables.

We manually generated 135 test queries, each including one or two words [5]. 31 of them included OOV words and are thus categorized as OOV queries, while the remaining 104 are categorized as in-vocabulary (IV) queries. Because each Chinese word is composed of one to several characters, and each single character carries explicit meaning and can therefore also considered a mono-character *word*, a well-known approach for handling OOVs in Chinese is to simply include all characters as mono-character *words* in the lexicon. Thus, when a query includes an OOV, we simply decompose it as a concatenation of in-vocabulary mono-characters or other words using the maximum matching algorithm [9] in the case of word-based indexing and retrieval. For instance, for the three-character OOV $\{w_1w_2w_3\}$, the following decompositions are possible: $\{w_1\}\{w_2\}\{w_3\}$, $\{w_1\}\{w_2w_3\}$, and $\{w_1w_2\}\{w_3\}$. In other words, for Chinese, OOV words are no longer OOV.

3.2. Experimental Results

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 IV, OOV and all queries are respectively shown in Table 1, each with six experiments:

- (a) `Trans(word)`: word-based indexing, manual transcriptions for spoken documents automatically segmented into words by maximum matching,
- (b) `1-best(word)`: word-based indexing, ASR 1-best hypotheses for spoken documents including word boundaries,
- (c) `PSPL(word)`: word-based indexing using PSPL,
- (d) `S-PSPL(character)`: character-based indexing using S-PSPL,
- (e) `S-PSPL(syllable)`: syllable-based indexing using S-PSPL,
- (f) `[PSPL(word)]+[S-PSPL(character)]`: a linear combination of the scores of experiments (c) and (d).

From these results we have some observations. First, the results on manual transcripts in experiment (a) were all not 100%, clearly because of the inconsistency in automatic segmentation of queries and documents by the maximum-matching algorithm. For example, for a single-word query consisting of the two characters $\{w_1w_2\}$, any given document d actually including that word will not necessary be found because this word may be incorrectly segmented in d as $\{w_3w_1\}$, $\{w_2w_4\}$ due to the contexts w_3 and w_4, for instance.

Second, very significant improvements were obtained from experiment (b) `1-best(word)` to (c) `PSPL(word)` in all cases. The relative improvement of over 30% was even higher than earlier results [1]. This confirms the superiority of `PSPL`. It is interesting to note that `PSPL` also offered significant improvements even for OOV queries. In this case, this is because OOV words in queries are decomposed as described above; thus OOV words can be properly indexed by word-based `PSPL` as well.

For experiment (d), `S-PSPL(character)`, significant improvements over word-based indexing in experiment (c) can be observed in all cases, but there were relatively less significant improvements for OOV queries. This seems to be a departure from the original thoughts that S-PSPL is able to offer improvements over PSPL primarily for OOV cases. We first noted there were quite a few IV queries including rarely-used but in-vocabulary words. These words were not recognized due to low N-gram scores, but were replaced by other words in the lattices, and thus were not well matched by `PSPL(word)`, but were well handled by `S-PSPL(character)`. This is why very significant improvements were obtained from (c) to (d) for IV queries. For OOV queries, on the other hand, the OOVs were decomposed as mentioned above, and thus were well indexed by `PSPL(word)`, and therefore the improvements brought by

Experiments		(a) Trans(word)	(b) 1-best(word)	(c) PSPL (word)	(d) S-PSPL (character)	(e) S-PSPL (syllable)	(f) [PSPL(word)]+[S-PSPL (character)]
IV queries	MAP	0.8982	0.5539	0.7338	0.8231 (34%)	0.7920 (22%)	0.8801 (55%)
	R-P	0.8982	0.5883	0.7099	0.7911 (28%)	0.7698 (21%)	0.8532 (49%)
OOV queries	MAP	0.9984	0.4536	0.5983	0.6234 (6%)	0.6719 (18%)	0.6461 (12%)
	R-P	0.9935	0.4830	0.5720	0.6056 (8%)	0.6405 (16%)	0.6306 (14%)
IV+OOV (all queries)	MAP	0.9212	0.5308	0.7027	0.7772 (25%)	0.7644 (21%)	0.8263 (42%)
	R-P	0.9201	0.5641	0.6783	0.7485 (21%)	0.7401 (19%)	0.8021 (38%)

Table 1: Mean Average Precision (MAP) and Recall-Precision average (R-P) for in-vocabulary (IV), out-of-vocabulary (OOV) and all (IV+OOV) queries. The numbers in parentheses indicate the relative improvements over (c) PSPL(word).

S-PSPL(character) turned out to be relatively limited. This is to be contrasted with languages like English, in which OOVs will *never* be matched when using PSPL.

Even under a Chinese environment, though, S-PSPL(character) still offers advantages for OOV queries. First, the query-side OOV decomposition might be quite different from that in the ASR word lattices and the respective PSPL. However, under S-PSPL, query words and lattices are all converted to subword units, and thus there can be no decomposition mismatches. Second, even if the query-side decomposition is consistent with some paths in the ASR lattice, the remaining paths in the lattice may still convey other decompositions, as shown in figure 1. As the result, the posterior probabilities in PSPL(word) of the decomposed query words are underestimated since it considers only specific decompositions and therefore takes into account only a subset of the probability mass of the whole lattice. In case of S-PSPL(character), on the other hand, the subword posterior probabilities of w_i are computed considering *all* the paths that contain these subword units at the proper positions. All possible decompositions of the OOV word are taken into account given its surrounding context in the spoken segment; therefore S-PSPL(character) is more precise than PSPL(word).

The situation in experiment (e) S-PSPL(syllable) is somewhat different. Syllables carry more confusing information in Chinese because usually a syllable is shared by many homonym characters with different meanings, but recognition accuracies for syllables are usually significantly higher than those for characters. As a result, S-PSPL(syllable) offered great advantages over PSPL(word) and S-PSPL(character) on OOV queries, because it is usually difficult 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 S-PSPL(syllable) were less than S-PSPL(character) due to the less precise syllable representation of queries. However, when comparing experiments (e) S-PSPL(syllable) with (c) PSPL(word), we see considerable improvements in all cases. This also demonstrates the superiority of subword-based PSPL.

Very interesting results were found in experiment (f) when linearly combining the scores of PSPL(word) and S-PSPL(character) (these two scores were given equal weights). The results were much better than either individual approach in all cases. This is interesting, since the information of PSPL(word) and S-PSPL(character) come from the same lattice and the posterior probabilities of characters are derived from those for words. Clearly, this information is not repetitive but additive. Indeed, it is possible to investigate the combination of the three: PSPL(word), S-PSPL(character) and S-PSPL(syllable), but this will be future work.

4. Conclusion

By exploiting subword units, the more intrinsic components of word lattices, we have taken the PSPL approach one step further to S-PSPL. As with PSPL, the basic procedure is to record the proximity information and posterior probabilities of these units. We proposed methods to approximate subword posterior probabilities based on conventional word lattices; these methods are simple and require no further training or modeling. In addition, with S-PSPL we can handle rare/OOV queries better than with PSPL. Although the considerable advantages of S-PSPL over PSPL were observed only in experiments on Mandarin Chinese, it is believed the concept described here is language independent.

5. References

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

[2] Zheng-Yu Zhou, Peng Yu, Ciprian Chelba, and Frank Seide, "Towards spoken-document retrieval for the internet: lattice indexing for large-scale web-search architectures," in *HLT*, 2006, pp. 415–422.

[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] 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.

[6] 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.

[7] Yao Qian, Frank K. Soong, and Tan Lee, "Tone-enhanced generalized character posterior probability (GCPP) for Cantonese LVCSR," in *ICASSP*, 2006, pp. 133–136.

[8] Lee-Feng Chien, "Pat-tree-based keyword extraction for chinese information retrieval," in *SIGIR*, 1997, pp. 50–58.

[9] Pak kwong Wong and Chorkin Chan, "Chinese word segmentation based on maximum matching and word binding force," in *ICCL*, 1996, pp. 200–203.