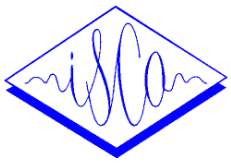

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

KEYWORD AND PHRASE SPOTTING WITH HEURISTIC LANGUAGE MODEL

Tatsuya Kawahara *Toshihiko Munetsugu* *Norihide Kitaoka* *Shuji Doshita*

Department of Information Science, Kyoto University
Sakyo-ku, Kyoto 606-01, Japan

Abstract

We discuss on a word spotting strategy that incorporates a language model as heuristics. It judges the word existence based on not only the matching score of the word itself but also the plausibility of the rest part as a sentence. Several language models are examined with respect to the accuracy and robustness, and the followings are conclusions: (1) Syllable-level knowledge is robust but insufficient. (2) Word-level knowledge is very effective and robust against spontaneous speech. (3) Word-pair constraint is most powerful but not robust. We further propose to incorporate phrase-level syntax into the spotting unit. The phrase-level syntax is rarely violated even in spontaneous utterances and significantly reduces the task perplexity. It turned out effective especially in getting a higher detection rate with a small number of false alarms.

1 Introduction

For robust recognition and understanding of spontaneous speech, it is desirable to take the spotting approach that extracts only recognizable parts and skips the rests. In the conventional spotting methods, length-free bottom-up matching is mainly adopted. But it is difficult to recognize words without using any linguistic constraint, due to the local noise or similarity.

Recently, there have been studies on incorporating low-level language model[1] and/or garbage model to spotting systems[2][3][4]. Especially, the use of linguistic knowledge as heuristics is effective to improve the spotting accuracy as it will constrain inputs and suppress unnatural false alarms. The most important issue for this approach is to construct a language model that will provide effective heuristics. In this paper, we examine several language models and present experimental evaluations.

Furthermore, we propose to use phrase-level syntax at the spotting stage. Since a phrase is uttered at one moment, its syntax is rarely violated even in spontaneous utterances. It will reduce the task perplexity and improve the spotting accuracy.

2 Spotting with Heuristic Language Model

Our goal here is to spot all the significant words which contribute to sentence understanding in a given task domain. Here, we know the input is a sort of language though it is very ill-formed. Therefore, it is desirable to model the language and incorporate this knowledge into the spotting phase.

In the conventional spotting method, the word model is matched assuming that every time-frame be a starting point or an end point of the word. But it is difficult to compare Viterbi scores that are different in length. It is also hard to precisely identify starting points and end points without investigating the neighboring parts. This sort of bottom-up matching is always annoyed with the local noise or similarity, which will cause unnatural false alarms.

Here we formulate the use of linguistic constraint on the whole input as heuristics for spotting. The evaluation function for a hypothesis that the input contains word w in time $t_1 \sim t_2$ is defined as the sum of the score $g(w,t_1,t_2)$ for the word itself and the score $h(w,t_1,t_2)$ for the rest part $(1 \sim t_1, t_2 \sim T)$. The spotting model is illustrated in Figure 1.

$$f(w,t_1,t_2) = g(w,t_1,t_2) + h(w,t_1,t_2) \qquad (1)$$

The heuristic score $h(w,t_1,t_2)$ that the rest part makes a plausible sentence in a task is ignored as zero in conventional strategies. The word spotting is formulated as finding the N-best hypotheses based on this evaluation function $f(w,t_1,t_2)$.

The heuristic score is divided into the preceding part $(1 \sim t_1)$ and the following part $(t_2 \sim T)$ of the word.

$$h(w,t_1,t_2) = h_l(w,1,t_1) + h_r(w,t_2,T) \qquad (2)$$

We call them left-context heuristics $h_l(w,1,t_1)$ and right-context heuristics $h_r(w,t_2,T)$, respectively.

This formulation solves the problems of scoring and segmentation which arise in length-free matching, as it judges after scanning the whole input. Moreover, heuristic language model will substantially reduce the perplexity of inputs and suppress unnatural false alarms. Since the heuristic model is constructed with the phonetic model that is also used for the word models, it will

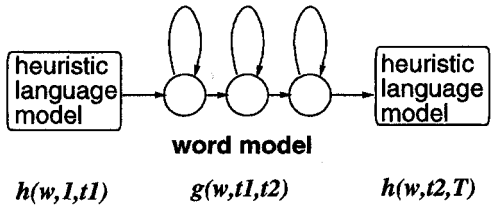

Figure 1: Spotting model with heuristic language model

provide consistent scores with the word scores. It is also possible to set the threshold for spotting dynamically with the heuristic score.

The spotting algorithm consists of two phases: heuristics computation and spotting itself.

First of all, the heuristic model is applied to the whole part of an input speech. Although the same model is used to approximate the left part and the right part of the spotted word, Viterbi scores for the two must be independently computed. Therefore, we apply the model both left-to-right and right-to-left, and stores the respective generated trellises. The evaluation function $f(w, t_1, t_2)$ is obtained by concatenating these trellises.

When we use a heuristic language model that includes the words to be spotted such as word-concatenation model, the score of the word part is obtained in heuristics computation. Thus, we can save most of the computation of the spotting phase.

Though it is three pass algorithm, it is possible to perform the two heuristics computations in parallel. Or, when we perform spotting in the right-to-left direction, it is possible to perform right-context heuristics computation and spotting in parallel and frame-synchronous. This is most efficient as the left-context heuristics can be computed in real time.

3 Language Modeling for Heuristics

The key of the spotting strategy is language modeling for heuristics that will be effective constraint and robust against variety of user utterances. Several models are examined.

Syllable-Concatenation Model

This model represents just a parallel concatenation of the possible syllables. It is self-looping of syllable models that are possible in the language. Thus, this model can accept any sentences. But it is the loosest constraint without lexical knowledge.

It is desirable to use syllable bigram since it characterizes the general language better. It is expected that bigram reduces perplexity of the language, even if the training text is not dependent on some task or domain.

Word-Concatenation Model

This model approximates an input utterance with a sequence of known words using large vocabulary lexical knowledge[1][4]. It is a parallel combination of all the possible word models. In addition to the keywords to be spotted, functional words are added to the lexicon.

Since it limits the vocabulary as in conventional parsers, the score for an input including unknown words or filled pauses is not guaranteed as correct. To cope with spontaneous utterances, models of filled pauses can also be added.

Word-Pair Model

In addition to the lexical knowledge, this model constrains the connections of the words. Only the syntactically admissible pairs of the words are connected. This constraint will improve the spotting accuracy, especially concerning short words whose phoneme sequences are similar to other words locally, for example, '*kyou* (today)' and 'Tokyo'. But syntactic constraint on the word connections is also often violated in conversational speech.

In this model, too, filled pauses can be incorporated.

Word-Syllable Hybrid Model

This is a hybrid model of words and syllables. It is a combination of syllable-concatenation and word-concatenation. The part of an input that makes some known word is expected to pass the corresponding word model, and the unexpected part such as unknown words and filled pauses will pass the syllable models.

As both syllable models and word models consist of the same phonetic model, the probability of passing the syllable models must be lowered relatively, lest the whole input pass the syllable models. It is regarded as a penalty for violation of the lexical constraint.

In this model, too, syllable bigram can be incorporated to the syllable concatenations in order to improve the model.

4 Incorporating Phrase-level Syntax

In the previous section, we examined language models for heuristics. Here, we propose to incorporate linguistic constraint into the spotting unit itself. Concretely, we use phrase-level syntax. A phrase consists of a few keywords and functional words. Even in spontaneous speech, a phrase is uttered at a moment without break, and its syntactic structure is rarely violated. Namely, local syntax is maintained even in ill-formed utterances. Moreover, since a phrase makes a semantic case and is directly mapped into a semantic representation, it will lead to robust understanding.

Unlike word spotting, it is unrealistic to evaluate all the possible phrases whose number is combinatorial of the vocabulary size. Therefore, some search technique is essential. Here we adopt best-first search.

When we need only the best segmented candidate of a phrase $\mathbf{w} = (w_1, w_2, \ldots, w_l)$, we compute the maximum score of the best path with Viterbi algorithm. It is easily extended to N-best algorithm, if we must consider multiple occurrences of the same phrase.

$$f(\mathbf{w}) = \max_{1 \le t_1 < t_2 \le T} f(\mathbf{w}, t_1, t_2) \quad (3)$$

For a partial phrase hypothesis $\mathbf{w}'$, the evaluation function is defined as an estimate that it completes a phrase.

$$\hat{f}(\mathbf{w}') = g(\mathbf{w}') + \hat{h}(\mathbf{w}') \quad (4)$$

Here, $\hat{h}(\mathbf{w}')$ represents not only the score for the rest input as in the previous sections but also the prospect for the incompleted part of the phrase.

If $\hat{h}(\mathbf{w}')$ gives upper bound of the actual best extension of the phrase, that is, if the linguistic constraint for $\hat{h}(\mathbf{w}')$ is subset of that for the phrase syntax, the search is admissible and guaranteed to find the optimal phrase candidate. In obtaining the N-best candidates as in spotting, the search outputs hypotheses correctly in order of their scores. The syllable-concatenation model satisfies this condition, though syllable-bigram does not. The word-concatenation model satisfies it if all of the word entries are included in the phrase syntax.

The spotting algorithm is much the same as word spotting, except that the search proceeds by extending the most plausible phrase hypothesis.

5 Experimental Evaluation

In order to evaluate spotting algorithms and language models, we performed speaker-independent large-vocabulary spotting experiments. The task domain we adopt is personal schedule management at a computer, and the task here is to spot 219 keywords that construct the meaning of utterances.

We used two different sets of sample sentences. One is 50 kinds of grammatical sentences, and the other is 25 kinds of un-grammatical sentences that include filled pauses. Each set of sentences is uttered by 8 male speakers. The total time (H) of 400 samples of grammatical sentences is 0.350 hours, and that of 200 samples of un-grammatical sentences is 0.224 hours.

The threshold for spotting is set based on the optimal score with a heuristic language model P_{max}. After the preliminary experiments, we set it to $P_{max} \times 1.05$ on the logarithm scaled Viterbi score.

The overall spotting accuracy is evaluated with the number of correct words (A) and false alarms (FA). As a compact measure for the spotting performance, a Figure of Merit (FOM) score is defined as the average spotting rate (A/W) from 0 to 10 FA/W/H. We also present the spotting rate in the W-best candidates, corresponding to the word recognition rate in parsing. We simply notate it as A/Wc.

5.1 Comparison of Language Models

Heuristic language models are constructed as follows.

Syllable-concatenation model consists of 110 Japanese CV (Consonant-Vowel) syllables. In order to estimate syllable bigram, we used dialogue text database of the Acoustic Society of Japan (ASJ)[5]. It is completely independent from our task domain. It has about 120 thousands of syllables. The syllable perplexity with bigram gets 21.6. Here we performed back-off smoothing on the bigram of the syllable pairs that do not appear in the corpus frequently[6].

Word-concatenation model consists of 230 words, including functional words as well as the words to be spotted. Moreover, we construct another word-concatenation model that also includes 6 filled pauses. Word-pair model constrains above word-concatenation models. Word-pair is derived by LR grammar described for continuous speech parsing. The word perplexity with the word pair is 46.2 without fillers and 51.7 with fillers.

We evaluated these language models on 219 word spotting. As well as the proposed methods, we also applied the conventional spotting algorithm based on length-free matching, which does not assume any language models. If a local peak of the Viterbi score normalized by the frame length exceeds a threshold, then the word is spotted.

The results for grammatical sentences and un-grammatical sentences are shown in Table 1.

These results show that incorporating heuristic language model is effective. The length-free matching spots words based on the local score, thus tends to generate false alarms in such parts that is locally similar to the words to be spotted.

However, syllable-concatenation model does not work well, since it does not give any constraints on language. It just works as syllable alignment of the whole input. Incorporating bigram of the syllables improves the spotting accuracy. It succeeds to model general spoken language even if its training text has nothing to do with the task domain.

Word-concatenation model works quite effectively. It brought about better accuracy than any syllable-level models. In grammatical utterances where no fillers exist, using fillers in the heuristic model had only bad effect, as it increases perplexity of the model. In un-grammatical utterances, the model with fillers was effective, but only a little. In other words, word model without fillers works quite well even in un-grammatical utterances that violate the lexicon. It is because fillers can be approximated by several short (one-syllable) words such as functional words.

Word-syllable hybrid model also works well, but it got much the same accuracy as word-concatenation model only. Incorporating bigram of the syllables had no effect in this case. It means that word-level constraint is so powerful and robust that syllable-level knowledge makes little sense.

Word-pair model got the best spotting accuracy in the experiments. While syllable-concatenation model allows any sequences, word-pair model does not approximate the rest part of a mis-placed word and prevents it from being spotted. But in un-grammatical utterances, the model without fillers is less effective. It means word-pair is not robust against spontaneous utterances where the constraint is not satisfied.

Table 1: Comparison of language models in 219 word spotting

language model	grammatical		ungrammatical	
	FOM	A/Wc	FOM	A/Wc
no heuristics	44.9	31.7	32.4	28.4
syllable-conc.	47.2	36.2	46.4	42.0
syllable-bigram	56.0	42.4	54.5	48.7
word-conc.	72.9	60.5	59.1	53.0
word-conc. (fillers)	71.9	59.4	60.5	55.8
word+syllable-conc.	70.5	60.2	60.3	55.6
word+syllable-bigram	70.5	60.0	60.2	55.6
word-pair	84.8	74.5	65.4	58.0
word-pair (fillers)	84.0	73.5	70.9	67.0

5.2 Effects of Phrase-level Syntax

Next, we incorporate phrase-level syntax. The word perplexity of the phrase grammar is 24.6, and the average number of words in a phrase is 1.98 in the sample sentences.

As the heuristic language model, we used word-syllable concatenation model, which proved both effective and robust. Actually, the coupling of the phrase grammar and the word-level heuristics realizes an effective search as their perplexities are close.

The result for 219 word detection is shown in Table 2. Here, the word rate (A/W) is computed from the phrase candidates. With phrase-level syntax, the spotting accuracy improved. The improvement of FOM that is the average rate for 0~10 FA/W/H seems only a little. But the detection rate with a rather small number of false alarms drastically improved. To see this effect, a graph of the spotting rate A/W versus the false alarm rate FA/W/H called Receiver-Operating-Characteristic (ROC) curve is shown in Figure 2. In the domain of 0~3 FA/W/H, phrase spotting got a 5~10% higher detection rate than simple word spotting. The higher accuracy with less false alarms is significant to guide the following language processing to a proper interpretation.

Table 2: Effects of phrase-level syntax

spotting method	grammatical		ungrammatical	
	FOM	A/Wc	FOM	A/Wc
word spotting (wsc)	70.5	60.2	60.3	55.6
phrase spotting (wsc)	71.8	62.3	62.2	59.0

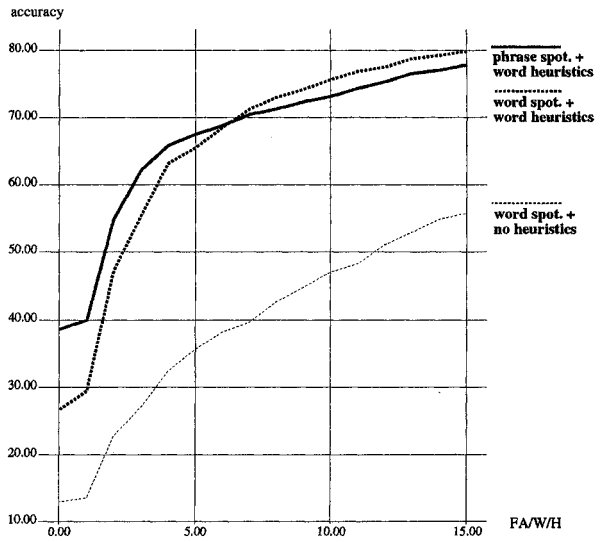

Figure 2: ROC curve for un-grammatical utterances

6 Conclusion

We have proposed two methods to utilize linguistic knowledge at the spotting stage. First, to incorporate a low-level language model as heuristics which approximates the whole input utterance. Second, to use phrase-level syntax, which is rarely violated even in spontaneous utterances and reduces the task perplexity.

Several language models are studied and compared. As experimental results of 219 word spotting, we found that word-level knowledge significantly improves the spotting accuracy. Then, we performed phrase spotting. With a phrase-level grammar, the word detection rate improved especially with a small number of false alarms.

References

[1] J.R.Rohlicek, P.Jeanrenaud, K.Ng, H.Gish, B.Musicus, and M.Siu. Phonetic training and language modeling for word spotting. In *Proc. of IEEE-ICASSP*, volume 2, pages 459–462, 1993.

[2] J.G.Wilpon, L.R.Rabiner, C.H.Lee, and E.R.Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Trans. Acoust., Speech & Signal Process.*, 38(11):1870–1878, 1990.

[3] R.C.Rose and D.B.Paul. A hidden markov model based keyword recognition system. In *Proc. of IEEE-ICASSP*, pages 129–132, 1990.

[4] M.Weintraub. Keyword-spotting using SRI's DECIPHER large-vocabulary speech-recognition system. In *Proc. of IEEE-ICASSP*, volume 2, pages 463–466, 1993.

[5] Acoustical Society of Japan. *Continuous Speech Corpus for Research*, 1993.

[6] S.K.Katz. Estimation from sparse data for the langage model for a speech recognition. *IEEE Trans. Acoust., Speech & Signal Process.*, 35(3):400–401, 1987.