音声言語情報処理 19－17
(1997. 12. 12)

意味・構文クラスタリングを用いた統計文法の獲得

荒井和博† Jeremy H. Wright‡
Giuseppe Riccardi‡ Allen L. Gorin‡

† NTT ヒューマンインタフェース研究所 ‡ AT&T Laboratories-Research
〒239 神奈川県 横須賀市 光の丘 Florham Park, NJ 07932, USA
arai@nttspch.hil.ntt.co.jp {jwright,dsp3,algor}@research.att.com

あらまし
本稿では意味的にも構文的にも類似した単語列の集合をクラスタとして生成し，自由発話の音声理解に用いるクラスタリングアルゴリズムについて述べる．本手法では，まず学習データ内で高い頻度で観測される単語列がクラスタリング候補として選択される．次に選択された候補単語列のそれぞれに対して，先行コンテキスト，後続コンテキスト及び対話システム応答に関する確率分布が求められる．候補単語列間の類似性は 3 種類の確率分布それぞれの Kullback-Leibler 距離によって定義され，各候補単語列間に 3 つの距離が求められる．単語列のクラスタリングにおいては，3 種類の距離がいずれも短い単語列どうしが同一のクラスタにまとめられ，単語列クラスタが構成される．本手法を AT&T の電話サービス案内タスクに適用したところ，学習データ内では観測されず評価データ内のみで観測された単語列を 246 個生成できた．また，コールタイプ識別率 (音声理解率) が約 3% 改善され，本手法の有効性が確認された．

キーワード 音声理解, 先行・後続コンテキスト, Kullback-Leibler 距離, 単語列の類似性, 単語列クラスタリング

Grammar Fragment Acquisition using Syntactic and Semantic Clustering

Kazuhiro Arai† Jeremy H. Wright‡
Giuseppe Riccardi‡ Allen L. Gorin‡

† NTT Human Interface Laboratories ‡ AT&T Laboratories-Research
Yokosuka, Kanagawa 239, Japan Florham Park, NJ 07932, USA
arai@nttspch.hil.ntt.co.jp {jwright,dsp3,algor}@research.att.com

Abstract
A new method for automatically acquiring grammar fragments for understanding fluently spoken language is proposed. The goal of this method is to generate a collection of grammar fragments each representing a set of syntactically and semantically similar phrases. First phrases observed frequently in the training set are selected as candidates. Each candidate phrase has three associated probability distributions: of succeeding contexts, of preceding contexts, and of associated machine actions. The similarity between candidate phrases is measured by applying the Kullback-Leibler distance to three probability distributions. Candidate phrases which are close in all three distances are clustered into a grammar fragment. This approach detected 246 phrases in the test-set that were not present in the training-set. Experimental results show that a 3% improvement in the call-type classification performance has been achieved by introducing these fragments.

key words spoken understanding, preceding and succeeding contexts, Kullback-Leibler distance, phrase similarity, phrase clustering

1 Introduction

In stochastic language modeling the probability of a word occurrence is generally calculated based on its frequency obtained from training transcriptions. This approach can be applicable to not only a word but also a phrase. In fact, frequencies of some phrases are higher than that of some words barely observed in a training set. It is therefore reasonable that a phrase can be regarded as a unit for language modeling [Riccardi97] [Giachin95] [Masataki96]. The phrase-based N-gram makes it possible to estimate the transition probability accurately between phrases observed frequently in the training set. On the other hand, other language models have been proposed based on the word-class N-gram [Bellegarda96] [Farhat96] [Ward96]. In this approach, all words in a training set are first clustered into some word classes. Words having a similar transition probability are generally clustered into the same word class. Because a word sequence is represented as a sequence of word classes, this approach makes it possible to estimate the probability for a word transition not observed in the training set.

These approaches however focus on succeeding words or phrases only to minimize the branching factor or the test set perplexity. Since analogous phrases have a similar distribution in not only the succeeding-word sequence but also in the preceding-word sequence, the similarity of word sequences can be clustered more effectively by referring to both of the succeeding and preceding word sequences. Furthermore, an utterance accepted as input of our call-router dialog system generally carries what the user requested or the *call-type* [Gorin97]. Some phrases carrying the analogous role in the dialog must have similar associations with the call-types. Thus the word sequence similarity can be also computed by using such associations between phrases and call-types [Wright97]. This paper proposes a new method for clustering phrases into grammar fragments. This method focuses on not only succeeding words but also on preceding words and on the call-types associated to each utterance for generating grammar fragments comprising similar phrases.

2 A Spoken Dialog System for A Call-Router

2.1 Call-type Classification

The goal of our dialog system for call-routing is to understand its input sufficiently to route the caller to an appropriate destination in a telecommunications environment. There are 14 call-types and an *other* class as a complement. Since in many situations the call-type can not be exactly determined from a single input, dialog is often necessary. Once the call-type has been successfully negotiated and agreed upon, there is often a subsequent form-filling dialog to complete the information necessary to provide a service. The most important issue in this kind of dialog is that it is not necessary to perfectly recognize and understand every word accurately. It has been shown that the dialog system can determine the call-type sufficiently if it can extract the some phrases strongly associated with a call-type.

2.2 Database

In order to enable experiments, a database of 10K spoken transcriptions between customers and human agents were generated as detailed in [Gorin97]. We focused on the first customer utterance, responding to the greeting prompt of *"How may I help you?"*. These utterances were endpointed, transcribed and labeled as to the call-type and quality of the speech and channel. The transcriptions were split into three subsets for training (8K), developing (1K) and testing (1K) the acoustic and language models for recognition and understanding. In the training set, there are 3.6K words which define the vocabulary.

3 Fragment Distance

3.1 Phrase and Fragment

An arbitrary word sequence in the training transcriptions is called a *phrase*. All phrases can be obtained by decomposing the transcriptions into n-tuple word sequences. Namely, each phrase is a substring of a sentence. The number of words in a phrase is constrained to be three or less in this experiment. The phrases having higher frequency than some thresholds are selected as *candidates*. The candidate phrases are regarded as units for generating the *grammar fragments*. Each grammar fragment is acquired via clustering of candidate phrases based on their similarity and is represented as a conventional finite-state machine.

A *Fragment grammar* is generated by using the grammar fragments. The fragment grammar furthermore defines the association between phrase and call-type. Examples of the association with the call-type, the distance calculation, and the phrase clustering algorithm are shown using only phrases in Sections 3 and 4. We remark, however, these can generally be applied in a straightforward manner to grammar fragments or sequences thereof.

3.2 Syntactic and Semantic Associations of a Fragment

There are many standpoints for the linguistic terms *syntax* and *semantics*, so that we first clarify our usage of these terms in this section. In this discussion, the *syntactic*

association signifies the relationship between a grammar fragment and phrases succeeding or preceding the fragment. If the roles of fragments are similar to each other in spoken dialog, then the distribution of these phrases will be similar between the fragments. On the other hand, the *semantic association* focuses on the association between a fragment in spoken language and the call-type corresponding to the speech. The distribution of call-types for a fragment must be comparable to that for another fragment, if the two fragments are to be clustered. The *semantic association* is therefore the cross-channel association between speech and call-types.

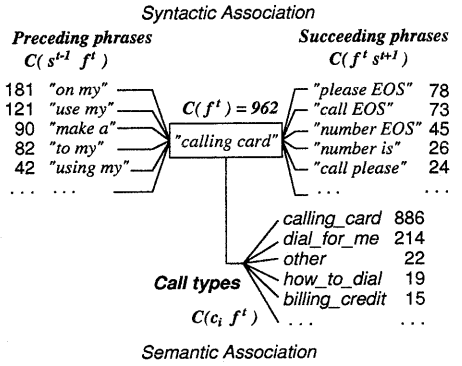


Figure 1: Syntactic and Semantic Association of a Fragment

An example of the syntactic and semantic associations of a fragment is illustrated in Figure 1. The symbol f denotes a grammar fragment, p, s and c define preceding or succeeding phrases and call-types respectively. In Figure 1, f comprises only one phrase *"calling card"*. Suffixes such as t and $t+1$ means order in sequence. Given a phrase, fragment, call-type or combination thereof as an argument, the function $C(\ )$ counts the frequency of the argument in the training transcriptions. For instance, $C(f^t\ s^{t+1})$ denotes the frequency of the fragment f followed by a phrase s. *BOS* and *EOS* denote *Beginning-Of-Sentence* and *End-Of-Sentence*, respectively. Figure 1 shows that the fragment *"calling card"* was observed 962 times in the training transcriptions. The phrase *"on my"*, for instance, preceded that fragment 181 times and *"number is"* succeeded it 26 times. The call-type *calling_card* was associated with the fragment 886 times, which is comparatively high frequency among the call-types. By counting preceding and succeeding phrases, two syntactic probability distributions for each fragment are obtained. The call-type probability distribution for each fragment is also obtained by counting call-types assigned to transcriptions in which the fragment is observed. This call-type probability distribution represents a semantic feature vector for a fragment.

In order to generate syntactic probability distributions, a set of phrases which precedes or succeeds fragments is generated first. In the following discussion, a phrase that succeeds or precedes a fragment is called its *context*. Though in our experiments the context consists of single words, our algorithm can be applied to the longer context so we describe the method in the general case. Consequently the context can contain words and non-terminal symbols corresponding to grammar fragments. The number of words and non-terminal symbols in each context is equalized for computation of syntactic probability distributions. A *context-frequency list* S for all fragments is then generated by storing all phrases of length N_c together with their frequencies. Using S, a unigram probability distribution is generated. This unigram probability distribution is utilized for back-off smoothing the syntactic probability distribution of each fragment. A set of call-type frequencies is also obtained from training transcriptions and can be utilized for smoothing the semantic probability distribution.

Three probability distributions for each grammar fragment are obtained by using preceding and succeeding context frequencies, and call-type frequency. The bigram probability distributions focusing on succeeding and preceding contexts are denoted in Equations (1) and (2), respectively.

$$p(s_i^{t+1}|f_j^t) = \frac{C(\ f_j^t\ s_i^{t+1}\ )}{C(\ f_j\ )} = \frac{C(\ f_j^t\ \ w_1^{t+1}w_2^{t+2}\cdots w_{N_c}^{t+N_c}\ )}{C(\ f_j\ )} \tag{1}$$

$$p(s_i^{t-1}|f_j^t) = \frac{C(\ s_i^{t-1}\ f_j^t\ )}{C(\ f_j\ )} = \frac{C(\ w_1^{t-N_c}\cdots w_{N_c-1}^{t-2}w_{N_c}^{t-1}\ \ f_j^t\ )}{C(\ f_i\ )} \tag{2}$$

In both Equations (1) and (2), s_i denotes the i-th context stored in the context frequency list S, f_j is the j-th grammar fragment in the fragment grammar, w_k denotes the k-th word in the context s_i, N_c ($N_c \geq 1$) is the number of items referred to as the context. Suffix such as t, $t+1$ and $t-1$ denotes order in sequence of word, context, or fragment. The function $C(\ )$ counts frequency of a sequence in the training transcriptions as described in 3.2.

The context s_i^{t+1} and s_i^{t-1} are equivalent to word sequences $w_1^{t+1}w_2^{t+2}\cdots w_{N_c}^{t+N_c}$ and $w_1^{t-N_c}\cdots w_{N_c-1}^{t-2}w_{N_c}^{t-1}$, respectively. The larger the parameter N_c is set, the more the variety in context can be theoretically observed, because these Equations refers to not word but context. In practice, however, these probability distributions become sparse when the parameter N_c is large. Therefore, the parameter N_c generally should be determined based on the size of the training transcriptions. These two probability distributions represent syntactic feature vectors of the

grammar fragments. On the other hand, the probability distribution focusing on semantic associations can be obtained from the call-type frequencies. Equation (3) shows a probability distribution based on call-type frequencies. The symbol c_i denotes one of the call-types in this task and $C(\ c_i\ f_j\ )$ is the frequency of the call-type c_i associated with the phrase f_j.

$$p(c_i|f_j) = \frac{C(\ c_i\ f_j\ )}{C(\ f_j\ )} \tag{3}$$

3.3 Kullback-Leibler Distance

The Kullback-Leibler distance is one of the most popular distances for measuring similarity between two probability distributions. Because of the logarithmic term in the Kullback-Leibler distance, the probabilities in Equations (1) and (2), and (3) must be positive. Therefore, back-off smoothing is applied in advance to each probability distribution by using a unigram probability distribution. The context frequency list S described in Section 3.2 and the set of call-type frequencies are utilized to make the context and the call-type unigram probability distributions, respectively. Equation (4) shows the definition of the Kullback-Leibler distance between fragments f_1 and f_2 exploiting the succeeding context probability distributions.

$$d_s(f_1\ f_2) = \sum_{\forall s_i \in S} \hat{p}(s_i^{t+1}|f_1^t) \cdot \log \frac{\hat{p}(s_i^{t+1}|f_1^t)}{\hat{p}(s_i^{t+1}|f_2^t)} \tag{4}$$

where S is the context frequency list described in Section 3.2. The symbol s_i is one of the context stored into the context frequency list. Symbols $\hat{p}(s_i^{t+1}|f_1^t)$ and $\hat{p}(s_i^{t+1}|f_2^t)$ are the smoothed probability distributions for the fragment f_1 and f_2, respectively. A distance based on the preceding context probability distributions can also be measured in the same manner. Equation (5) defines the distance based on preceding context probability distributions.

$$d_p(f_1\ f_2) = \sum_{\forall s_i \in S} \hat{p}(s_i^{t-1}|f_1^t) \cdot \log \frac{\hat{p}(s_i^{t-1}|f_1^t)}{\hat{p}(s_i^{t-1}|f_2^t)} \tag{5}$$

where $\hat{p}(s_i^{t-1}|f_1^t)$ and $\hat{p}(s_i^{t-1}|f_2^t)$ are smoothed predecessor probability distributions for the fragments f_1 and f_2, respectively. Equation (6) defines the distance based on call-type probability distributions. In Equation (6), c_i is one of the call-types belonging to the call-type set C. $\hat{p}(c_i|f_1)$ and $\hat{p}(c_i|f_2)$ are smoothed probability distributions for the call-type c_i associated with fragments f_1 and f_2, respectively.

$$d_c(f_1\ f_2) = \sum_{\forall c_i \in C} \hat{p}(c_i|f_1) \cdot \log \frac{\hat{p}(c_i|f_1)}{\hat{p}(c_i|f_2)} \tag{6}$$

In general, the Kullback-Leibler distance is a asymmetric measure. Namely, the distance from f_1 to f_2 is not equal to that from f_2 to f_1. We therefore symmetrize the Kullback-Leibler measure by defining each type of distance as the average of two distances measured from both fragments. Thus the fragment distances shown in Equations (7), (8) and (9) are used in fragment clustering.

$$D_s(f_1\ f_2) = \frac{d_s(f_1\ f_2)\ +\ d_s(f_2\ f_1)}{2} \tag{7}$$

$$D_p(f_1\ f_2) = \frac{d_p(f_1\ f_2)\ +\ d_p(f_2\ f_1)}{2} \tag{8}$$

$$D_c(f_1\ f_2) = \frac{d_c(f_1\ f_2)\ +\ d_c(f_2\ f_1)}{2} \tag{9}$$

4 Grammar Fragment Clustering

The basic idea for grammar fragment clustering is that the fragments having a comparatively small distance from a reference fragment are regarded as being similar and are clustered into the same grammar fragment. In this study, however, three distances based on preceding contexts, on succeeding contexts and on call-types are obtained between fragments. Therefore the fragments of which all distances are small are clustered together. At first all candidate phrases described in 3.1 are generated from the training transcriptions. Then each candidate phrase forms a grammar fragment as the initial set of grammar fragments. Namely each grammar fragment consists of one candidate phrase at the first stage. The following procedure in the fragment clustering algorithm is described as follows.

The frequency of each grammar fragment is obtained by summing candidate phrase frequencies. Grammar fragment f_0 having the highest frequency and consisting of one phrase is selected as the *reference fragment*. All fragments are sorted in the order of fragment distances measured from f_0. The fragment distance lists based on preceding contexts, on succeeding contexts, and on call-types are sorted independently. Thus three fragment lists in order of distance are obtained as the result of the sorting. In each fragment list, the subset of fragment for clustering is determined based on the maximum difference in distance between successive fragments in that list. For instance, in the fragment list based on the distance on succeeding contexts, the number of candidate fragments $N_s(f_0)$ is determined by:

$$N_s(f_0) = \underset{1 \le i \le N_m}{\operatorname{argmax}} \{\ D_s(f_0\ f_{i+1}) - D_s(f_0\ f_i)\ \} \tag{10}$$

Table 1: An Example of Fragment Clustering with reference *"charge it to"*

Rank	Preceding Context		Succeeding Context		Call-type	
	Phrase	$D_p(f_0 f_i)$	*Phrase*	$D_s(f_0 f_i)$	*Phrase*	$D_c(f_0 f_i)$
1	*charge it*	0.01	*charge this to*	0.53	*charge it*	0.01
2	**bill it to**	0.47	*it charged to*	0.55	*it to my*	0.02
3	*bill it*	0.48	*this call on*	0.55	*and charge it*	0.03
4	*put it on*	0.67	*trying to use*	0.55	*and bill it*	0.04
5	*have it*	0.84	*put this on*	0.59	**bill it to**	0.08
6	*put it*	1.03	*this on*	0.63	*it to*	0.09
7	*charge*	1.35	*put it on*	0.66	*call and bill*	0.09
8	*I keep getting*	1.44	*call using*	0.67	*and have*	0.10
9	*then I*	1.64	*charge to*	0.68	*bill it*	0.10
10	*T calling card*	1.81	**bill it to**	0.71	*charged to my*	0.10
...	...	...	...	...	...	...
17	*I can't seem*	2.42	*like to use*	0.88	*to my*	0.17
18	*they said*	**2.52** N_p	*using*	0.88	**charge to**	0.18
19	**charge to**	**3.04**	*it on*	0.91	*to bill*	**0.21** N_c
20	*he*	3.07	*use*	0.94	*like to bill*	**0.27**
21	*I didn't*	3.23	*billed to*	1.10	*have it*	0.29
22	*reverse*	3.26	*it billed to*	1.16	*it charged to*	0.29
23	*I can't get*	3.29	*call with*	1.38	*billed to my*	0.30
24	*for some reason*	3.30	*bill this to*	1.91	*a call and*	0.31
25	*see if*	3.31	**charge to**	2.35	*it charged*	0.31
26	*every time*	3.36	*on*	**2.79** N_s	*billed to*	0.32
27	*I got*	3.48	*give you*	**4.02**	*phone call and*	0.34
28	*I can't*	3.55	*number to*	4.69	*to charge*	0.37
...	...	...	...	...	...	...

where, f_i and f_{i+1} are rank ordered fragments with respect to the distance on succeeding context. $D_s(f_0\ f_{i+1})$ and $D_s(f_0\ f_i)$ are the distance from reference fragment f_0 to fragment f_{i+1} and f_i, respectively. The distance $D_s(f_0\ f_i)$ monotonically increases with i. N_m is the maximum number of fragments to be compared. The number of candidate fragments based on the distance focusing on preceding contexts $N_p(f_0)$ and call-types $N_c(f_0)$ can also be determined by using distances $D_p(f_0\ f_i)$ and $D_c(f_0\ f_i)$. Following these determinations, the maximum number of candidates among three types of distance is determined by:

$$\mathcal{N}(f_0) = \max\{\ N_p(f_0),\ N_s(f_0),\ N_c(f_0)\ \} \qquad (11)$$

All fragments, whose rank order in each fragment list is less than $\mathcal{N}(f_0)$, are selected as the candidates of similar fragments. Fragments listed within the ranking $\mathcal{N}(f_0)$ among all three types of candidate list are syntactically and semantically similar to the reference fragment f_0. Such fragments are merged into the reference fragment f_0. Equation (12) shows the criterion of fragment classification based on fragment distance order.

$$\begin{aligned} f_0' = \{f_i \mid\ & O_p(f_i) \le \mathcal{N}(f_0)\ \ \& \\ & O_s(f_i) \le \mathcal{N}(f_0)\ \ \& \\ & O_c(f_i) \le \mathcal{N}(f_0)\} \end{aligned} \qquad (12)$$

Table 2: Example of Grammar Fragments

< 000 >
"hi operator" *"yes good morning"* *"yes please"* *"yes ma'am"* *"yes hi"* *"hi"* *"yeah"* *"yes"* *"yes operator"*
< 001 >
"a" *"a a"*
< 002 >
"make" *"place"*
< 003 >
"operator I'd like" *"I want"* *"I would like"* *"I like"* *"I'd lile"*
< 004 >
"make this" *"place a"* *"make a"*
< 005 >
"have" *"need"* *"want"* *"would like"*
< 012 >
"calling card" *"credit card"*
< 013 >
"collect phone call" *"collect call"* *"collect call please"*
< 015 >
"want to make" *"like to place"* *"like to make"*
< 028 >
"home phone" *"home number"* *"home phone number"*
< 038 >
"charge it to" *"bill it to"* *"charge to"*

where f_0' denotes the new grammar fragment generated by this merging. $O_p(f_i)$, $O_s(f_i)$, and $O_c(f_i)$ are the

ranked order focusing on preceding and succeeding contexts, and call-types, respectively. If there is a fragment similar to the reference fragment, reference fragment f_0 is updated by clustering similar fragments. The clustering algorithm is iterated over the updated fragment set. If the grammar fragment f_0' is not augmented, f_0 is referred to as one of the candidates when another fragment is selected as the reference in this iteration.

Table 1 shows an example of fragment clustering. In this example, the reference fragment contains *"charge it to"* only. All distances were measured from this reference fragment. Two fragments **"bill it to"** and **"charge to"** were merged into the reference fragment. In the fragment list on Table 1 focusing on the preceding context, for instance, the maximum difference in the distance determined the number of candidate fragments N_p(*"charge it to"*) = 18. The maximum number of candidates among three types of distance $\mathcal{N}$(*"charge it to"*) = 26 based on the equation (11). The reference fragment *"charge it to"* and other two fragments *"bill it to"* and *"charge to"*, which were listed within the ranking $\mathcal{N}$*("charge it to")* = 26, were merged into the reference fragment to form the fragment grammar.

Table 2 shows an example of the fragment grammar generated through the algorithm with the following parameter values. The number of words in a phrase was constrained up to three. Each phrase observed 30 times or more in the training transcription was selected as a *candidate* to participate in the clustering. The maximum number of candidate fragments was N_m = 80. The fragment clustering algorithm yielded a total of 288 phrases in 111 grammar fragments. The average number of phrases in a grammar fragment was 2.59 (= 288 / 111). A grammar fragment named < 000 > consists of 9 phrases as shown in Figure 2. This fragment contains the maximum number of phrases. The fragment demonstrates that several kind of greeting phrases, which are generally observed in spontaneous speech dialogs, were clustered by our algorithm.

Some phrases in the fragment can be partially replaced into another grammar fragment. Namely, by using the fragment having a higher fragment frequency than that of a target fragment, the fragment is parsed and some words in the fragment can be replaced by a non-terminal symbol representing another fragment. A phrase *"want to make"* in the fragment < 015 > in Figure 2, for instance, can be decomposed into *"want"*, *"to"*, and *"make"*. The words *"want"* and *"make"* are one of the phrases in the fragment < 005 > and < 002 >, respectively. Therefore, the phrase *"want to make"* can be represented as "< 005 > *to* < 002 >". As a consequence of this parsing, the fragment grammar acquires the ability to represent not only phrases given as input but also word sequences not observed in the training transcriptions. Figure 3 shows an example of the grammar fragment generalization by parsing fragments. In this example, the phrases in the fragment < 015 > are generalized by using the fragments < 002 > and < 005 >. Three phrases in the fragment < 015 > can be represented as "< 005 > *to* < 002 >" and *"like to* < 002 >". These non-terminal symbols in fragment < 015 > are expanded into phrases such as *"need to place"* and *"would like to place"*. As a consequence of this generalization, fragment < 015 > has acquired additional seven phrases such as *"want to place"*, *"would like to make"*, and *"have to place"*.

The generalization of grammar fragments is performed in order of the grammar fragment frequency. A parser replaces word sequence of each phrase into non-terminal symbol that represents the grammar fragment in which the phrase belongs. When a set of grammar fragments has been created, the frequency of each fragment is obtained by summing the frequencies of the phrases represented by that fragment. By applying the grammar fragment generalization, 495 phrases are created in 85 grammar fragments. This reveals that the average number of phrases in a fragment = 5.82 (495 / 85) was increased by this generalization.

Table 3: An example of Grammar Fragment Generalization

a. Several Grammar Fragments

< 002 >	
"make"	*"place"*
< 005 >	
"have"	*"need"*
"want"	*"would like"*
< 015 >	
"want to make"	*"like to place"*
"like to make"	

b. Grammar Fragment created from other Fragments

< 015 >	
"< 005 > *to* < 002 >"	*"like to* < 002 >"
word sequences matching this grammar fragment	
"want to place"	*"would like to place"*
"need to place"	*"have to place"*
"want to make"	*"would like to make"*
"need to make"	*"have to make"*
"like to place"	*"like to make"*

For call-type classification, *salient grammar fragments* are automatically generated from the parsed training transcriptions and associated call-types [Gorin97]. Each *salient grammar fragment* comprises a call-type of the highest association score and a corresponding sequence that comprises conventional words and non-terminal symbols for grammar fragments. Figure 2 shows examples of the *salient grammar fragment*. The fragment grammar enables the *salient grammar fragments* to represent several

kinds of word sequences having both syntactic and semantic similarity. In the call-type classification process, the ASR output is first parsed by using the fragment grammar and the *salient grammar fragments* are extracted from the parsed output. A call-type classifier determines 1st and 2nd most likely call-types for each utterance by using the association between *salient grammar fragments* and call-types. The call-type classification performance is evaluated by a scorer by using call-type assigned to each test-set utterances.

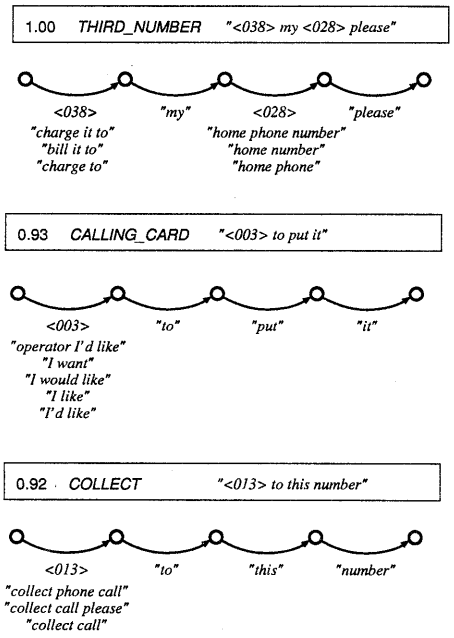


Figure 2: An Example of Salient Grammar Fragments

5 Experiment

The engine used for speech recognition is the AT&T Watson recognizer [Sharp97]. The speech recognition process is performed using the *Variable N-gram Stochastic Automaton (VNSA)* [Riccardi96] as the language model. The acoustic model for the process was trained with a database of telephone-quality spontaneous utterances. The training transcription contains 7,844 sentences while the test transcription comprises 1,000 sentences. For grammar fragment acquisition, the number of words in a phrase was constrained to be three or less in this experiment. Each phrase observed 30 times or more in the training transcription was selected as a *candidate* to participate in the clustering. In total 1,108 candidate phrases were obtained. Context length N_c for computing the distances between two fragments was set to one. In all, 3,582 context phrases are used for creating the syntactic probability distributions. The maximum number of candidate fragments, N_m was 80.

In the call-type classification, there are two important performance measures. The first measure is the *false rejection rate*, where a call is falsely rejected or classified as the call-type *other*. Since such calls are transferred to a human operator, this measure corresponds to a missed opportunity for automation. The second measure is the *probability of correct classification*. Errors in this measure lead to mis-understanding that must be resolved by a dialog manager[Boyce96] [Abella97]. Figure 3 illustrates the probability of correct classification versus the false rejection rate. As a baseline for comparison, the performance without the fragment grammar is also shown in this figure. The curves are generated by varying a salience threshold [Gorin95a].

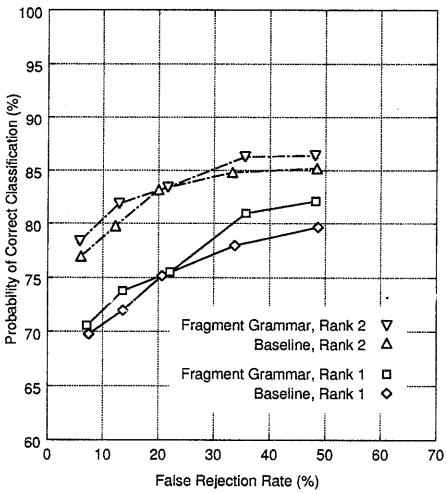


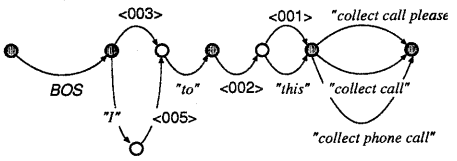
Figure 3: Call-type Classification Performance

The call-type classification performance is significantly improved by the fragment grammar. This improvement is because the *salient grammar fragments* used in the call-type classifier now accept various phrases that are syntactically and semantically similar to the originals providing generality. From this experiment result, we can conclude that by generalizing grammar fragments, unobserved phrases are obtained without deteriorating the call-type classification performance.

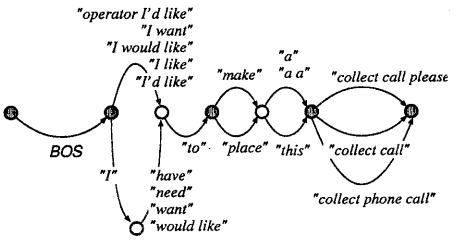
A example of variety of phrases accepted by a *salient grammar fragment* is illustrated in Figure 4. The fragment *"BOS* < 017 > < 004 > < 013 > " shown in Figure 4 has an association with the call-type *"COLLECT"* with an association score of 0.97. The fragment classes "< 017 >", "< 004 >", and "< 013 >" used in this *salient grammar fragment* can be expanded into phrases and other fragment grammars. A grammar fragment "< 017 >",

for instance is expanded into two path, "< 003 >" or "I < 005 >". In consequence of this expansion, the salient fragment network fully expanded is obtained as shown in the third of Figure 4. This phrase network accepts a total of 126 types of phrases. It is remarkably worthwhile noting that some phrases represented by this *salient grammar fragment* such as *"BOS I like to make a collect phone call"* are not observed in the training transcriptions. In all, 246 unseen salient phrases have been discovered by clustering and generalizing the fragment grammar.

[a] Salient Grammar Fragment (not Expanded)



[b] Salient Grammar Fragment (Partially Expanded)



[c] Salient Grammar Fragment (Fully Expanded)

Figure 4: Example of Phrases Accepted by a Salient Grammar Fragment

6 Conclusion

We have described a new method for automatically acquiring grammar fragments for understanding fluently spoken language. Grammar fragments representing a set of syntactically and semantically similar phrases were generated by using three probability distributions: of succeeding words, of preceding words, and of associated call-types. The similarity between phrases was measured by applying the Kullback-Leibler distance to these three probability distributions. Phrases being close in all three distances were clustered into a grammar fragment. By parsing phrases in grammar fragments, the fragment grammar detected 246 phrases in the test-set that were not present in the training set. This result revealed that unseen phrases have been automatically discovered by our new method. The experimental results show that a 3% improvement in the call-type classification performance was achieved by introducing the grammar fragments.

Acknowledgments The authors wish to thank Larry Rabiner and Jay Wilpon of AT&T Laboratories-Research, Nobuhiko Kitawaki and Shigeki Sagayama of NTT Human Interface Laboratories, and Sadaoki Furui of Tokyo Institute of Technology for their support and encouragement of this research.

References

[Abella97] A. Abella and A. L. Gorin, *"Generating Semantically Consistency Inputs to a Dialog Manager,"* Proc. of Eurospeech, pp.1879-1882, *(1997)*.

[Bellegarda96] J. R. Bellegarda, J. W. Butzberger, Y. L. Chow, N. B. Coccaro and D. Naik, *"A Novel Word Clustering Algorithm Based on Latent Semantic Analysis,"* Proc. of ICASSP, pp.172-175, *(1996)*.

[Boyce96] S. Boyce and A. L. Gorin, *"User Interface Issues for Natural Spoken Dialog Systems,"* Proc. of ICSLP, pp. 1577-1580, *(1996)*.

[Farhat96] A Farhat, J. F. Isabella and D O'Shaughnessy, *"Clustering Words for Statistical Language Models Based on Contextual Word Similarity,"* Proc. of ICASSP, pp.180-183, *(1996)*.

[Giachin95] E. Giachin, *"Phrase Bigrams for Continuous Speech Recognition,"* Proc. of ICASSP, pp.225-228, *(1995)*.

[Gorin97] A. L. Gorin, G. Riccardi and J. H. Wright, *"How May I Help You?,"* (to appear in *Speech Communication*).

[Gorin95a] A. L. Gorin, *"Processing of Semantic Information in Fluently Spoken Language,"* Proc. of ICSLP, pp.1001-1004, *(1996)*.

[Gorin95b] A. L. Gorin, *"On Automated Language Acquisition,"* Journal of the Acoustic Society of America, 97(6), pp.3441-3461, *(1995)*.

[Masataki96] H. Masataki and Y. Sagisaka, *"Variable-order N-gram Generation by Word-Class Splitting and Consecutive Word Grouping,"* Proc. of ICASSP, pp.188-191, *(1996)*.

[Ney93] H. Ney and U. Esseh, *"Estimating Small Probabilities by Leaving One Out,"* Proc. of EuroSpeech, pp.2239-2242, *(1993)*.

[Riccardi96] G. Riccardi, R. Pieraccini and E. Bocchieri, *"Stochastic Automata for Language Modeling,"* Computer Speech and Language, 10 (4), pp.265-293, *(1996)*.

[Riccardi97] G. Riccardi, A. L. Gorin, A. Ljolje and M. D. Riley, *"A Spoken Language System for Automated Call Routing,"* Proc. of ICASSP, pp.1143-1140, *(1997)*.

[Sharp97] R. D. Sharp et al., *"The WATSON Speech Recognition Engine,"* Proc. of ICASSP, pp.4065-4068, *(1997)*.

[Ward96] W. Ward and S Issar, *"A Class Based Language Model for Speech Recognition,"* Proc. of ICASSP, pp.416-418, *(1996)*.

[Wright97] J. Wright, A. L. Gorin and G. Riccardi, *"Automatic Acquisition of Salient Grammar Fragments for Call-type Classification,"* Proc. of Eurospeech, pp.1419-1422, *1997*.