# MULTI-CLASS COMPOSITE N-GRAM BASED ON CONNECTION DIRECTION

*Hirofumi Yamamoto and Yoshinori Sagisaka*

ATR Interpreting Telecommunications Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
E_mail : yama@itl.atr.co.jp

## ABSTRACT

A new word-clustering technique is proposed to efficiently build statistically salient class 2-grams from language corpora. By splitting word neighboring characteristics into word-preceding and following directions, multiple (two-dimensional) word classes are assigned to each word. In each side, word classes are merged into larger clusters independently according to preceding or following word distributions. This word-clustering can provide more efficient and statistically reliable word clusters. Further, we extend it to Multi-Class Composite N-gram that unit is Multi-Class 2-gram and joined word. Multi-Class Composite N-gram showed better performance both in perplexity and recognition rates with one thousandth smaller size than conventional word 2-grams.

## 1.INTRODUCTION

Word N-grams have been widely used as a statistical language model for continuous speech recognition. Though word N-grams are more effective and flexible than rule-based grammatical constraints in many cases, they require huge amount of memory compared to grammatical rule expression. Sometimes, N-gram size can be a bottleneck to reduce on-line memory size for down-sizing the recognition system.

From statistical viewpoint, it is quite frustrating to keep redundant information embedded in N-grams and statistically obscure scores for word pairs with small occurrences. Some ideas such as class N-grams and variable length N-grams [1]-[4] can be served to resolve this data reduction problem.

In all these studies, word-class definition is quite crucial to reduce model size without losing its accuracy. For example, word 2-grams made from a Japanese language corpus consisting of 459 383 words including 7 221 different words give a perplexity of 18.51, whereas class 2-grams can only give a perplexity of 31.53 with two-thousandth smaller size if 158 word classes are chosen simply as conventional part-of-speech categories. To build a more accurate word class N-grams, word clustering is to be considered.

In this paper, two-dimensional word-clustering is newly proposed by splitting word neighboring characteristics into following and preceding characteristics independently. This word clustering technique is applied to the generation of Multi-Class Composite N-grams consisting of word classes, frequent words and their successions to build Multi-Class Composite N-grams. Finally recognition results are shown. It is confirmed that this model can provide better performance than conventional word 2-grams with one-thousandth smaller model size.

## 2.MULTI-CLASS

### 2.1. Multi-Class

Let us consider the difference between clustering by part-of-speech and clustering based on only connections. Take "a" and "an" as an example. Both are classified by part-of-speech as the Indefinite Article, and are assigned to the same class. In this case, we've lost information about the difference in the words that can follow it. If we consider the type of word preceding it and the type of word following it as different attributes, then because the connectivity of the word preceding "a" and "an" is the same, it's no problem to assign them to the same class. The connectivity for the word following them is different (starting with vowel vs. starting with consonant), so they must be assigned to different classes. So we use 3 classes to express these words' connectivities. The class that expresses the connectivity of the preceding word we call the "to-class", and the class for the connectivity of the following word the "from-class". As another example, "is" and "are" would be assigned to different to-classes, but the same from-class.

This can be extended to N 3, following the same line of thought, and in this case each word would have N classes as attributes. Attributing classes like this we'll call Multi-class, and an N-gram with this we'll call a Multi-class N-gram. When N=2, the probability of $W$ directly following $W_{n-1}$ is described by the following equation.

$$P(C_t(W_n) \mid C_f(W_{n-1})) \times P(W_n \mid C_t(W_n)) \qquad (1)$$

where $C_t$ represents the to-class and $C_f$ represents the from-class. In this case the number of parameters for the Multi-Class 2-gram is the number of to-classes multiplied by the number of from-classes plus the number of words; for the class 2-gram is the square of the number of classes plus the number of words.

### 2.2. Multi-Class:Automatic Clustering

Classification by part-of-speech is not always optimal for N-grams; clustering by only the property of the connection is

more beneficial. So we decided to automatically extract from the corpus the classes which were clustered by connection characteristic only. Other ways of clustering have been proposed (e.g.[1]); we chose the following method:

1. Assign one class per word.
2. Assign a vector to each class or to each word X. It represents the characteristics of the connection.
   $$V_t(X) = \{P_t(W_1 \mid X), P_t(W_2 \mid X) \ldots P_t(W_n \mid X)\} \quad (2)$$
   $$V_f(X) = \{P_f(W_1 \mid X), P_f(W_2 \mid X) \ldots P_f(W_n \mid X)\} \quad (3)$$
   For a class 2-gram, they must be expressed together
   $$V(X) = \{V_t(X), V_f(X)\} \quad (4)$$
   For Multi-Class 2-gram, the to-class and from-class are expressed separately:
   $$V(X) = V_t(X) \quad (5)$$
   $$V(X) = V_f(X) \quad (6)$$
   In the above equations, $P_t$ is the value of the probability of the succeeding class-word 2-gram or word 2-gram; $P_f$ is the same for the preceding.
3. Merge 2 classes. we choose the classes whose dispersion weighted with the 1-gram probability results in the lowest rise, and merge those 2 classes:
   $$U_{old} = \sum_W P(W) \times D(V(C_{old}(W)), V(W)) \quad (7)$$
   $$U_{new} = \sum_W P(W) \times D(V(C_{new}(W)), V(W)) \quad (8)$$
   where we merge the classes whose $U_{new} - U_{old}$ is lowest. Here, $D$ represents the square of the Euclidean distance between vectors, $C_{old}$ represents the classes before merging, and $C_{new}$ represents the classes after merging
4. Repeat step 2 until the number of classes is reduced to the desired number.

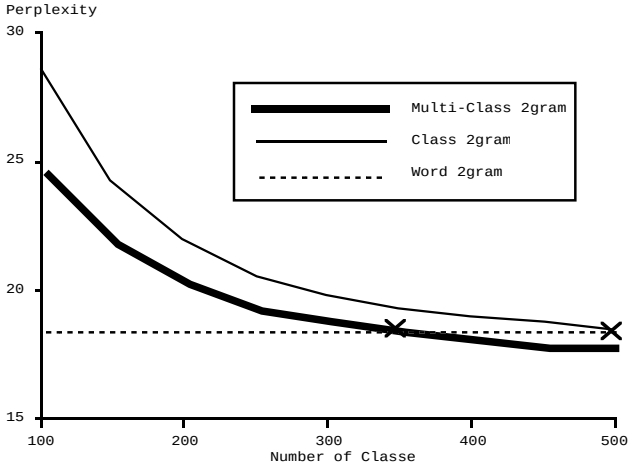


Figure 1: Perplexity of Multi-Class 2-gram

We evaluated the perplexity of a Multi-Class 2-gram and class 2-gram created by this method. For the Multi-Class 2-gram, the number of to-classes and the number of from-classes need not be the same, but for the purposes of comparison, we set them to be equal. The result was that class 2-gram with 500 classes and a Multi-Class 2-gram with 350 classes had about the same perplexity as a word 2-gram. As can be seen in detail in figure 1, the Multi-Class 2-gram requires fewer classes than the class 2- gram for the same performance.

### 2.3. Continuous Word Recognition

The perplexity does not directly determine the recognition rate for continuous word recognition. We evaluated the performance of a Multi-Class 2-gram in the continuous word recognition experiment. We use a Multi-Class 2-gram with 350 classes, a class 2-gram with 500 classes and a word 2-gram, which all have about the same perplexity. The results were that the Multi-Class 2-gram gave 70.29%, the class 2-gram gave 69.78%, the word 2-gram gave 69.05%. This shows that, for continuous recognition, a Multi-Class 2-gram for the same perplexity can yield the even or better recognition results while using fewer parameters. The conditions of this experiment follow.

Test set: 41 conversations with 536 utterances
Acoustic model: 800 state speaker independent HMnet [5]
Decoder: Frame synchronized 2pass search [6]

## 3 MULTI-CLASS COMPOSITE N-GRAM

### 3.1. Introducing Joined Words

The word unit appropriate for an N-gram may not be the same as the word unit best for recognition, because in Japanese, sometimes a special kanji character sequence gives rise to a special phoneme sequence, just as the word sequence "going to" becomes "gonna" in American English. In this case, the lexicon needs addition of a special entry, and this requires retraining of the usual N-gram; this could create a new data sparseness problem. The Multi-Class 2-gram would not require retraining for the addition of the new entry, and thus causes no problem with sparse data. The probability of occurrence of word $X$ followed by the joined words $A+B$ and then by $C$ is:

$$P(C_t(A+B) \mid C_f(X)) \times P(A+B \mid C_t(A+B)) \times P(C_t(C) \mid C_f(A+B)) \times P(C \mid C_t(C)) \quad (9)$$

The kinds of words that can precede $A+B$, i.e. $C_t(A+B)$, are the same as what can follow $A$, i.e. $C_t(A)$; the kinds of words that can come after the combined word $A+B$, i.e. $C_f(A+B)$, are the same as those that can come after $B$, i.e. $C_f(B)$. Therefore,

$$C_t(A+B) = C_t(A) \quad (10)$$
$$C_f(A+B) = C_f(B) \quad (11)$$

and eq.9 is transformed as follows

$$P(C_t(A) \mid C_f(X)) \times P(A+B \mid C_t(A)) \times P(C_t(C) \mid C_f(B)) \times P(C \mid C_t(C)) \quad (12)$$

where

$$P(A+B \mid C_t(A)) = P(A \mid C_t(A)) \times P(B \mid A) \quad (13)$$

$P(B \mid A)$ for the Multi-Class 2-gram, according to eq.1, is

$$P(B \mid A) = P(C_t(B) \mid C_f(A)) \times P(B \mid C_t(B)) \quad (14)$$

so eq.12 becomes

$$P(C_t(A) \mid C_f(X)) \times P(A \mid C_t(A)) \times$$
$$P(C_t(B) \mid C_f(A)) \times P(B \mid C_t(B)) \times$$
$$P(C_t(C) \mid C_f(B)) \times P(C \mid C_t(C)) \quad (15)$$

The parameters needed by introduction of the joined $A+B$ can be obtained by eq.15 without any retraining of the Multi-Class 2-gram. Furthermore the increase in parameters is just a 1-gram in the to-class of the joined word $A+B$.

### 3.2. Multi-Class Composite N-gram

Even when a class 2-gram is used because of insufficient data for a word N-gram, some of the word pairs may have enough data for using a word N-gram. Where the rate of occurrence of word pairs is adequate, the word 2-gram can be used because it is more reliable; and where there aren't enough, the class 2-gram can be used. When the number of occurrences of the word sequence $A, B$ is sufficient, the probability that the word sequence $A, B, C$ occurs after the word $X$ is

$$P(C(A) \mid C(X)) \times P(A \mid C(A)) \times P(B \mid A) \times$$
$$P(C(C) \mid C(B)) \times P(C \mid C(C)) \quad (16)$$

and, when the Multi-class 2-gram is used, eq.16 becomes

$$P(C_t(A) \mid C_f(X)) \times P(A \mid C_t(A)) \times P(B \mid A) \times$$
$$P(C_t(C) \mid C_f(B)) \times P(C \mid C_t(C)) \quad (17)$$

where, as described in the previous paragraph, a Multi-Class 2-gram does not require the introduction of a new class when 2 words are joined. This feature (eq.10 eq.11) can be used, and results in

$$C_t(A) = C_t(A+B) \quad (18)$$
$$C_t(A) = C_f(A+B) \quad (19)$$

and eq.17 becomes

$$P(C_t(A+B) \mid C_f(X)) \times P(A+B \mid C_t(A+B)) \times$$
$$P(C_t(C) \mid C_f(A+B)) \times P(C \mid C_t(C)) \quad (20)$$

which shows that the form of the Multi-Class 2-gram is maintained even with the introduction of a word 2-gram, and it can be represented as a joined word; the new parameter consists only of the 1-gram of the joined word. This is the same for any word 3-gram (or any N 3): the word consisting of the joined 3 words is introduced. In specific terms, this results in the following algorithm to create a model.

1. Assign a Multi-Class 2-gram, for state initialization.
2. Find word pair whose rate of occurrence is above a threshold.
3. Create joined word from word pair. The to-class of this joined word is the same as the to-class of the first word in the pair; its from-class of it is the same as the from-class of the last word in it.
4. Add to the lexicon the joined word.
5. Repeat step 2 with the newly added joined words.
6. Stop when nothing remains whose rate of occurrence is above the threshold.

The model obtained in this way is called the Multi-Class Composite N-gram.

### 3.3. Variable Order N-Gram Comparison

The Variable Order N-gram [2] has been introduced as a model to fill in for the weaknesses of the class 2-gram and word N-gram. The Variable Order N-gram is based on the class 2-gram; it splits words from a class to make a separate class; and it joins these separated words to make joined words and introduces them as new classes, all with the purpose of reducing the entropy. This procedure is repeated. The Variable Order N-gram performs well but has the following problems which are solved by the Multi-Class Composite N-gram.

- Because words can't be joined unless they have been split, even optimal clustering must be split.
- A word that has been split usually has sufficient rate of occurrence(1-gram), but this may not have sufficient rate for 2-gram; and there is the sparse data problem.

Table 1 summarizes the difference between the Multi-Class Composite N-gram and the Variable Order N-gram. The Variable Order N-gram has more freedom in the units of expression than the Multi-Class Composite N-gram, but when the initial class setting is optimal, then there's little difference between the class 2-gram and the class-word 2-gram or word-class 2-gram, so the comparison is valid. Furthermore, the criterion for splitting in the Variable Order N-gram could be rate of occurrence, and for the Multi-Class Composite N-gram it could be the entropy, so the comparison is valid.

Table 1: Variable Order N-gram vs. Multi-Class Composite N-gram

| | Variable Order N-gram | Multi-Class Composite N-gram |
|---|---|---|
| target of splitting | from class to word | from class 2 gram to word 2 gram |
| criterion for splitting | reducing the entropy | rate of occurrence of word pair |
| unit of expression of N-gram | Class 2-gram<br>Class-Word 2-gram<br>Word-Class 2-gram<br>Word N-gram | Class 2-gram<br>Word N-gram |
| increase in number of parameters | number of split words +<br>square of number of joined words | number of joined words |

### 3.4. Evaluation of Perplexity

First, in order to evaluate the performance of the Multi-Class Composite N-gram, we evaluated the perplexity. Initialization

of the Multi-Class Composite N-gram was done by automatic clustering, and the rate of occurrence criterion for joining words was 20 times. Figure 2 shows the results for a training set of 7 360 unique words (459 383 words total) and a test set of 6 826 words total. This figure shows that at about 400 classes the Multi-Class composite 2-gram yielded a performance just about equal to a word 3-gram. The total number of joined words was 2 122, and the number of occurrences of these, i.e. the number of word N-grams used, was 116 525, or about 20% of the total. The number of occurrences of joined words of length 3 or more - the number of word 3-grams or higher - was 60 529, or about 10% of the total.

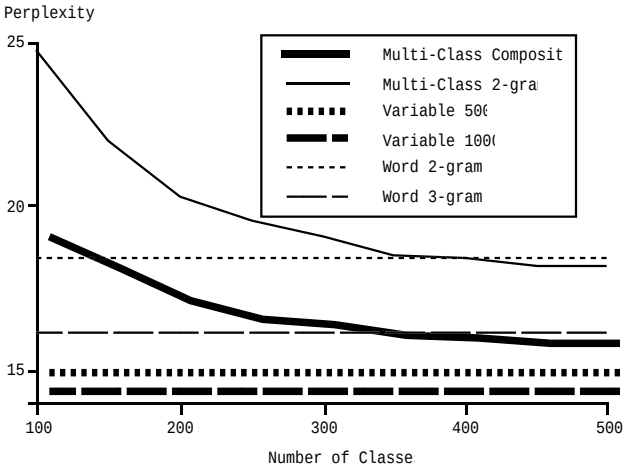


Figure 2: Perplexity of Multi-Class Composite N-gram

### 3.5. Continuous Word Recognition

The perplexity suggests but does not determine the continuous word recognition rate. So we performed a continuous word recognition experiment (conditions are same as section 2.3.), and evaluated the performance for the Multi-Class Composite N-gram. We did this against a Variable Order N-gram with 1 000 split classes( i.e. 1 158 classes in total), and a Multi-Class Composite N-gram with 100, 200, 300, and 400 classes. The recognition results are shown in table 2, which shows that the Multi-Class Composite N-gram with 200 classes gave about the same results as the Variable Order N-gram. Notice that the Multi-Class Composite N-gram uses only about 4% of the parameters of the Variable Order N-gram, or about 1/1000 of a word 2-gram.

Table 2: Performance of Multi-Class Composite N-gram

| Number of Class | Parameter Size | Perplexity | Word Accuracy |
|---|---|---|---|
| 100 | 19 433 | 19.62 | 74.47 |
| 200 | 49 433 | 17.54 | 76.30 |
| 300 | 99 433 | 16.83 | 74.89 |
| 400 | 169 433 | 16.29 | 75.79 |
| Variable Order N-gram | 1 348 426 | 14.84 | 75.51 |

## 4. CONCLUSION

In this paper, a new word-clustering technique is proposed to build efficient Multi-Class Composite N-grams. This modeling enables huge data reduction of model size without loosing perplexity reduction using efficient clustering of neighboring word characteristics. Recognition experiments also supported the usefulness of this model. Though the modeling experiment was only carried out using continuous speech recognition of Japanese, it is expected that this modeling will be effective for other languages by considering language independent statistical formulation. This modeling is not only be useful to conventional model size reduction as shown in this paper but it is also expected that the idea of multiple assigning the word attributes (in this case, two-dimensional preceding and following word neighboring characteristics) would be useful to simplify and newly extract linguistic statistics from corpora.

## ACKNOWLEDGMENTS

We'd like to thank Ben Reaves for assistance in writing some of the explanations in this paper.

## REFERENCES

[1] Shuanghu Bai, Haizhou Li, Zhiwei Lin, Baosheng Yuan : "Building Class-based Language Models with Contextual Statistics, "Proc.ICASSP, vol. 1, pp. 173-176, 1998

[2] Hirokazu Masataki, Yoshinori Sagisaka : "Variable-Order N-gram Generation by Word-Class Splitting and Consecutive Word Grouping, "Proc.ICASSP, vol. 1, pp. 188-191, 1996

[3] P.F.Brown et al :"Class-Based n-gram Models of Natural Language, "Computational Linguistics,vol. 18, No. 4, pp. 467-479, 1992

[4] Sabine Deligne, Frederic Bimbot :"Language modeling by variable length sequences, "Proc ICASSP, vol. 1, pp. 169-172, 1995

[5] M.Ostendorf, H.singer :"HMM topology design using maximum likelihood successive state splitting, " Computer Speech and Language, vol. 11, pp. 17-41, 1997

[6] Tohru Shimizu, Hirofumi Yamamoto, Hirokazu Masataki, Shoichi Matsunaga, Yoshinori Sagisaka : "Spontaneous Dialogue Speech Recognition Using Cross-Word Context Constrained Word Graphs," Proc.ICASSP, vol. 1, pp. 145-148, 1996