Language Modeling by String Pattern N-gram for Japanese Speech Recognition

Akinori Ito and Masaki Kohda

Yamagata University
Jonan 4-3-16, Yonezawa 992 Japan

ABSTRACT

This paper describes a new powerful statistical language model based on N-gram model for Japanese speech recognition. In English, a sentence is written word-by-word. On the other hand, a sentence in Japanese has no word boundary character. Therefore, a Japanese sentence requires word segmentation by morphemic analysis before the construction of word N-gram. We propose an N-gram based language model which requires no word segmentation. This model uses character string patterns as units of N-gram. The string patterns are chosen from the training text according to a statistical criterion. We carried out several experiments to compare perplexities of the proposed and the conventional models, which showed the advantage of our model.

For many of the readers' interest, we applied this method to English text. As the result of a preliminary experiment, the proposed method got better performance than conventional word trigram.

1. Introduction

Markov model based statistical language model (N-gram) is very popular among continuous speech recognition. In English and many other languages, a word or a word class is often used as a unit of N-gram. It is natural because an English sentence is described word-by-word. The process to divide a sentence into words is quite simple because word boundaries are indicated by space characters. On the other hand, a sentence has no word boundary indicator in Japanese and several Asian languages. Figure 1 shows an example of a Japanese sentence and its word boundary. Suppose if an English sentence were written as "`Showme10AMflightsfromBostontoDenverandhowmuchtheycost`". Therefore, word segmentation by morphemic analysis is indispensable for word-by-word processing of Japanese text. Several Japanese morphemic analyzers are available and they are used as preprocessors of machine translation, information retrieval and other natural language processings. In the field of continuous speech recognition, a morphemic analyzer was used to make Japanese word N-gram[1].

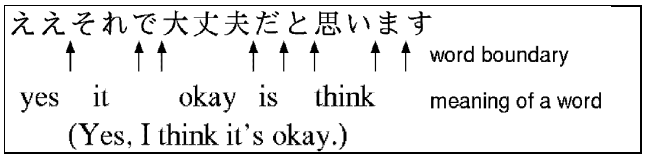

Figure 1: A Japanese sentence and its word boundary

The use of morphemic analyzer to make word N-gram has several problems.

1. Performance of N-gram is affected by morphemic analyzer. Almost all analyzers are tuned to analyze written language. Therefore, the precision of analysis for transcription of spontaneous speech is not high.

2. Criteria of what to be a word are different between analyzers. This means that a string an analyzer takes as one word might be regarded as two words by another analyzer. This problem arises from the fuzziness of Japanese "word". There is no consensus of the definition of word among Japanese grammarians.

3. At the viewpoint of statistical language model, there is no evidence that word N-gram model is optimal.

To avoid the problems around morphemic analysis, several models are proposed to use sub-word unit (phoneme[2], syllable[3] or character[4]) as the unit of N-gram. Among these models, character N-gram is simple and effective. This model, however, might be weaker as a linguistic constraint than word N-gram because a character is shorter than a word. This problem can be solved by making N large, but it causes other problems about calculation time, memory space and probability estimation.

We propose a new solution of this problem. Our model uses a string pattern (sequence of character) as a unit of N-gram. The string patterns are statistically chosen from the training text. This model enables completely automatic construction of N-gram without morphemic analysis.

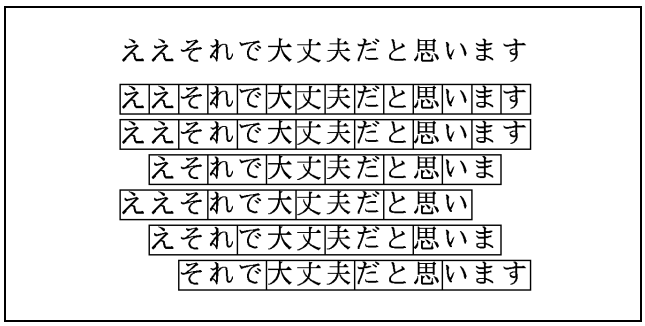

Figure 2: Examples of string patterns in a sentence.

2. N-gram by string pattern

A written Japanese sentence is regarded as a sequence of characters. In the conventional word segmentation by morphemic analysis, word boundaries of the sequence are decided using a dictionary and rules, which are given a priori. In the proposed method, items in the dictionary are chosen from the training text using a statistical criterion, and simple longest-match rule is applied for the segmentation.

Figure 2 shows a Japanese sentence and substrings in it. Up to 3 character strings are shown in the example. In the proposed method, string patterns in the training sentences are scored according to a statistical criterion. Up to 6 character strings are considered as patterns in the later experiment. Then several hundreds of high-score patterns are selected as pseudo-words. Using the selected patterns, longest-match based pattern matching is applied to training and test sentences. Parts of the sentence to which no pseudo-word matches are divided character-by-character. After the division, N-gram model of string pattern and character sequence is constructed.

For example, using frequency-based 250 patterns (described later), the sentence in Figure 2 is divided as

え|え|それで|大|丈|夫|だ|と|思います

This result differs from traditional word based analysis in Figure 1. While words with low frequency (for example, "大丈夫(*okay*)") are not regarded as a single pattern, some idioms (for example, "思います(*think*, with an auxiliary verb of honorific form)") are recognized as a single pattern.

We examined following three criteria to select string patterns.

1. Frequency of string pattern. First, count occurrence of all patterns in the training sentences. Overlapping patterns are counted individually. Then patterns with high frequency are selected.
2. Normalized frequency of string pattern. Patterns with high frequency tend to be short, while meaningful patterns are longer than them. The normalized frequency is proposed to compensate the effect of pattern length. The normalized frequency of pattern x is calculated as

$$NF(x) = F(x) \sum_{i=1}^{|x|} r_i \tag{1}$$

where $F(x)$ is the frequency of x, $|x|$ is the length of x and r_i stands for the number of patterns with length i.

3. Pattern distance. First, measure statistical distance between each patterns in the training sentences. We used divergence[5] as distance of two patterns. Let p_1, p_2 be string patterns, and c be a context the patterns occur. Then distance of two patterns is defined as

$$d(p_1, p_2) = \sum_c d(p_1, p_2; c)$$

$$d(p_1, p_2; c) = \begin{cases} (P(c|p_1) - P(c|p_2)) \log \frac{P(c|p_1)}{P(c|p_2)} \\ \quad \text{if} \quad P(c|p_1) > 0 \text{ and} \\ \quad \quad P(c|p_2) > 0 \\ \alpha \quad \text{otherwise} \end{cases} \tag{2}$$

We used characters right before and after the pattern as a context[6]. α is a penalty value and is set to 10 in the later experiment. After calculating all distances, pattern pairs with small distances are chosen. String patterns in the selected pairs are regarded as pseudo-words.

3. Experiment 1 — Comparison between models

We carried out an experiment to compare the traditional models and the proposed model. Compared models are as follows:

- Word bigram and trigram
- Character bigram and trigram
- Bigram and trigram of character and string pattern. The patterns are selected by frequency, normalized frequency or pattern distance.

To calculate the probability of unseen N-gram, linear back-off[7] was employed. To evaluate each model, adjusted perplexity[8] was used instead of ordinary task perplexity. All perplexity values in the result were calculated character-by-character.

We used human-to-human dialog corpus from ASJ continuous speech database for training and testing the models. Size of the corpus is shown in Table 1. This corpus was divided into words manually to construct the word N-gram.

The results of the experiment are shown in Table 2. "avg. len." in the table shows the average length of patterns in

Table 1: Text corpus used in experiment 1

	training	test
# dialog	44	5
# sentence	3606	649
# phrase	19031	2446
# word	52588	6698
# character	87520	11278

Table 2: Results of experiment 1

model		APP		avg.	npat
		bigram	trigram	len.	
word		61.34	61.10	1.65	3922
character		52.66	37.34	1.00	1378
freq	125	43.19	36.18	1.22	1444
	250	40.42	36.34	1.33	1524
	500	38.13	36.74	1.46	1690
	1000	38.22	38.41	1.62	2007
	2000	40.64	41.89	1.83	2607
	4000	44.15	46.44	2.10	3665
	8000	55.76	58.81	2.41	5279
norm. freq	125	42.52	36.50	1.27	1482
	250	40.50	36.50	1.37	1593
	500	39.40	37.53	1.50	1812
	1000	39.40	38.85	1.66	2198
	2000	41.07	42.52	1.87	2912
	4000	46.21	48.17	2.13	4022
	8000	58.08	61.39	2.51	5893
pattern distance	1000	42.58	36.98	1.27	1522
	2000	41.12	37.82	1.38	1668
	4000	39.94	38.58	1.47	1862
	8000	40.53	39.95	1.58	2145

the training text. "npat" stands for the number of kind of pattern (word, character or string pattern) in the training text. The best result was obtained when 125 patterns are selected using frequency-based criterion. Differences between frequency, normalized frequency and pattern distance were relatively small. To our surprise, word bigram and trigram were inferior to character N-gram. This result is caused by the adjustment of perplexity. When the test sentences were evaluated using word N-gram, many unknown words (2100 words) were observed, which made the perplexity worse. In the character N-gram, the number of unknown characters was only 93. Another reason is that the average length of a word is short (1.65). This fact causes character trigram (length 3) comparable to word bigram (average length 3.3). This result means that character N-gram model is more robust than word N-gram, and string pattern N-gram is better than them.

Table 3: Text corpus used in experiment 2

	training	test
# sentence	187007	20778
# word	4148956	461709
# character	6806464	758794

Table 4: Result of experiment 2

model		APP		avg.	npat
		bigram	trigram	len.	
word		36.00	32.22	1.641	118626
character		54.41	28.11	1.000	3494
string pattern	250	46.88	27.13	1.122	3565
	1000	40.39	25.79	1.265	3967
	2000	36.28	25.55	1.396	4709
	4000	32.74	25.90	1.573	6393

4. Experiment 2 — Effect of corpus size

As the corpus used in experiment 1 was not very large, we investigate the performance of the proposed method for larger corpus. We used EDR corpus, which has about 5 million Japanese words. We used the training and test sentences shown in Table 3. In this experiment, traditional word N-gram, character N-gram and the proposed model were compared. The frequency-based criterion was used to select string pattern, which showed best performance in the last experiment.

The experimental result is shown in Table 4. The best result was obtained when 2000 patterns are selected using the proposed model. This result proved that the perplexity of the proposed model was lower than the traditional models even for the larger corpus. Moreover, the improvement was greater than the result for the small corpus.

5. Experiment 3 — Application to English text

The proposed model groups characters into a pattern – a pseudo-word. This model can be naturally applied to group words into a pseudo-phrase. Several models are proposed to group words automatically[5, 9]. As the proposed model is much simpler than them, it can be applied to very large corpus.

Figure 3 shows an English sentence and subsequences of word (word string patterns) contained in the sentence. Same as the model explained in experiment 1, occurrence of all word string pattern up to 6 words in the training sentence is counted. Then the patterns with high frequency are extracted as pseudo-phrase, and the longest-match based analysis is carried out on the training and test sentences.

We examined to make pseudo-phrases using small corpus.

Figure 3: Examples of word string patterns in an English sentence.

Table 5: Text corpus used in experiment 3

	training	test
# dialog	221	23
# sentence	6874	755
# word	71933	7648

The training and test sentences are English part of ATR speech database. This corpus is a collection of keyboard dialogs about international conference. The size of the corpus is shown in Table 5.

The experimental result is shown in Table 6. In this experiment, the best result was obtained when 250 patterns are selected. While the improvement was not very drastic in this experiment, we expect more improvement for larger corpora. Examples of selected word string pattern are shown in Figure 4.

Table 6: Result of experiment 3

model		APP		avg.	npat
		bigram	trigram	len.	
word		188.79	158.10	1.000	3815
word string pattern	125	175.24	155.87	1.092	3844
	250	165.19	155.67	1.165	3910
	500	164.20	161.59	1.235	4025
	1000	164.18	166.81	1.324	4288
	2000	174.08	179.88	1.449	4909

of the	on the	i will
the conference	i have	we are
thank you	the secretariat	of the conference
like to	for the	you very much
will be	to the	you very
this is	for your	thank you very
at the	i 'll	much
i 'm	if you	thank you very
may i	we will	secretariat of
in the	would like to	computer
i am	very much	conference
would like	is the	the secretariat of

Figure 4: Examples of selected word string pattern

6. Summary

A new statistical language model for Japanese text is proposed. This model uses string patterns instead of words and characters for the unit of N-gram. From the experimental results, it was found that the simplest frequency-based selection of string pattern was best, and more improvement was obtained for the larger corpus. We applied this model to English text, which improved the traditional word-based trigram.

7. REFERENCES

1. K. Ohtsuki *et al.* : "Study of Large-Vocabulary Continuous Speech Recognition using Read-Speech Data", Tech. Rep. IEICE, NLC95-55, pp. 63–68 (1995-12) (In Japanese)
2. T. Kawabata *et al.* : "Japanese Phonetic Typewriter using HMM Phone Recognition and Stochastic Phone-sequence Modeling", IEICE Trans. Information and Systems, Vol. E74, No. 7, pp.1783–1787 (1991)
3. T. Araki *et al.*: "Effect of Reducing Ambiguity of Recognition Candidates in Japanese "Bunsetsu" Units by 2nd-order Markov Model of Syllables", Trans. IPSJ, vol. 30, No. 4, pp. 467–477 (1989-4) (in Japanese)
4. T. Yamada *et al.*: "Speech Recognition Using Stochastic Language Models Based on a Prior Probabilities of Kana and Kanji Characters", Trans. IEICE (A), vol. J77-A, No. 2, pp. 198–205 (1994-2) (in Japanese)
5. M. K. McCandless *et al.* : "Empirical Acquisition of Language Models for Speech Recognition", Proc. ICSLP-94, pp. 835–838 (1994)
6. F. Jelinek *et al.*: "Classifying Words for Improved Statistical Language Models", Proc. ICASSP-90, pp. 621–624 (1990)
7. P. Placeway *et al.*: "The Estimation of Powerful Language Models from Small and Large Corpora", Proc. ICASSP-93, vol. II, pp. 33–36 (1993)
8. J. Ueberla: "Analysing a Simple Language Model – Some General Conclusion for Language Models for Speech Recognition", Computer Speech and Language, vol. 8, No. 2, pp. 153–176 (1994-4)
9. S. Deligne *et al.*: "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams", Proc. ICASSP-95, vol.I, pp. 169-172 (1995)