ELSEVIER

Speech Communication 27 (1999) 43–62

SPEECH COMMUNICATION

Grammar Fragment acquisition using syntactic and semantic clustering

Kazuhiro Arai [a,*], Jeremy H. Wright [b,1], Giuseppe Riccardi [b,2], Allen L. Gorin [b,3]

[a] *NTT Human Interface Laboratories, 1-1 Hikari-no-oka, Yokosuka, Kanagawa 239-0847, Japan*
[b] *AT&T Laboratories-Research, 180 Park Ave., Florham Park, NJ 07932, USA*

Received 14 October 1997; received in revised form 16 July 1998; accepted 2 September 1998

Abstract

A new method for automatically acquiring Fragments for understanding fluent speech is proposed. The goal of this method is to generate a collection of 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 following contexts, of preceding contexts, and of associated semantic actions. The similarity between candidate phrases is measured by applying the Kullback–Leibler distance to these three probability distributions. Candidate phrases that are close in all three distances are clustered into a Fragment. Salient sequences of these Fragments are then automatically acquired, and exploited by a spoken language understanding module to classify calls in AT&T's "How may I help you?" task. These Fragments allow us to generalize unobserved phrases. For instance, they detected 246 phrases in the test-set that were not present in the training-set. This result shows that unseen phrases can be automatically discovered by our new method. Experimental results show that 2.8% of the improvement in call-type classification performance was achieved by introducing these Fragments.

Zusammenfassung

Es wird ein neues Verfahren zur automatischen Erfassung von Fragmenten für die intelligente Verarbeitung gesprochener Sprache vorgeschlagen. Ziel dieses Verfahrens ist es, Sammlungen von Fragmenten zu erstellen, die aus syntaktisch und semantisch ähnlichen Sätzen bestehen. Zunächst werden alle Sätze, die während der Übungsphase häufig aufgetreten sind, als "Kandidaten" ausgewählt. Jedem dieser Sätze sind drei Wahrscheinlichkeitsverteilungen zugeordnet: der des nachfolgenden Kontextes, des vorhergehenden Kontextes und der zugehörigen semantischen Aktionen. Die Ähnlichkeiten zwischen den ausgewählten Sätzen wird durch Anwendung der Kullback–Leibler-Entfernung auf diese drei Wahrscheinlichkeitsverteilungen gemessen. Alle Redewendungen, die bei allen drei Verteilungen ähnliche Entfernungen aufweisen, werden zu einem Fragment zusammengefaßt. Anschließend werden alle auffälligen Sequenzen innerhalb dieser Fragmente automatisch erfaßt und von einem gesprochene Sprache verstehenden Modul ausgewertet, damit die Anrufe in der AT&T-Aufgabe "Wie kann ich Ihnen weiterhelfen?" klassifiziert werden können. Mit Hilfe dieser Fragmente lassen sich bisher unbeachtete Sätze verallgemeinern. In einer Testanordnung wurden

* Corresponding author. E-mail: arai@nttspch.hil.ntt.co.jp
[1] E-mail: jwright@research.att.com.
[2] E-mail: dsp3@research.att.com.
[3] E-mail: algor@research.att.com.

0167-6393/99/$ – see front matter
PII: S0167-6393(98)00065-X

beispielsweise 246 Sätze registriert, die in der Übung nicht enthalten waren. Dieses Ergebnis zeigt, daß unbemerkte Sätze mit unserem neuen Verfahren automatisch entdeckt werden. Die Versuche ergaben, daß durch Einführung dieser Fragmente die Klassifizierung der Anruftypen um 2,8 Prozent verbessert werden konnte.

Résumé

Une nouvelle méthode d'acquisition automatique des Fragments pour la compréhension du langage courant est désormais proposée. L'objectif de cette méthode est de générer une collection de Fragments, chacun représentant un ensemble de phrases similaires d'un point de vue syntaxique et sémantique. En premier lieu, les phrases fréquemment rencontrées dans le kit de formation sont sélectionnées en qualité de phrases candidates. Chaque phrase candidate présente trois répartitions de probabilité associées: contextes suivants, contextes précédents et actions sémantiques associées. La similitude entre les phrases candidates est mesurée en appliquant la distance Kullback–Leibler à ces trois répartitions de probabilité. Les phrases candidates proches des trois distances sont regroupées au sein d'un Fragment. Les séquences représentatives de ces Fragments sont ensuite acquises automatiquement, et exploitées par un module de compréhension du langage courant afin de classer les apples dans la tâche AT&T dénommée "How May I Help You?" ("*Comment puis-je vous aider*?"). Ces fragments nous permettent de généraliser les phrases non observées. Par exemple, ils ont permis de détecter 246 phrases présentes dans les kits de test et absentes des kits de formation. Ce résultat montre que les phrases qui n'ont pas été vues peuvent être découvertes automatiquement grâce à notre nouvelle méthode. Des résultats expérimentaux montrent une amélioration de 2,8% des performances de classification des types d'appels après la mise en place de ces Fragments.

Keywords: Spoken language understanding; Phrase similarity; The Kullback–Leibler distance; Grammar fragments; Call-type classification

1. Introduction

Stochastic language models generated from huge text corpora have been successfully applied to several speech recognition-based tasks. Although the stochastic language model achieves remarkable performance in large vocabulary continuous speech recognition, it is difficult for conventional stochastic language models to accurately recognize spontaneous speech spoken by non-expert users.

We have developed a spoken dialog system for call routing. In the task of call routing, services that the user can access are categorized into 14 types and *other* as a complement (Gorin et al., 1997). Each of the 14 services is assigned to a different *call-type*. The system determines which call-type is required by accepting spontaneous speech as input at the first stage of the procedure. Once the call-type has been determined, the information needed for completing each service is requested using another dialog. This paper describes a new stochastic language model for spontaneous speech understanding; it mainly focuses on call-type classification from spontaneous utterances. In our task, there is no need for the dialog system to achieve high performance in *word-by-word* decoding. The system should rather extract continuous word sequences strongly linked to the call-types.

Early versions of this method were applied to several different tasks of various complexity: call routing, data retrieval, robotics blocks world (Gorin, 1995). In general, contents of spontaneous speech can be classified into one of several categories such as the call-types, if the word sequences in the spontaneous speech can be linked to one or more categories by using a stochastic process. Therefore this approach is widely applicable to not only the application of call routing but also other applications if a set of training transcriptions is provided together with the classification category of each transcription.

The proposed dialog system for call routing and the new stochastic language model used for understanding spontaneous speech are described below. Fig. 1 outlines the dialog system for call routing. The goal of this system is to understand an input well enough to identify the service type

desired by the caller in a telecommunications environment. Since in many situations the call-type cannot be unambiguously determined from a single input, dialog is often necessary. This can be due to an ambiguous request or to the weakness of the spoken language understanding algorithm. This system accepts spontaneous speech spoken by non-expert users as input in a telecommunications environment. As shown in Fig. 1, the *Speech Recognizer* recognizes the speech and generates a word sequence by using a conventional stochastic language model. The *Call-type Classifier* accepts the word sequence as input and determines the call-type corresponding to the utterance by using the interpretation knowledge, of what we call "*Salient Fragments*", that defines associations between a word sequence and call-types. It has been shown that the dialog system can classify the call-type adequately if the system can extract the phrases strongly associated with particular call-types. For instance, the word sequence "*my credit card*" is strongly associated with the call-type *calling_card*. The algorithm for call-type classification is designed to exploit the salient association between the phrases in the user's utterance and call-types (Gorin et al., 1997).

Once the call-type has been agreed upon, a dialog is conducted to collect all information necessary to provide the service. For instance, the service called *calling_card* has the following procedures. First, a credit card number is entered into the system. A telephone number that the user wants to call is then entered, if the card number is accepted. Speech recognition or some other technique can be applied to perform these procedures. The call is finally connected to the user to complete this service.

This paper mainly focuses on a new stochastic language model used in the speech understanding process for determining the call-type. The Call-type Classifier and the Salient Fragments in Fig. 1 perform this process. In conventional stochastic language modeling, the probability of a word occurrence is generally calculated from its frequency as obtained from training transcriptions. This approach is applicable to not only words but also phrases. In fact, frequencies of some phrases are higher than those of some words that are rarely observed in the training set. It is therefore reasonable that a phrase is regarded as a unit for language modeling (Giachin, 1995; Masataki and Sagisaka, 1996; Riccardi et al., 1997). The phrase-

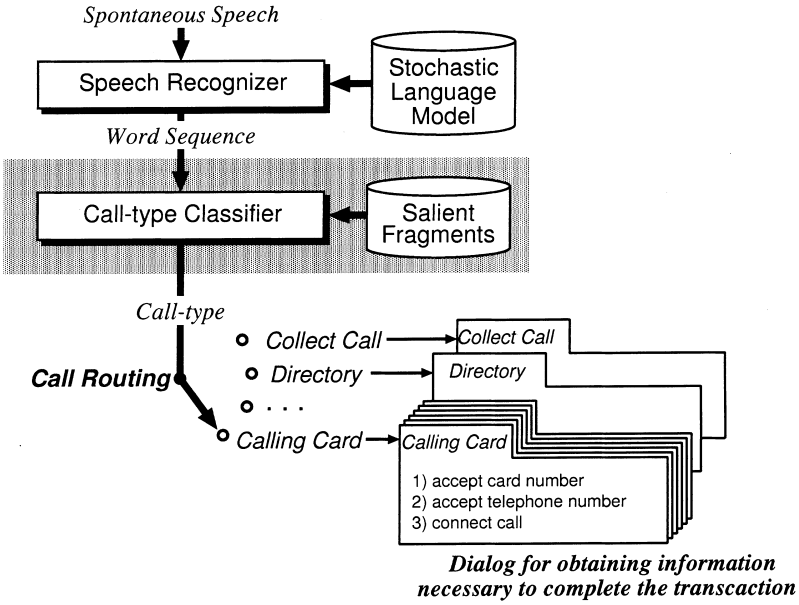

Fig. 1. Dialog system with call routing.

based N-gram makes it possible to estimate accurately the transition probabilities between the phrases observed frequently in the training set. Other language models have been proposed based on the word-class N-gram (Brown et al., 1992; Bellegarda et al., 1996; Farhat et al., 1996; Ward and Issar, 1996). In that approach, words having a similar pattern of transition probabilities are generally clustered into the same word class. Since 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.

Conventional clustering approaches, however, focus on following 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 following word sequence but also in the preceding word sequence, the similarity of word sequences can be clustered more effectively by referring to both following and preceding word sequences. Furthermore, an utterance accepted as input of the call-router dialog system can usually be classified semantically into one of 14 call-types. Some phrases performing an analogous role in this task must have similar association with the call-type. Thus word sequence similarity can also be computed by using such associations between phrases and call-types (Wright et al., 1997). In this paper, a new method for gathering phrases into clusters, what we call *Fragments*, is proposed. This method focuses not only on following words but also on preceding words and on the call-types associated with each utterance in order to generate the Fragments that consist of similar phrases. Distances between phrases are calculated based on the distribution of preceding and following words and the call-types.

This paper proceeds as follows. The collection and specification of training and test transcriptions are described in Section 2. In Section 3, three types of distance between phrase pairs are discussed. For each phrase, the probability distributions for following and preceding contexts and for call-types are obtained via back-off smoothing of relative frequencies in the training transcriptions. By using these distributions, three distances between two phrases are calculated by using the Kullback–Leibler distance measure. Section 4 addresses the clustering of similar phrases based on these three types of distances. Phrases clustered into the same Fragment form a part of the grammar used in the spoken understanding module. In Section 5, experimental conditions and results are described along with some of the examples generated in experiments.

2. Training and test transcriptions

A database of 10 K spoken transcriptions between users and human agents was generated as detailed in (Gorin et al., 1997). First, both channels of the dialog were recorded from the agents' headset jacks onto a digital audio tape (DAT). These recordings were then automatically segmented, filtered, and down-sampled to generate a stereo speech file for each transaction.

In this study, we focus on the first user utterance following the greeting prompt of "*How may I help you*?". These utterances were end-pointed, transcribed, and labeled as to the call-type and quality of the speech and channel. In call-type labeling, one of 15 call-types was assigned to each transcription. In some cases, two or more call-types were labeled to a transcription. For instance, the two call-types *dial_for_me* and *calling_card* were assigned to the transcription, "*Yeah can you dial a number for me I wanna put this on my calling card it's two one...*". The transcriptions were split into three subsets for training (8 K), developing (1 K), and testing (1 K) the acoustic and language models for recognition and understanding. The training set had approximately 3.6 K words to define the vocabulary.

3. Fragment distance

3.1. Phrase and Fragment

We now define some of the terms used in this paper to clarify their use in our study.

Phrase: An arbitrary continuous 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 three or less in this experiment.

Fragment: The phrases having higher frequency than some threshold are selected as *candidates*. The candidate phrases are regarded as units for generating the *Fragments*. Each Fragment is acquired via the 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 Fragments. The Fragment grammar also 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, that these techniques can generally be applied in a straightforward manner to Fragments or sequences thereof.

3.2. Generation of candidate phrases

Table 1 shows the numbers of phrases for lengths up to three as observed in the training transcriptions. In Fig. 2, the rank frequency distribution of each phrase consisting of one, two or three words is illustrated. Fig. 2 demonstrates that the most frequent phrase consisting of two words is, as an example, observed approximately 2000 times in the training transcriptions. On the other hand, the frequency of an individual word whose ranking is lower than 100 is fewer than that of some frequent phrases consisting of two or three words.

Some phrases and their frequencies are shown in Fig. 3. The frequency of each phrase was counted independently of other phrases. For instance, Fig. 3 shows that the phrase "*call*" was observed more than 4000 times and this phrase was preceded by the word "*collect*" in "*collect call*" more than 1000 times in the training transcriptions. Fig. 3 also illustrates that the frequency of some phrases, such as "*like to*" and "*I'd like*", is higher than that of some individual words. These frequencies reveal that some phrases that appear in

Table 1
Number of phrases

Number of words	Number of phrases
1	3566
2	22,233
3	48,457

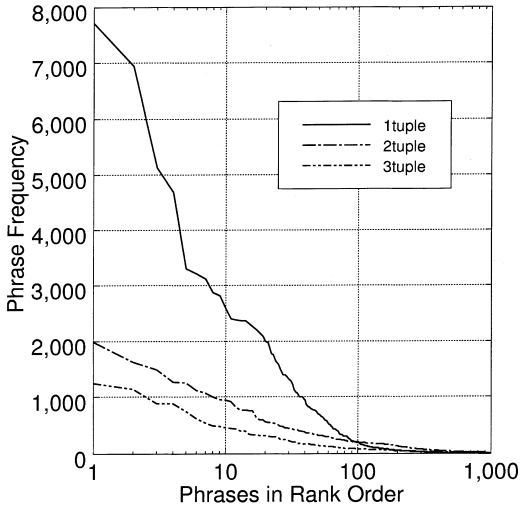

Fig. 2. Phrase rank–frequency distributions.

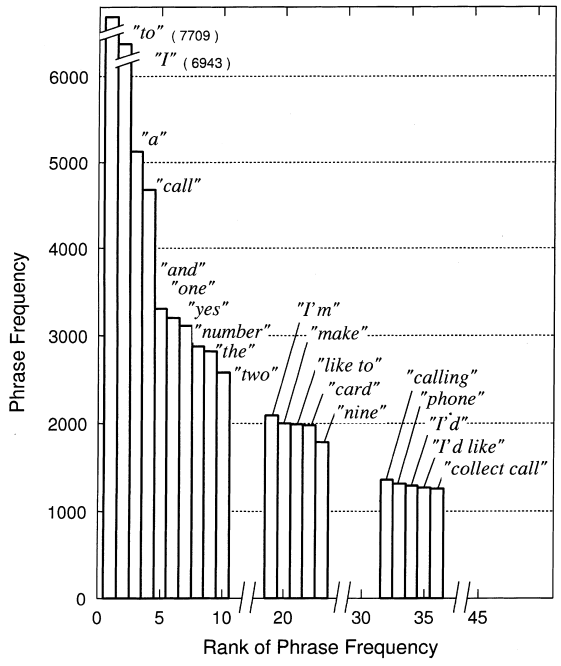

Fig. 3. Phrase frequency.

the training transcriptions can be defined as units for language modeling. Transition probabilities for phrases can be calculated in the same way as for single words, by using the conventional approach of N-gram language modeling.

3.3. Syntactic and semantic associations of a Fragment

There are many definitions of the linguistic terms *syntax* and *semantics*, so we now define some of the terms used in this paper for clarification. In this discussion, *syntactic association* signifies the relationship between a Fragment and the phrases following or preceding the Fragment. Several kinds of following and preceding phrases for each Fragment are generally observed in training transcriptions. If the roles of Fragments are similar to each other in spoken dialog, then the distribution of these phrases will be similar for the Fragments. Thus by syntactic association, we do not explicitly focus on grammatical issues such as *part-of-speech* and *tense* but rather on the distribution of phrases surrounding a Fragment. On the other hand, *semantic association* signifies the relationship 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-type.

An example of the syntactic and semantic associations seen for a Fragment is illustrated in Fig. 4. f denotes a Fragment, s and c define a preceding or following phrase and call-type, respectively. In Fig. 4, f consists of only one phrase "*calling card*". Suffixes such as t and $t+1$ denote sequence order. 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 combination Fragment f followed by phrase s. *BOS* and *EOS* denote *Beginning-Of-Sentence* and *End-Of-Sentence*, respectively. Fig. 4 shows that the phrase "*calling card*" was observed 962 times in the training transcriptions. The phrase "*on my*", for

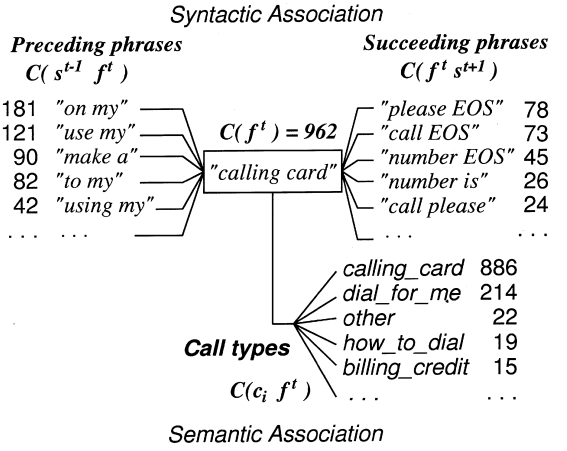

Fig. 4. Syntactic and semantic associations of a Fragment.

instance, preceded that Fragment 181 times and "*number is*" followed it 26 times. In the syntactic association, the sum of the preceding phrase frequencies is equivalent to the frequency of the Fragment. The sum of the following phrase frequencies is also equivalent to $C(f^t)$. Eq. (1) shows the relationship among $C(f^t)$, $C(s^{t-1}f^t)$ and $C(f^t s^{t+1})$.

$$C(f^t) = \sum_{\forall s} C(s^{t-1} f^t) = \sum_{\forall s} C(f^t s^{t+1}). \tag{1}$$

We note that the sum of the call-type frequencies is not equivalent to the Fragment frequency in some cases. The reason for this inequality is as follows. Several call-types may be assigned to a training transcription as described in Section 2. When a training transcription has two or more call-types and a phrase is extracted from such transcription, the frequencies of all call-types associated with the phrase are incremented. For instance, the phrase "*calling card*" is observed in the transcription "*Yeah can you dial a number for me I wanna put this on my calling card it's two one...*", and this transcription has two call-types, *dial_for_me* and *calling_card*. In this case, the frequencies for these two call-types are incremented. By counting the preceding and following phrases, two syntactic probability distributions for each Fragment are obtained.

The call-type probability distribution for each Fragment is also obtained by using call-type

frequencies. This call-type probability distribution represents the semantic feature vector for a Fragment. In order to generate syntactic probability distributions, a set of phrases that precede or follow Fragments is generated first. In the following discussion, a phrase that follows or precedes a Fragment is called its *context*. Although in our experiments the context consists of single words, our algorithm can be applied to longer contexts so we describe the method in its general form. Consequently, the context can contain words and non-terminal symbols corresponding to Fragments. The number of words and non-terminal symbols in each context is equalized to compute the 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.

The most frequent contexts are shown in Table 2. Since the contexts are the predecessor or successor of a Fragment, they consist of not only words but also the symbols BOS and EOS. In other words, a phrase in a Fragment cannot contain these symbols because it must have both preceding and following contexts. Three probability distributions for each Fragment are obtained by using preceding and following context frequencies, and call-type frequency. The bigram probability distributions focusing on following and preceding contexts are defined in Eqs. (2) and (3), respectively.

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

$$p(s_i^{t-1}|f_j^t) = \frac{C(s_i^{t-1} f_j^t)}{C(f_j)} = \frac{C(w_1^{t-N_c} \cdots w_{N_c-1}^{t-2} w_{N_c}^{t-1} f_j^t)}{C(f_i)}. \tag{3}$$

In both Eqs. (2) and (3), s_i denotes the ith context stored in the context frequency list S, f_j is the jth Fragment in the Fragment grammar, w_k denotes the kth word in the context s_i, and N_c ($N_c \geqslant 1$) is the number of items referred as the context. Suffixes such as t, $t+1$ and $t-1$ denote order in the word, context, or Fragment sequence. The function $C(\,)$ counts the frequency of a sequence in the training transcriptions as described in Section 3.3.

The contexts 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 variety in context can theoretically be observed. In practice, however, these probability distributions become sparse when parameter N_c is large. Therefore, the parameter N_c should vary as a function of the size of the training corpus. These two probability distributions represent syntactic feature vectors of the Fragments. On the other hand,

Table 2
Most frequent contexts (context length = 1, 2, 3)

Rank	1		2		3	
1	*BOS*	7844	*BOS yes*	3065	*to make a*	1253
2	*EOS*	7844	*like to*	1990	*I'd like to*	1146
3	*to*	7709	*make a*	1629	*BOS yes I*	991
4	*I*	6943	*to make*	1496	*a collect call*	894
5	*a*	5124	*I'd like*	1266	*like to make*	891
6	*call*	4685	*collect call*	1265	*I'm trying to*	748
7	*and*	3311	*call to*	1117	*make a collect*	621
8	*one*	3211	*trying to*	1073	*my calling card*	540
9	*yes*	3117	*yes I*	1008	*to place a*	493
10	*number*	2877	*BOS yeah*	981	*I need to*	480

the probability distribution focusing on semantic associations can be obtained from the call-type frequencies. Eq. (4) shows the probability distribution based on call-type frequencies. c_i denotes one of the call-types in this task and $C(c_i f_j)$ is the frequency of call-type c_i associated with the phrase f_j.

$$p(c_i|f_j) = \frac{C(c_i f_j)}{C(f_j)}. \tag{4}$$

As a result, three types of probability distribution are obtained for each Fragment. The distance between two Fragments is calculated by comparing each type of probability distribution. Namely, three distances between two Fragments are measured by using following and preceding context probability distributions and call-type probability distribution.

3.4. Kullback–Leibler distance

The Kullback–Leibler distance is one of the most popular distance measures for calculating the similarity between two probability distributions. Because of the logarithmic term in the Kullback–Leibler distance, the probabilities in Eqs. (2)–(4) must be positive. Therefore, back-off smoothing is applied in advance to each probability distribution by using the unigram probability distribution. The context frequency list S described in Section 3.3 and the set of call-type frequencies are utilized to make the context and the call-type unigram probability distributions, respectively. The back-off smoothing method used in this approach is described in detail in Appendix A. Eq. (5) shows the definition of the Kullback–Leibler distance between Fragments f_1 and f_2; it exploits the following context probability distributions.

$$d_{\mathrm{f}}(f_1 f_2) = \sum_{\forall s_i \in S} \hat{p}(s_i^{t+1}|f_1^t) \log \frac{\hat{p}(s_i^{t+1}|f_1^t)}{\hat{p}(s_i^{t+1}|f_2^t)}. \tag{5}$$

S is the context frequency list described in Section 3.3 and s_i is one of the contexts stored in the context frequency list. $\hat{p}(s_i^{t+1}|f_1^t)$ and $\hat{p}(s_i^{t+1}|f_2^t)$ are the smoothed probability distributions for Fragments f_1 and f_2, respectively. The distance based on the preceding context probability distributions can also be measured in the same manner. Eq. (6) defines the distance based on preceding context probability distributions.

$$d_{\mathrm{p}}(f_1 f_2) = \sum_{\forall s_i \in S} \hat{p}(s_i^{t-1}|f_1^t) \log \frac{\hat{p}(s_i^{t-1}|f_1^t)}{\hat{p}(s_i^{t-1}|f_2^t)}, \tag{6}$$

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 Fragments f_1 and f_2, respectively. Eq. (7) defines the distance based on call-type probability distributions. In Eq. (7), c_i is one of the call-types belonging to call-type set C. $\hat{p}(c_i|f_1)$ and $\hat{p}(c_i|f_2)$ are smoothed probability distributions for call-type c_i associated with Fragments f_1 and f_2, respectively.

$$d_{\mathrm{c}}(f_1 f_2) = \sum_{\forall c_i \in C} \hat{p}(c_i|f_1) \log \frac{\hat{p}(c_i|f_1)}{\hat{p}(c_i|f_2)}. \tag{7}$$

In general, the Kullback–Leibler distance is an 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 the two distances measured from both Fragments. Thus the Fragment distances shown in Eqs. (8)–(10) are used in the Fragment clustering.

$$D_{\mathrm{f}}(f_1 f_2) = \frac{d_{\mathrm{f}}(f_1 f_2) + d_{\mathrm{f}}(f_2 f_1)}{2}, \tag{8}$$

$$D_{\mathrm{p}}(f_1 f_2) = \frac{d_{\mathrm{p}}(f_1 f_2) + d_{\mathrm{p}}(f_2 f_1)}{2}, \tag{9}$$

$$D_{\mathrm{c}}(f_1 f_2) = \frac{d_{\mathrm{c}}(f_1 f_2) + d_{\mathrm{c}}(f_2 f_1)}{2}. \tag{10}$$

It is useful to plot histograms of these three types of Fragment distance. In Fig. 5, two reference Fragments f_{a} and f_{b} are selected to illustrate the difference in the histograms. In this example, each Fragment contains only one phrase. One of the phrases, f_{a}, is "*charge it to*" and the other phrase, f_{b}, is "*area code*". Three distances between one of the reference Fragments and an arbitrary Fragment f_i were measured to create the histograms. For instance, the histogram focusing on the preceding context, shown in Fig. 5(a), plots the distributions of $D_{\mathrm{p}}(f_{\mathrm{a}}\, f_i)$ and $D_{\mathrm{p}}(f_{\mathrm{b}}\, f_i)$. The histograms of $D_{\mathrm{f}}(f_{\mathrm{a}}\, f_i)$ and $D_{\mathrm{f}}(f_{\mathrm{b}}\, f_i)$, and those of

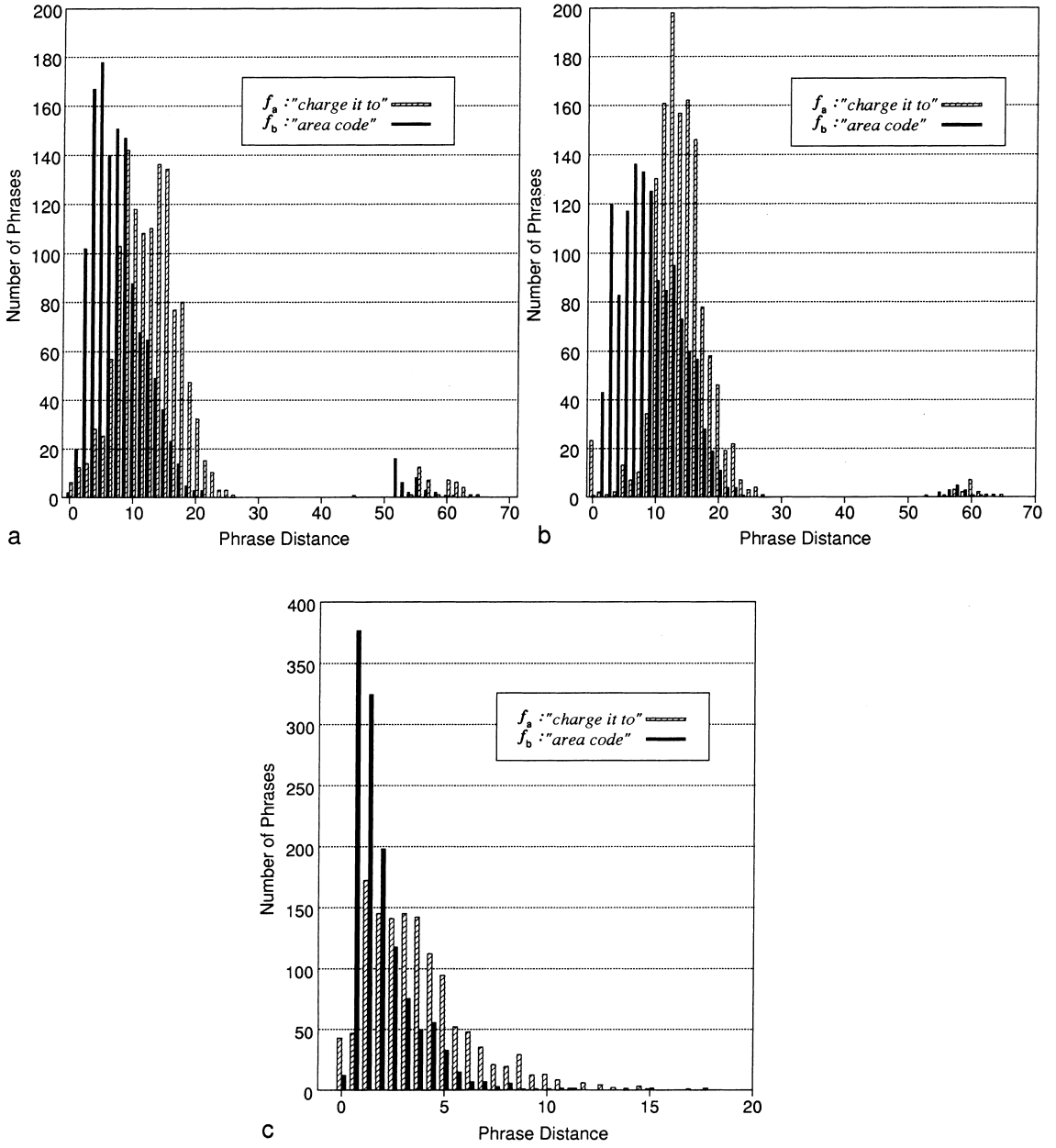

Fig. 5. Fragment distance histograms. (a) Preceding context. (b) Succeeding context. (c) Call-type.

$D_c(f_a\ f_i)$ and $D_c(f_b\ f_i)$ are illustrated in Fig. 5(b) and (c), respectively. The two histograms shown in Fig. 5(a) and (b) have similar distributions and show a large peak approximately within the range from 5.0 to 15.0. The peak for the phrase "*area code*" shifts leftward compared to that for the phrase "*charge it to*" in both histograms. In the histogram based on call-type distributions, the histogram for the phrase "*area code*" is skewed compared to that for the phrase "*charge it to*".

4. Fragment clustering

4.1. Fragment clustering based on Fragment distances

The basic idea for Fragment clustering is that Fragments having a comparatively small distance from a reference Fragment are regarded as being similar and are clustered into the same Fragment. In this study, however, three distances based on preceding contexts, on following contexts and on call-types are obtained between Fragments. Therefore, the Fragments for which all distances are small are clustered together. At first, all candidate phrases described in Section 3.1 are generated from the training transcriptions. Each candidate phrase forms a Fragment as the initial set of Fragments. Namely, each Fragment initially consists of one candidate phrase. We will now describe the remaining steps of the Fragment clustering algorithm. The frequency of each Fragment is obtained by summing candidate phrase frequencies. The Fragment with the highest frequency and consisting of one phrase f_0 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 following contexts, and on call-types are sorted independently which yields three sorted Fragment lists. In each Fragment list, the subset of Fragments for clustering are determined based on the maximum difference in distance between following Fragments in that list. For instance, in the Fragment list based on the distance on following contexts, the number of candidate Fragments $N_f(f_0)$ is determined by

$$N_f(f_0) = \underset{1 \leqslant i \leqslant N_m}{\arg\max}\{D_f(f_0 f_{i+1}) - D_f(f_0 f_i)\}, \qquad (11)$$

where f_i and f_{i+1} are rank ordered Fragments with respect to the distance on the following context. $D_f(f_0\, f_{i+1})$ and $D_f(f_0\, f_i)$ are the distances from the references Fragment f_0 to Fragments f_{i+1} and f_i, respectively. The distance $D_f(f_0\, f_i)$ 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 be also 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 using the three types of distance is determined by

$$\mathcal{N}(f_0) = \max\{N_p(f_0), N_f(f_0), N_c(f_0)\}. \qquad (12)$$

All Fragments, whose rank order in each Fragment list is less than $\mathcal{N}(f_0)$, are selected as the candidate 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. Eq. (13) shows the criterion of Fragment classification based on Fragment distance orders.

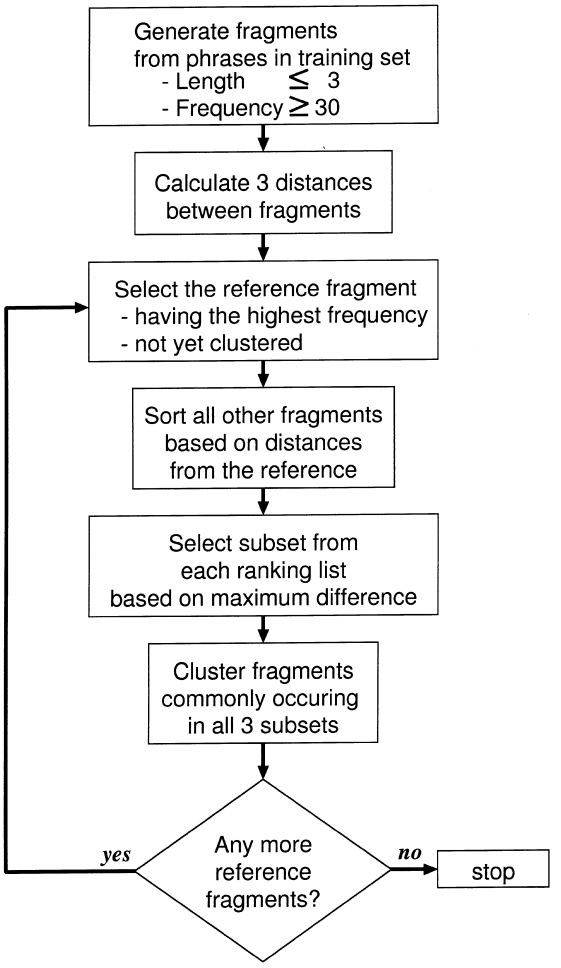

Fig. 6. Flow chart of Fragment clustering.

$$f_0' = \{f_i | O_P(f_i) \leqslant \mathcal{N}(f_0) \cap O_f(f_i) \leqslant \mathcal{N}(f_0) \cap O_c(f_i) \leqslant \mathcal{N}(f_0)\}, \quad (13)$$

where f_0' denotes the new Fragment generated by this merger. $O_P(f_i)$, $O_f(f_i)$ and $O_c(f_i)$ are the ranked orders focusing on preceding and following contexts, and call-types, respectively. If there is a Fragment similar to the reference Fragment, reference Fragment f_0 is updated by clustering the similar Fragments. The clustering algorithm is iterated over the updated Fragment set. When no Fragment is merged into the reference Fragment f_0 by Fragment clustering, Fragment f_0 is regarded as one of the Fragments in the next iteration of Fragment clustering in which other Fragments are considered for selecting the next reference.

The Fragment clustering algorithm used in the performance evaluation is illustrated in Fig. 6. Table 3 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 in Table 3 based on the preceding context, for instance, the maximum difference in the distance, N_p ("*charge it to*"), yielded 18 candidate Fragments. The maximum number of candidates among the three types of distance $\mathcal{N}$ ("*charge it to*") was 26 based on Eq. (12). 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.

Fig. 7 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 to be three or less. Each phrase observed 30 times or more in the training transcription was selected as a *candidate* to participate in clustering. The maximum number of candidate Fragments $N_m = 80$. The Fragment

Table 3
An example of Fragment clustering with reference Fragment f_0: "*charge it to*"

Rank	Preceding context		Following context		Call-type	
	Phrase: f_i	$D_p(f_0\ f_i)$	*Phrase:* f_i	$D_f(f_0\ f_i)$	*Phrase:* f_i	$D_c(f_0\ f_i)$
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
$N(f_0)$ 26	*every time*	3.36	*on*	**2.79** N_f	*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
...	...	...	...	...	...	...

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

Fig. 7. Example of Fragments.

clustering algorithm yielded, in total, 288 phrases in 111 Fragments. The average number of phrases per Fragment was 2.59 (= 288/111). The Fragment named ⟨000⟩ consists of 9 phrases as shown in Fig. 7. This Fragment contains the maximum number of phrases. The Fragment demonstrates that several kinds of greeting phrases frequently observed at the beginning of spontaneous speech dialogs, were clustered by our algorithm.

4.2. Alternative methods for Fragment clustering

This section addresses alternative methods for Fragment clustering and compares them to determine the most appropriate method for this type of clustering. Although the method so selected is described above, it is interesting to discuss the alternatives for Fragment clustering. Section 4.2.1 describes another method to determine the Fragment subsets. Although the three types of distance are processed independently to determine the Fragment subsets, these distances can be merged into one distance. The clustering method using this single distance will be discussed in Section 4.2.2. An alternative Fragment clustering algorithm that does not use one threshold but individual thresholds for the three types of distance will be described in Section 4.2.3.

4.2.1. Determination of Fragment subset

It may be possible to determine Fragment subsets without considering the distance from the

reference Fragment. For instance, the nearest N Fragments from the reference Fragment in each list can be selected as the candidate Fragments. The Fragment clustering algorithm can be processed by using such candidate Fragments. Several kinds of new Fragments can be obtained by varying N. However, the threshold N is not based on the syntactic or semantic procedures so it is very difficult to determine N for the different kinds of reference Fragment. Therefore, the number of candidate Fragments in each Fragment list should be determined based on the syntactic or semantic similarity among the Fragments.

4.2.2. Mingled Fragment distance

The clustering method described above is based on the approach of not mingling each ranking order obtained from the Fragment distances until the final stage of Fragment clustering. However, several other methods that mingle the three types of Fragment distance during Fragment clustering can be also applied. In these methods, first the new distance $\mathscr{D}(f_0 f_i)$ between the reference Fragment f_0 and a Fragment f_i is calculated by mingling the three types of distance described in Section 3.4. The Fragments within a distance threshold are merged into the reference Fragment. Eqs. (14) and (15) are examples of possible distance functions $\mathscr{D}(f_0 f_i)$.

$$\mathscr{D}_1(f_0 f_i) = \max\{D_{\mathrm{f}}(f_0 f_i), D_{\mathrm{p}}(f_0 f_i), D_{\mathrm{c}}(f_0 f_i)\}, \quad (14)$$

$$\mathscr{D}_2(f_0 f_i) = D_{\mathrm{f}}(f_0 f_i) + D_{\mathrm{p}}(f_0 f_i) + \alpha D_{\mathrm{c}}(f_0 f_i), \quad (15)$$

where $D_{\mathrm{f}}(f_0 f_i)$, $D_{\mathrm{p}}(f_0 f_i)$ and $D_{\mathrm{c}}(f_0 f_i)$ are the distances focusing on following and preceding context probability distributions, and call-type probability distribution, respectively. α is the coefficient for weighing the semantic distance. As described in Section 3.4, the syntactic distances are calculated by using the context frequency list, and the semantic distance is estimated from the distribution of the call-type frequency. Therefore, it is, in general, difficult to compare syntactic and semantic distances directly. In order to align the range of these two distances, the coefficient α is applied in Eq. (15). The advantage of these alternative distances is that between two syntactically and semantically similar Fragments the distances are extremely short, while dissimilar Fragments have long distances. On the other hand, it is clear that these alternative distances are increased if one of the original distances D_{p}, D_{f} or D_{c} is long, even though the other original distances are short: i.e., the two Fragments f_0 and f_i are comparatively similar from that viewpoint. In the Fragment clustering algorithm, if one of the three distances between a Fragment and the reference Fragment is the shortest of all distances, the Fragment should be preferentially selected as a candidate. Thus the mingled distance is not a useful criterion for determining similar Fragment candidates. In this study, the original distances are not mingled until the final stage of Fragment clustering.

4.2.3. Clustering based on individual number of candidates

In Fragment clustering, the number of candidates $\mathscr{N}(f_0)$ is determined by using the maximum number of candidates among the three types of distance as shown in Eq. (12). Other methods for determining $\mathscr{N}(f_0)$ can be discussed. For instance, each number of candidate Fragments $N_{\mathrm{p}}(f_0)$, $N_{\mathrm{f}}(f_0)$ and $N_{\mathrm{c}}(f_0)$ can be used in each Fragment list. In this case, the criterion of the Fragment classification is defined by

$$f_0' = \{f_i | O_{\mathrm{p}}(f_i) \leqslant N_{\mathrm{p}}(f_0) \cap O_{\mathrm{f}}(f_i) \leqslant N_{\mathrm{f}}(f_0) \cap O_{\mathrm{c}}(f_i) \leqslant N_{\mathrm{c}}(f_0)\}. \quad (16)$$

The advantage of this method is that Fragment clustering is performed by using the Fragment list made under a strict condition, and the clustering process will generate a Fragment grammar consisting of reliably similar Fragments. However, if Eq. (16) were applied to Fragment clustering and only few Fragments had short distances in a Fragment list, few Fragments would be selected as candidates so there may be no Fragment commonly listed within the ranking in all three Fragment subsets. As described in Section 4.2, a Fragment should be selected as the candidate as much as possible, even if just one of the three distances from the reference Fragment is short. The criterion based on the minimum number of candidates among three types of distance may cause the same result for many reference Frag-

a. Several Fragments

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

b. Fragment created from other Fragments

< 015 >	*"< 005 > to < 002 >"*	*"like to < 002 >"*		
	word sequences matching this 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"*		

Fig. 8. An example of Fragment generalization.

ments. Therefore, the clustering algorithm uses the maximum number of candidates $\mathcal{N}(f_0)$.

4.3. Generalization of Fragment grammar

Sometimes, a given Fragment contains substrings which are themselves Fragments with a higher frequency of occurrence than the given Fragment. When this is the case, the given Fragment can be "parsed" – i.e., the Fragment substrings are replaced by the appropriate non-terminal symbols representing them. The phrase "*want to make*" in Fragment ⟨015⟩ in Fig. 7, for instance, can be decomposed into "*want*", "*to*" and "*make*". The words "*want*" and "*make*" can be replaced by non-terminal symbols ⟨005⟩ and ⟨002⟩, respectively. Therefore, the phrase "*want to make*" can be represented as "⟨005⟩ *to* ⟨002⟩". This parsing allows the Fragment grammar to acquire the ability to represent not only phrases given as input but also word sequences not observed in the training transcriptions. Fig. 8 shows an example of Fragment generalization by parsing Fragments. In this example, the phrases in Fragment ⟨015⟩ are generalized by using Fragments ⟨002⟩ and ⟨005⟩. The three phrases in 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*". In consequence of this generalization, the Fragment ⟨015⟩ has acquired additional seven phrases such as "*want to place*", "*would like to make*" and "have to place".

The generalization of Fragments is performed in the order of Fragment frequency. When a set of Fragments has been created, the frequency of each Fragment is obtained by summing the frequencies of the phrases represented by that Fragment. A parser for Fragment generalization first sorts Fragments in order of Fragment frequency. The parser then selects a Fragment in the frequency

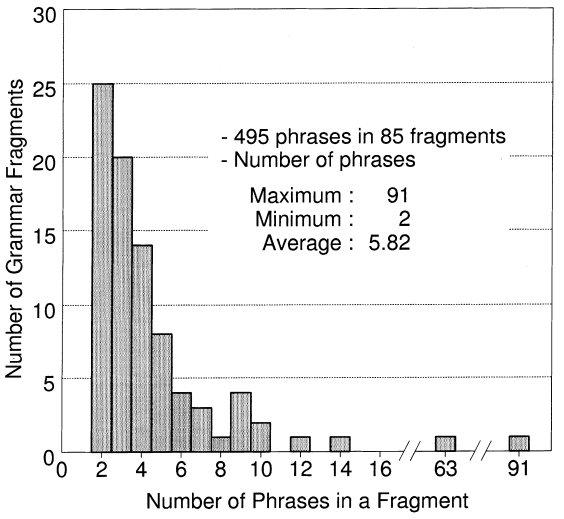

Fig. 9. Histogram of phrases in a Fragment.

order and performs the following procedure. The parser looks for each phrase classified in the selected Fragment inside other Fragments. When the corresponding word sequence is spotted in a Fragment, it is replaced by the appropriate non-terminal symbol representing the selected Fragment. This replacement is iterated until every Fragment has been selected at least once. Fig. 9 shows the histogram of the number of phrases in each Fragment. By applying Fragment generalization, 495 phrases were created in 85 Fragments. This reveals that the average number of phrases in a Fragment, 5.82 (495/85), was increased by generalization.

For call-type classification, *Salient Grammar Fragments* are automatically generated from the parsed training transcriptions and associated call-types (Gorin et al., 1997). Salient Grammar Fragments are traditionally generated by using the training transcriptions not parsed with the Fragment grammar. In this case, each Salient Grammar Fragment consists of a call-type of the highest association score and a corresponding word sequence. Namely, Salient Grammar Fragment is applied to only one word sequence. However, Salient Grammar Fragment, which can be applied to several kinds of word sequences, is generated if the training transcriptions parsed with Fragment grammar are used for the generation. In this case, the sequence corresponding to a call-type in each Salient Grammar Fragment consists of conventional words and non-terminal symbols for Fragments. Fig. 10 shows examples of a Salient Grammar Fragment. The Fragment grammar enables the Salient Grammar Fragment to represent several kinds of word sequences having both syntactic and semantic similarity. The call-type classification process is shown in Fig. 11. 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 using the call-type assigned to each test-set utterance.

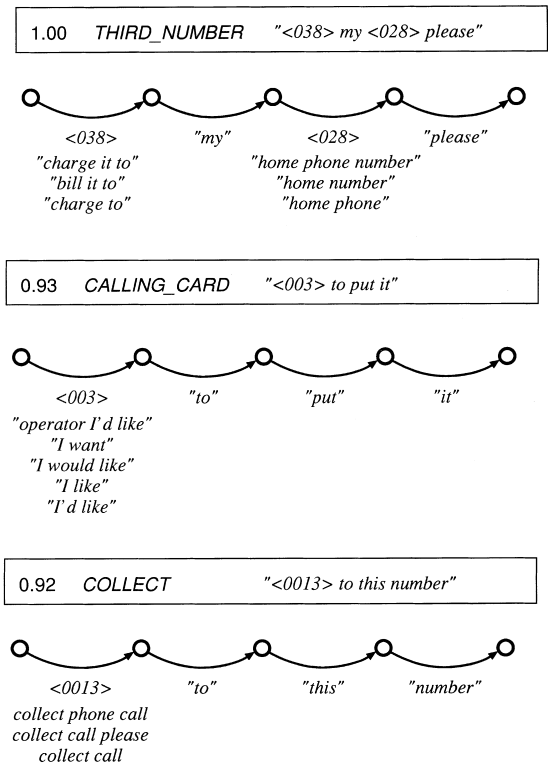

Fig. 10. An example of Salient Grammar Fragments.

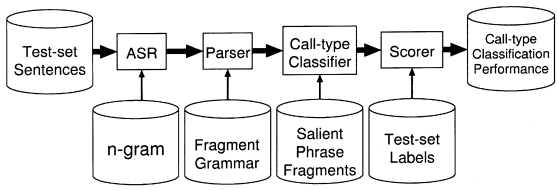

Fig. 11. Call-type classification process.

5. Experiments

The engine used for speech recognition is the AT&T WATSON recognizer (Sharp et al., 1997). The speech recognition process is performed by using the *Variable N-gram Stochastic Automaton* (*VNSA*) (Riccardi et al., 1996) as the language model. The acoustic model for the process was trained with a database of telephone-quality spontaneous utterances. Out-of-vocabulary words were not dealt with in the experiments. In order to confirm that there is no need to deal with out of

vocabulary words in this task, we examined the training transcriptions from the following two viewpoints.

Out-of-vocabulary words include many proper nouns such as city and personal names in this task. Therefore we first examined proper nouns in the training transcriptions. Words and phrases in the training transcriptions were classified into several categories by using word lists. Each word list consists of proper nouns such as country, city, or personal names. Some phrases can be classified into two or more categories. The phrase "*New York*", for instance, is either a city name or state name in the United States. Some word lists were prepared to store such words and phrases. In total, 1407 words and phrases were extracted from the training transcriptions and were classified into one of 16 categories. They were observed 5787 times in the training transcriptions. The other 2254 words, not classified into any category, were observed 136,179 times. Therefore, the token coverage of the proper nouns and other words is approximately 4.1% (= 5787/(5787 + 136,179)) and 95.9%, respectively.

The disposition of the out-of-vocabulary words can be estimated by examining *low frequency words* in the training transcriptions. We therefore counted the number of low frequency words in the training transcriptions as the next stage of this examination. Table 4 shows numbers of low frequency words and their numbers of tokens. For instance, the number of words observed twice in the training set was 316. The token coverage of the words observed three times or less was approximately 1.5%. The test set is assumed to have similar disposition to the training set, because the test set was created from the same set of transcriptions as the training set was created. The goal of this study is to extract the continuous word sequences frequently linked to the call-types. Words used sparsely are not important in this task. We therefore concluded that out-of-vocabulary words should not be dealt with in the experiments.

Table 4
Low frequency words

Word frequency	Number of words	Number of tokens
1	957	957
2	316	632
3	161	483
Total	1434	2072

The training transcription contained 7844 sentences while the test transcription contained 1000 sentences. For 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 clustering. A total of 1108 candidate phrases were obtained. The context length N_c for computing the distances between two Fragments was set to one. 3582 context phrases were used for creating the syntactic probability distributions. The maximum number of candidate Fragments, N_m, was 80.

In 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 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 misunderstandings that must be resolved by a dialog manager (Boyce and Gorin, 1996; Abella and Gorin, 1997). Fig. 12 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 Fig. 12. The curves in Fig. 12 were generated by varying salience threshold (Gorin, 1996).

The average of the difference in the call-type classification performance was calculated by using Eq. (17).

$$P_a = \frac{\int_{r_l}^{r_h} (p_{fg} - p_b)\, df}{r_h - r_l}, \tag{17}$$

where P_a is the average of the difference in the probability of correct classification between the language model as the baseline and that using Fragment grammar. p_{fg} and p_b denote the probabilities of correct classification achieved by the Fragment grammar and baseline, respectively. r_l and r_h define the range of the false rejection rate which were set to 7.4% and 48.3% for calculating

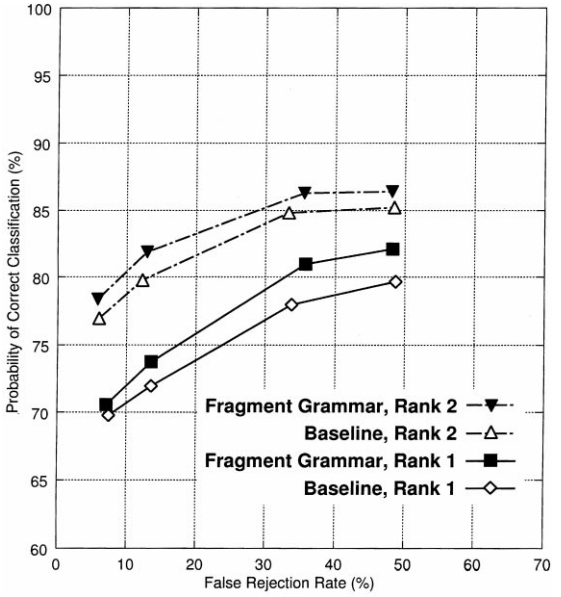

Fig. 12. Call-type classification performance.

the difference in rank 1. The probability of correct classification p_{fg} and p_b can be estimated by interpolation. By using p_{fg}, p_b and the aligned section of the false rejection rate r_l and r_h, the average of the difference in the probability of correct classification was evaluated using Eq. (17). As a result, the average difference in the call-type classification performance in rank 1 was approximately 2.2%.

On the other hand, the maximum difference in the probability of correct classification P_m was obtained by using Eq. (18).

$$P_m = \max(p_{fg} - p_b). \tag{18}$$

By interpolating the performance curves of rank 1, the difference in the probability of correct classification between baseline and Fragment grammar is obtained. The result revealed that approximately 2.8% of the maximum performance difference was observed where the false rejection rate was 35.7%. This result shows that the Fragment grammar achieved the best performance at rank 1 compared to the conventional approach when the false rejection rate was set to 35.7%.

In total, these results show that call-type classification performance is 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 original phrases used in the baseline system. From the results of this experiment, we can conclude that by generalizing Fragments, unobserved phrases are handled without degrading the call-type classification performance. An example of the variety of phrases accepted by a Salient Grammar Fragment is illustrated in Fig. 13. The Fragment "*BOS* ⟨017⟩ ⟨004⟩ ⟨013⟩" shown in Fig. 13 has an association with the call-type "*COLLECT*"; the association score is 0.97. The Fragment classes "⟨017⟩", "⟨004⟩" and "⟨013⟩" used in this Salient Grammar Fragment can be expanded into phrases and other Fragment grammars. Fragment "⟨017⟩", for instance, is expanded into two paths, "⟨003⟩" or "I ⟨005⟩". The consequence of this expansion, the fully expanded Salient Fragment network, is shown in Fig. 13(c). This phrase network accepts 126 types of phrases. It is interesting to note that

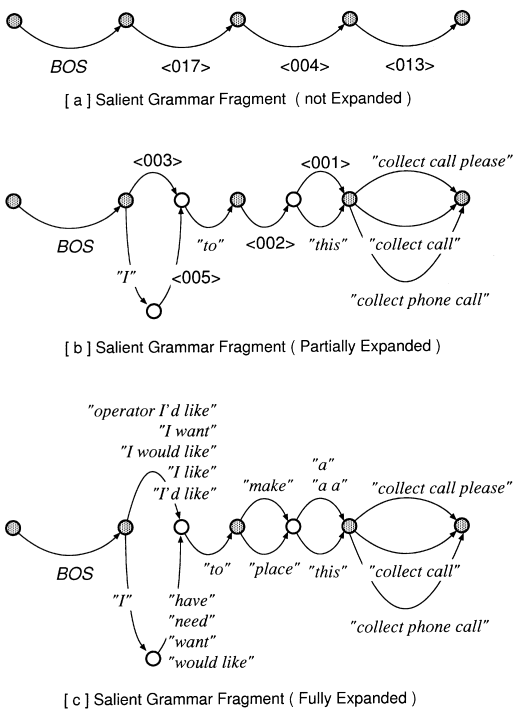

Fig. 13. An example of phrases accepted by a Salient Grammar Fragment.

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. Altogether, 246 unseen salient phrases have been discovered by clustering and generalizing the Fragment grammar.

6. Conclusion

We have described a new method for automatically acquiring Fragments for understanding fluent speech. Fragments representing a set of syntactically and semantically similar phrases were generated by using three probability distributions: of following 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 that were close in all three distances, were clustered into a Fragment. By spotting small Fragments inside larger phases, and then carrying out the appropriate substitutions of non-terminal symbols into the large phrases, the Fragment was able to detect 246 salient phrases in the test set that were not present in the training set. This result revealed that unseen phrases were automatically discovered by the proposed method. The experimental results show that the average and maximum improvements in call-type classification performance of 2.2% and 2.8% were achieved, respectively, by introducing the Fragment.

Future works of this study include improving Fragment performance on call-type classification. In this study, we focused on continuous word sequences in which the number of words was three or less, and whose frequency was 30 or more in the training transcriptions. The maximum number of candidate Fragments was constrained to be 80. By varying these parameter values for Fragment clustering, we obtained the best performance on call-type classification. However, when more training transcriptions become available, we will reconsider these parameter values because different sets of Fragment generated using different values may yield improved performance on call-type classification.

It is also interesting to introduce non-terminal symbols for the proper nouns common in this task. It is currently difficult for our definition of the Fragment distance to deal with low frequency phrases. Though many kinds of proper nouns such as city and personal names were observed only a few times in this task, a proper noun can be replaced by another if both of them are assigned to the same class. By introducing non-terminal symbols for such proper nouns, the training transcription is filtered and proper nouns are replaced by non-terminal symbols. The frequency of each non-terminal symbol can be high enough for calculating probability and the Fragment distance based on the Kullback–Leibler measure. This approach will make new Fragments consisting of not only word sequences but also non-terminal symbols.

Acknowledgements

The authors would like to thank Dr. Larry Rabiner, Jay Wilpon, Dr. Sadaoki Furui and Dr. Nobuhiko Kitawaki for their support and encouragement of this research.

Appendix A

In this appendix, the estimation of Fragment bigram probability is described with particular emphasis on the probability distribution for the following context $\hat{p}(s_i^{t+1}|f_1^t)$. This estimation method is used in the Fragment distance calculation. Probability distributions of preceding contexts and call-types can be also estimated in a similar manner. In general, the probability of a transition from Fragment f_1 to context s_i is calculated by Eq. (A.1).

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(f_1^t s_i^{t+1})}{C(f_1)}, \tag{A.1}$$

where $C(f_1)$ and $C(f_1^t s_i^{t+1})$ denotes the frequency of Fragment f_1 and the frequency of Fragment f_1 followed by context s_i, respectively. However, the probability for contexts having no transition from f_1 in the training set is poorly estimated by Eq. (A.1) since $C(f_1^t s_i^{t+1}) = 0$. Also, for some

contexts having low transition frequency, the probability cannot be calculated with high reliability. Therefore, a threshold τ is introduced to determine whether or not the transition frequency of a context is high enough for the probability to be estimated with Eq. (A.1). Eq. (A.1) is applied only if the transition frequency is equal to τ or more, i.e. $C(f_1^t s_i^{t+1}) \geqslant \tau$. Consequently, we conclude that there is a relationship between low and high transition frequency as shown in Eq. (A.2). Eq. (A.2) implies that the total probability for low frequency transition can be obtained by subtracting the total probability of high frequency transitions from 1.

$$\sum_{s_i \in S_l(f_1\tau)} \hat{p}(s_i^{t+1}|f_1^t) = 1 - \sum_{s_i \in S_h(f_1\tau)} \hat{p}(s_i^{t+1}|f_1^t). \tag{A.2}$$

In Eq. (A.2), $S_l(f_1\tau)$ and $S_h(f_1\tau)$, are the set of contexts based on context frequency and on the threshold τ as shown in Eqs. (A.3) and (A.4), respectively.

$$S_l(f_1\tau) = \{s_i | C(f_1^t s_i^{t+1}) < \tau\}, \tag{A.3}$$

$$S_h(f_1\tau) = \{s_i | C(f_1^t s_i^{t+1}) \geqslant \tau\}. \tag{A.4}$$

Eq. (A.5) is then applied to estimate the probability for the context having low transition frequency (Ney and Essen, 1993; Riccardi et al., 1996).

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(s_i)}{\sum_{s_j \in s_l(f_1\tau)} C(s_j)} \left\{ 1 - \sum_{s_j \in s_h(f_1\tau)} \hat{p}(s_j^{t+1}|f_1^t) \right\}. \tag{A.5}$$

In Eq. (A.5), the sum of transition probability for contexts having low transition frequency is first obtained by using the relationship shown in Eq. (A.2). The sum of probability for low frequency contexts is then distributed based on context unigram probability.

However, if every context following Fragment f_1 has transition frequency greater than τ, the sum of transition probabilities for contexts having low transition frequency is exactly zero in Eq. (A.2). In order to cope with this problem, the following equations are applied. First, the sum of probability for low frequency contexts, $\mathscr{L}(f_1)$ is introduced.

$$\mathscr{L}(f_1) = \sum_{s_j \in S_l(f_1\tau)} C(f_1^t s_i^{t+1}). \tag{A.6}$$

If $\mathscr{L}(f_1)$ is not equal to zero, Eqs. (A.1) and (A.5) can be applied to estimate the probability. On the other hand, if $\mathscr{L}(f_1)$ is equal to zero, Eq. (A.7) is applied to estimate the transition probability for contexts having high transition frequency.

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(f_1^t s_i^{t+1})}{C(f_1) + \delta}, \tag{A.7}$$

where δ is a small constant for all Fragments. Use of Eq. (A.7) in the case of $\mathscr{L}(f_1) = 0$ implies that the sum of transition probability for contexts having high transition frequency can be less than 1 as shown in Eq. (A.8).

$$\sum_{s_i \in S_h(f_1\tau)} \hat{p}(s_i^{t+1}|f_1^t) = \sum_{\forall s_i} \frac{C(f_1^t s_i^{t+1})}{C(f_1) + \delta} = \frac{C(f_1)}{C(f_1) + \delta} < 1. \tag{A.8}$$

As a consequence of using of Eqs. (A.5) and (A.7), the probability for contexts with low transition frequency can be estimated by Eq. (A.9).

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(s_i)}{\sum_{s_j \in S_l(f_1\tau)} C(s_j)} \left\{ 1 - \frac{C(f_1)}{C(f_1) + \delta} \right\} = \frac{C(s_i)}{\sum_{s_j \in S_l(f_1\tau)} C(s_j)} \frac{\delta}{C(f_1) + \delta}. \tag{A.9}$$

Thus the following equations are applied for probability estimation.

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(f_1^t s_i^{t+1})}{C(f_1)} \quad \text{if } \mathscr{L}(f_1) > 0 \quad \text{and} \quad s_i \in S_h(f_1\tau), \tag{A.10}$$

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(s_i)}{\sum_{s_j \in S_l(f_1\tau)} C(s_j)} \left\{ 1 - \sum_{s_j \in S_h(f_1\tau)} \frac{C(f_1^t s_j^{t+1})}{C(f_1)} \right\} \quad \text{if } \mathscr{L}(f_1) > 0 \quad \text{and} \quad s_i \in S_l(f_1\tau), \tag{A.11}$$

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(f_1^t s_i^{t+1})}{C(f_1) + \delta} \quad \text{if } \mathscr{L}(f_1) = 0 \quad \text{and} \quad s_i \in S_h(f_1\tau), \tag{A.12}$$

$$\hat{p}(s_i^{t+1}|f_1^t) = \frac{C(s_i)}{\sum_{s_j \in S_l(f_1\tau)} C(s_j)} \frac{\delta}{C(f_1)+\delta}$$
$$\text{if } \mathscr{L}(f_1) = 0 \quad \text{and} \quad s_i \in S_l(f_1\tau). \tag{A.13}$$

References

Abella, A., Gorin, A.L., 1997. Generating semantically consistent inputs to a dialog manager. In: Proc. of EuroSpeech'97, Vol. 4, pp. 1879–1882.

Bellegarda, J.R., Butzberger, J.W., Chow, Y.L., Coccaro, N.B., Naik, D., 1996. A novel word clustering algorithm based on latent semantic analysis. In: Proc. of ICASSP'96, Vol. 1, pp. 172–175.

Boyce, S., Gorin, A.L., 1996. User interface issues for natural spoken dialog systems, in: Proc. of Internat. Symp. on Spoken Dialog (ISSD), pp. 65–88.

Brown, P.F., Pietra, V.J.D., deSouza, P.V., Lai, J.C., Mercer, R.L., 1992. Class-based n-gram models of natural language. Computational Linguistics 18, 467–479.

Farhat, A., Isabelle, J.F., O'Shaughnessy, D., 1996. Clustering words for statistical language models based on contextual word similarity. In: Proc. of ICASSP'96, Vol. 1, pp. 180–183.

Giachin, E., 1995. Phrase bigrams for continuous speech recognition. In: Proc. of ICASSP'95, Vol. 1, pp. 225–228.

Gorin, A.L., 1995. On automated language acquisition. Journal of the Acoustic Society of America 97 (6), 3441–3461.

Gorin, A.L., 1996. Processing of semantic information in fluently spoken language. In: Proc. of ICSLP'96, Vol. 2, pp. 1001–1004.

Gorin, A.L., Riccardi, G., Wright, J.H., 1997. How may I help you? Speech Communication 23 (1–2), 113–127.

Masataki, H., Sagisaka, Y., 1996. Variable-order N-gram generation by word-class splitting and consecutive word grouping. In: Proc. of ICASSP'96, Vol. 1, pp. 188–191.

Ney, H., Essen, U., 1993. Estimating small probabilities by leaving one out. In: Proc. of EuroSpeech'93, Vol. 3, pp. 2239–2242.

Riccardi, G., Pieraccini, R., Bocchieri, E., 1996. Stochastic automata for language modeling. Computer Speech and Language 10, 265–293.

Riccardi, G., Gorin, A.L., Ljolje, A., Riley, M.D., 1997. A spoken language system for automated call routing. In: Proc. of ICASSP'97, Vol. 2, pp. 1143–1146.

Sharp, R.D., Bocchieri, E., Castillo, C., Parthasarathy, S., Rath, C., Riley, M.D., Rowland, J., 1997. The WATSON speech recognition engine. In: Proc. of ICASSP'97, Vol. 5, pp. 4065–4068.

Ward, W., Issar, S., 1996. A class based language model for speech recognition. In: Proc. of ICASSP'96, Vol. 1, pp. 416–418.

Wright, J., Gorin, A.L., Riccardi, G., 1997. Automatic acquisition of salient grammar fragments for call-type classification. In: Proc. of EuroSpeech'97, Vol. 3, pp. 1419–1422.