SPEECH RECOGNITION USING A STOCHASTIC LANGUAGE MODEL INTEGRATING LOCAL AND GLOBAL CONSTRAINTS

Ryosuke Isotani, Shoichi Matsunaga

ATR Interpreting Telecommunications Research Laboratories
Seika-cho, Soraku-gun, Kyoto 619-02, Japan

ABSTRACT

In this paper, we propose a new stochastic language model that integrates local and global constraints effectively and describe a speech recognition system based on it. The proposed language model uses the dependencies within adjacent words as local constraints in the same way as conventional word N-gram models. To capture the global constraints between non-contiguous words, we take into account the sequence of the function words and that of the content words which are expected to represent, respectively, the syntactic and semantic relationships between words. Furthermore, we show that assuming an independence between local- and global constraints, the number of parameters to be estimated and stored is greatly reduced.

The proposed language model is incorporated into a speech recognizer based on the time-synchronous Viterbi decoding algorithm, and compared with the word bigram model and trigram model. The proposed model gives a better recognition rate than the bigram model, though slightly worse than the trigram model, with only twice as many parameters as the bigram model.

1. INTRODUCTION

At present, word N-gram models [1], especially bigram ($N = 2$) or trigram ($N = 3$) models, are recognized as effective and are widely used as language models for speech recognition. Such models, however, represent only local constraints within a few successive words and lack the ability to capture global or long distance dependencies between words. They might represent global constraints if N were set at a larger value, but it is not only computationally impractical but also inefficient because dependencies between non-contiguous words are often independent of the contents and length of the word string between them. In addition, estimating so many parameters from a finite number of text corpora would result in sparseness of data.

Recently some papers treat long distance factors. In the long distance bigrams by Huang et al. [2] a linear combination of distance-d bigrams is used. All the preceding words in a window of fixed length are considered, and bigram probabilities are estimated for each distance d between words respectively. The extended bigram model by Wright et al. [3] uses a single word selected for each word according to a statistical measure as its "parent." The extended bigrams are insensitive to the distance between the word and its parent, but this model does not utilize multiple information. The trigger pairs described in [4, 5] also represent relationships between non-contiguous words. They are also extracted automatically and insensitive to the distance. The way of combining the evidence from trigger pairs with local constraints ("the static model" in their term) is also given. But this approach has the disadvantage that it is computationally expensive. Another approach is a tree-based model [6], which automatically generates a binary decision tree from training data. Although it could also extract similar dependencies by setting binary questions appropriately, it has the same disadvantage as the trigger-based model.

We therefore proposed a new language model based on function word N-grams[1] and content word N-grams [7]. Global constraints are captured effectively without significantly increasing computational cost nor number of parameters by utilizing simple linguistic tags. Function word N-grams are mainly intended for syntactic constraints, while content word N-grams are for semantic ones. We already showed their effectiveness for Japanese speech recognition by applying them to sentence candidate selection from phrase lattices obtained by a phrase speech recognizer. We also gave a method to combine these global constraints with local constraints similar to conventional bigrams, and demonstrated that it improves performance.

In this paper, we extend and modify this model so that it can be incorporated directly into the search process in continuous speech recognition based on the time-synchronous Viterbi decoding algorithm. The new model uses the conventional word N-grams for local constraints with N being a small value, and uses function- and content word N-grams as global constraints, where N can again be small. These constraints are treated statistically in a unified manner.

A similar approach is found in [8], where, to compute a word probability, the headwords of the two phrases immediately preceding the word are used as well as the last two words. Our model is different from this method in that the former also takes function words into consideration, and treats function words and content words separately in computing the probability to extract more effective syntactic and semantic information, respectively.

In the following sections, we first explain the proposed language model, where we also show that the number of parameters can be reduced by assuming an independence between local- and global constraints. Then we describe how it is incorporated into the time-synchronous Viterbi decoding algorithm. Finally, results of speaker-dependent sentence recognition experiments are presented, where our model is compared with the word bigram and trigram models in the viewpoints of number of parameters, perplexity, and recognition rate.

2. LANGUAGE MODELING

Linguistic constraints between words in a sentence include syntactic ones and semantic ones. The syntactic constraints are often specified by the relationships between the cases of the words or phrases. Con-

[1] Previously referred to as "particle N-grams."

(a)	*Kaigi*	*-wa /*	*futsuka*	*-kara /*	*itsuka*	*-made /*	*Kyoto*	*-de /*	*kaisaisare*	-masu.
	the conference	(CM)	the 2nd	from	the 5th	to	Kyoto	in	be held	(aux. v.)
	(The conference will be held in Kyoto from the 2nd to the 5th.)									
(b)	*Soredewa /*	*tourokuyoushi*		*-o /*	*ookuri*	*-itashi*	*-masu.*			
	then	the registration form		(CM)	send	(aux. v.)	(aux. v.)			
	(Then I will send you the registration form.)									

Figure 1: Examples of Japanese sentences
(CM: case marker, aux. v.: auxiliary verb)

sequently, they are expected to be reflected in the sequence of the cases of the words or phrases. Taking notice that case information is mainly conveyed by function words in Japanese, we consider function word sequences to capture syntactic constraints while ignoring content words in the sentences. On the contrary, semantic information is mostly contained in the content words. Accordingly the idea of content word sequences is also introduced to extract semantic constraints.

After briefly explaining the roles of the function words and content words in Japanese sentences, we will propose a new model, model I, as an extension of the conventional N-gram model. In this model, the relationships between function words and between content words are taken into consideration only implicitly. Then by making some assumptions, model II will be derived as an approximation of model I. Model II uses the probabilities of function word N-grams and content word N-grams directly and may be easier to grasp intuitively.

2.1. Function Words and Content Words in Japanese

A common Japanese sentence consists of phrases ("*bunsetsu*"), each of which typically has one content word and optional function words. Figure 1 shows examples of Japanese sentences. In the figure, "/" represents a phrase separator. Words after "-" in a phrase are function words and all others are content words[2]. The corresponding English words are given in the figure. Content words include nouns, verbs, adjectives, adverbs, etc. Function words are particles and auxiliary verbs. Japanese particles include case markers such as "*ga*" (subjective case marker), "*o*" (objective case marker) as well as words such as "*kara* (from)" or "*de* (in)." Every word in a sentence is classified either as a content word or as a function word.

Paying attention only to function words and ignoring content words in sentences, "*kara* (from)" often comes before "*made* (to)" while "*ga*"s (subjective case markers) rarely appear in succession in a sentence. Thus, a sequence of function words is expected to reflect the syntactic constraints of a sentence. If we consider the content word sequence instead, such words as "*sanka* (participate)" or "*happyou* (give a presentation)" appear more frequently than words such as "*okuru* (send)" after "*kaigi* (conference)." On the other hand, after "*youshi* (form)," "*okuru* (send)" comes more frequently. Like these examples, a sequence of content words in a sentence is expected to be constrained by semantic relationships between words.

These kinds of constraints can be described statistically. To acquire these global constraints, the proposed language model makes use of the N-gram probabilities of both function words and content words.

[2]These marks are for explanation only and never appear in actual Japanese text.

2.2. Proposed Language Model I

Suppose a sentence S consists of a word string $w_1, w_2, \ldots, w_n$, and denote a substring $w_1, w_2, \ldots, w_i$ as w_1^i. Then the probability of the sentence S is written as

$$P(S) = P(w_1, w_2, \ldots, w_n) \quad (1)$$

$$= \prod_{i=1}^{n} P(w_i \mid w_1^{i-1}). \quad (2)$$

In conventional word N-gram models, each term of the right hand side of expression (2) is approximated as the probability given for a single word based on the final $N-1$ words preceding it. In the bigram model, for example, the following approximation is adopted:

$$P(w_i \mid w_1^{i-1}) \simeq P(w_i \mid w_{i-1}). \quad (3)$$

The proposed model is an extension of the N-gram model and utilizes the global constraints represented by function- and content word N-grams as well. For simplicity, only a single preceding word is taken into account, both for global and local relationships. Let f_i and c_i denote the last function word and the last content word in the substring w_1^i, respectively. The probability of a word w_i given w_1^{i-1} is, taking f_{i-1} and c_{i-1} into consideration as well as w_{i-1}, represented approximately as follows:

$$P(w_i \mid w_1^{i-1}) \simeq P(w_i \mid w_{i-1}, c_{i-1}, f_{i-1}). \quad (4)$$

As w_{i-1} is identical to c_{i-1} or f_{i-1}, it is rewritten as

$$P(w_i \mid w_{i-1}, c_{i-1}, f_{i-1}) = \begin{cases} P(w_i \mid w_{i-1}, f_{i-1}), & w_{i-1}\text{: content word} \\ P(w_i \mid w_{i-1}, c_{i-1}), & w_{i-1}\text{: function word.} \end{cases} \quad (5)$$

We refer to the model based on equation (5) as "proposed model I." Figure 2 shows how the word dependencies are taken into account in

Figure 2: Word dependency in model I

this model. The probability of each word in a sentence is determined by the preceding content- and function-word pair. If content words and function words appear alternately, this model reduces to the trigram model. But when, for example, a function word is preceded by more than one content word, the most recent function word is used to predict it instead of the last word but two (w_{i-2}).

2.3. Proposed Model II — Reduction of the Number of Parameters

The following two assumptions are introduced as an approximation to reduce the number of parameters:

1. Mutual information between w_i and w_{i-1} is independent of f_{i-1} if w_{i-1} is a content word, and independent of c_{i-1} if w_{i-1} is a function word, i.e., the following approximations hold;

$$I(w_i, w_{i-1} \mid f_{i-1}) = I(w_i, w_{i-1}) \quad (6)$$

if w_{i-1} is a content word, and

$$I(w_i, w_{i-1} \mid c_{i-1}) = I(w_i, w_{i-1}) \quad (7)$$

if w_{i-1} is a function word.

2. The appearance of a content word and that of a function word are mutually independent when they are located non-contiguously in a sentence, i.e.,

$$P(w_i \mid f_{i-1}) = P(w_i) \quad (8)$$

if w_{i-1} and w_i are content words, and

$$P(w_i \mid c_{i-1}) = P(w_i) \quad (9)$$

if w_{i-1} and w_i are function words.

From these approximations, expression (5) is rewritten as

$$P(w_i \mid w_{i-1}, c_{i-1}, f_{i-1}) = \begin{cases} P_L(w_i \mid w_{i-1}) \cdot \dfrac{P_G(f_i \mid f_{i-1})}{P_G(f_i)} \\ \quad w_{i-1}\text{: content word, } w_i\text{: function word } (=f_i) \\ P_L(w_i \mid w_{i-1}) \cdot \dfrac{P_G(c_i \mid c_{i-1})}{P_G(c_i)} \\ \quad w_{i-1}\text{: function word, } w_i\text{: content word } (=c_i) \\ P_L(w_i \mid w_{i-1}) \qquad \text{otherwise,} \end{cases} \quad (10)$$

where P_L and P_G represent the probabilities of local and global constraints between words. To be more exact, $P_G(f_i)$ is the probability that the i-th word is f_i knowing that it is a function word, and $P_G(f_i \mid f_{i-1})$ is the probability that the i-th word is f_i given that the most recent function word is f_{i-1} and also knowing that the i-th word is a function word. $P_G(c_i)$ and $P_G(c_i \mid c_{i-1})$ are explained in the same way. In other words, $P_G(\cdot)$ denotes a probability in the function (or content) word sequences obtained by extracting only function (or content) words from sentences. Notice should be taken that $P_G(\cdot)$ is used only when two function (or content) words appear non-contiguously. We refer to the model based on equation (10) as "proposed model II."

This approximate equation shows that the probabilities of words in a sentence are expressed as the product of word bigram probabilities and function word (or content word) bigram probabilities, which describe local and global linguistic constraints, respectively. The term

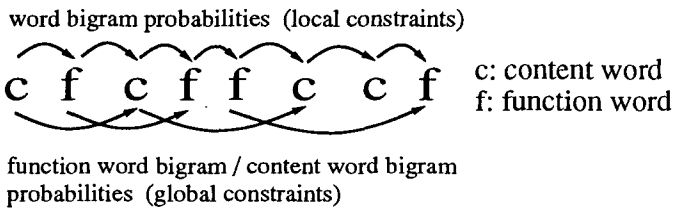

Figure 3: Word dependency in model II

$P_G(f_i)$ and $P_G(c_i)$ in the denominators can be intuitively interpreted as the compensation for the probability of word w_i being multiplied twice.

Figure 3 shows how the word dependencies are taken into account in this model. The probability of each word is determined from the word immediately before it, and also from the preceding word of the same category (function word or content word) if the category of the word immediately before it is different from that of the current word. The first corresponds to the word bigram probability and the latter corresponds to the function word (or content word) bigram probability, which are computed independently. It is easy to extend this model so as to use a word trigram model or a function word (content word) trigram model.

The decomposition of probabilities greatly reduces the number of parameters to be estimated. The number of parameters in each model is summarized in Table 1, where V, V_c, V_f is the vocabulary size, the number of content words, and the number of function words, respectively ($V = V_c + V_f$). The word trigram model and the proposed model I has $O(V^3)$ parameters, while the proposed model II has only $O(V^2)$ parameters, which is comparable to the word bigram model.

3. APPLICATION TO SPEECH RECOGNITION

Since, like N-gram models, the proposed language models are Markov models, they can easily be incorporated into a speech recognition system based on the time-synchronous Viterbi decoding algorithm. They could also be used in rescoring for N-best hypotheses, but it would bring some loss of information.

Figure 4 shows the network representation of the language model. Symbols c^i, c^j, c^k represent content words, and f^l, f^m, f^n represent function words. Each node of the network is a Markov state

Language Model	Number of Parameters
Bigram	V^2
Trigram	V^3
Proposed (I)	$2V_cV_fV$
Proposed (II)	$V^2 + V_c^2 + V_f^2$

V: vocabulary size ($= V_c + V_f$)
V_c: number of content words
V_f: number of function words

Table 1: Number of parameters of each model

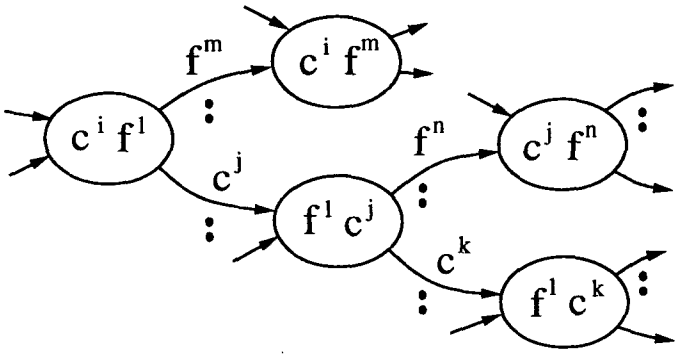

Figure 4: Network representation of the proposed language model

corresponding to a word pair of either (c_{i-1}, f_{i-1}) or (f_{i-1}, c_{i-1}), and each arc is a transition corresponding to a word w_i. In the case of the trigram model, each node would correspond to a word pair (w_{i-2}, w_{i-1}). Each arc is assigned with a probability value according to equation (5) (model I) or (10) (model II). The number of nodes is $2V_cV_f$ and the total number of arcs is $2V_cV_fV$ for both model I and model II. In the case of the trigam model, they would be V^2 and V^3, respectively.

Ordinary time-synchronous Viterbi decoding controlled by this network is possible. As the numbers of nodes and arcs are still huge although reduced compared with the trigram model, a beam search is necessary in the decoding process.

4. EXPERIMENTS

4.1. Estimation of Language Model Parameters

A 11,000-sentence text database of Japanese conversations concerning conference registration was used to train the language models. This database is manually labeled with part of speech tags. Each word was classified as a function word or a content word according to its part of speech. The size of the vocabulary is 5,389 words (5,041 content words and 348 function words), where words having the same spelling but different pronunciation, or were different parts of speech, were counted as different words.

The probability values in the language models were estimated by the maximum likelihood method. These values were then smoothed using the deleted interpolation method [9]. To cope with the unknown word problem, 'zero-gram' probabilities (uniform distribution) were also used in the interpolation. In the model II, this interpolation was applied to probabilities of local constraints (P_L) and those of global constraints (P_G), respectively.

4.2. Experimental Conditions

Speaker-dependent continuous speech recognition experiments were carried out under the conditions shown in Table 2. The domain of the recognition task is the same as that of the training data, but the text of the test speech data was not included in the training data. Context-independent continuous mixture HMMs were used as acoustic models. The details of the acoustic models are shown in Table 3.

Task	International conference registration
Vocabulary Size	1,500 words
Speaker	1 male speaker
Test Data	261 sentences
	(7.0 words/sentence, on average)

Table 2: Experimental conditions for speech recognition

The proposed model was compared with the word bigram and trigram models in their perplexities for test sentences and in sentence recognition rates. As for the proposed model I, only perplexity was calculated. The ratios of the numbers of parameters were also calculated based on Table 1.

In the calculation of perplexity for model II, use of the values obtained by equation (10) does not give the correct perplexity because

$$\sum_{w_i} P(w_i \mid w_{i-1}, c_{i-1}, f_{i-1}) = 1 \qquad (11)$$

does not hold due to the approximation. Therefore the values of $P(w_i \mid w_{i-1}, c_{i-1}, f_{i-1})$ were normalized in order to satisfy this equation. This normalization was done by simply multiplying a constant value found for each combination of $(w_{i-1}, c_{i-1}, f_{i-1})$. It was omitted in the recognition experiment for computational reasons.

Beam width for recognition was fixed at 6,000 in all cases. Weighting values for the acoustic score and linguistic score were determined by preliminary experiments. Common weighting values were used for all models.

4.3. Results

The results are shown in Table 4. The proposed models give lower perplexities than the bigram model, although not so low as the trigram model, which is reflected in the speech recognition accuracy. The perplexity of model II is higher than that of model I, which we think is caused by the approximation used to derive model II, but the smallness of the increase supports the validity of the assumptions described in 2.3.

Although the perplexity and recognition rate are improved compared with the bigram model, the gain is modest. This may be due to a lack of training data or to a mismatch between the training and test data, especially since the difference in performance is also small between the bigram model and the trigram model.

However, the fact that the performance obtained by the proposed model II lies almost half way between the bigram and trigram, shows that the proposed model has the capability to capture linguistic con-

Number of Phonemes	38
Topology	4-state 3-loop, left-to-right model
Output Probabilities	Gaussian mixtures
Number of Mixtures	max 14 (variable)
Training Data	2620 word utterance

Table 3: HMM used as the acoustic models

Language Model	Perplexity	Sentence Recognition Rate	Ratio of Number of Parameters
Bigram	41.2	51.3%	1.0
Trigram	36.3	54.0%	5.4×10^3
Proposed (I)	36.9	—	6.5×10^2
Proposed (II)	38.1	52.5%	1.9

Table 4: Test set perplexity and sentence recognition rate

straints effectively with a comparatively small number of parameters. Its performance could be improved by extending it to use trigram probabilities for local or global constraints,

5. DISCUSSIONS

In an attempt to capture the global constraints, we took note of the role of function words as case markers and used their N-gram probabilities to extract the syntactic constraints. We also used the N-gram probabilities of the content words to extract the semantic constraints.

One of its advantages is that it does not need expensive computational cost compared with previous works [3, 4, 5, 6]. Furthermore, as the syntactic constraints are considered to be less dependent on the domain than the semantic ones, function word N-grams could be trained with a task-independent large database and combined with content word N-grams trained with a task-dependent smaller database.

One of the disadvantages of our approach is that the labels indicating whether a word is a function word or a content word are necessary in the training data. We think it would not be so difficult to automatically label if we only have to classify the words into these two categories, because the category of function words can be regarded as a closed class.

Another problem is its generality, especially its applicability to other languages. English, for example, has different structure of sentences and different way of specifying the cases, although relationships between the content words are expected to exist. We think similar approach could be also useful for other languages, but some modification may be needed.

6. CONCLUSIONS

In this paper, a speech recognition system using a new stochastic language model that integrates local and global linguistic constraints was proposed. Function word bigrams and content word bigrams were introduced to capture global syntactic and semantic constraints, and combined with a conventional word bigram model. The number of parameters was reduced by decomposing local and global dependency.

Continuous speech recognition based on the time-synchronous Viterbi decoding algorithm with the proposed language model incorporated into it was presented, and speaker-dependent speech recognition experiments were conducted. Although the improvements in performance over the conventional bigram model are rather modest, results show that the proposed model has the capability to capture linguistic constraints effectively.

The assumptions made to reduce parameters do not degrade perplexity, but their validity needs to be verified from the linguistic point of view. The number of parameters is reduced in the proposed model, but the size of database we used is still not large enough to estimate the statistics in the model. More data would be necessary to evaluate the effectiveness of the proposed model. The use of part of speech or word equivalence classes generated automatically (for example, [10]) could help to increase the robustness of the estimates obtained from the limited size of the corpora.

In the future, we plan to further investigate the effective utilization of linguistic knowledge as well as statistical approaches to extract more useful global constraints.

Acknowledgments

We would like to thank Mr. Sagayama, NTT Interface Laboratories, for his valuable advice. We are also grateful to Dr. Sagisaka and the members of Department 1 for their useful comments and help.

References

1. Bahl, L. R., Jelinek, F., and Mercer, R. L., "A maximum likelihood approach to continuous speech recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, 1983, pp. 179–190.
2. Huang, X., Alleva, F., Hon, H-W., Hwang, M-Y., Lee, K-F., and Rosenfeld, R., "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7, 1993, pp. 137–148.
3. Wright, J. H., Jones, G. J. F., and Lloyd-Thomas, H., "A consolidated language model for speech recognition," *Proc. Eurospeech 93*, 1993, pp. 977–980.
4. Lau, R., Rosenfeld, R., and Roukos, S., "Adaptive language modeling using the maximum entropy principle," *Proc. ARPA Human Language Technology Workshop*, 1993.
5. Lau, R., Rosenfeld, R., and Roukos, S., "Trigger-based language models: a maximum entropy approach," *Proc. ICASSP 93*, 1993, pp. II-45–II-48.
6. Bahl, L. R., Brown, P. F., de Souza, P. V., and Mercer, R. L., "A tree-based statistical language model for natural language speech recognition," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol. 37, 1989, pp. 1001–1008.
7. Isotani, R. and Sagayama, S., "Speech recognition using particle N-grams and content-word N-grams," *Proc. Eurospeech 93*, 1993, pp. 1955–1958.
8. Jelinek, F., "Self-organized language modeling for speech recognition," IBM research report, 1985. Also available in *Readings in Speech Recognition*, Waibel, A. and Lee, K-F., eds., 1990.

9. Jelinek, F. and Mercer, R. L., "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, Gelsema, E. S. and Kanal, L. N., eds., North-Holland Publishing Company, 1980.

10. Kneser, R. and Ney, H., "Improved clustering techniques for class-based statistical language modelling," *Proc. Eurospeech 93*, 1993, pp. 973–976.