Learning to Model Domain-Specific Utterance Sequences for Extractive Summarization of Contact Center Dialogues

Ryuichiro Higashinaka[†], **Yasuhiro Minami**[‡], **Hitoshi Nishikawa**[†],
Kohji Dohsaka[‡], **Toyomi Meguro**[‡], **Satoshi Takahashi**[†], **Genichiro Kikui**[†]
† NTT Cyber Space Laboratories, NTT Corporation
‡ NTT Communication Science Laboratories, NTT Corporation
higashinaka.ryuichiro@lab.ntt.co.jp, minami@cslab.kecl.ntt.co.jp
nishikawa.hitoshi@lab.ntt.co.jp, {dohsaka,meguro}@cslab.kecl.ntt.co.jp
{takahashi.satoshi,kikui.genichiro}@lab.ntt.co.jp

Abstract

This paper proposes a novel extractive summarization method for contact center dialogues. We use a particular type of hidden Markov model (HMM) called Class Speaker HMM (CSHMM), which processes operator/caller utterance sequences of multiple domains simultaneously to model domain-specific utterance sequences and common (domain-wide) sequences at the same time. We applied the CSHMM to call summarization of transcripts in six different contact center domains and found that our method significantly outperforms competitive baselines based on the maximum coverage of important words using integer linear programming.

1 Introduction

In modern business, contact centers are becoming more and more important for improving customer satisfaction. Such contact centers typically have quality analysts who mine calls to gain insight into how to improve business productivity (Takeuchi et al., 2007; Subramaniam et al., 2009). To enable them to handle the massive number of calls, automatic summarization has been utilized and shown to successfully reduce costs (Byrd et al., 2008). However, one of the problems in current call summarization is that a domain ontology is required for understanding operator/caller utterances, which makes it difficult to port one summarization system from domain to domain.

This paper describes a novel automatic summarization method for contact center dialogues without the costly process of creating domain ontologies. More specifically, given contact center dialogues categorized into multiple domains, we create a particular type of hidden Markov model (HMM) called **Class Speaker HMM** (**CSHMM**) to model operator/caller utterance sequences. The CSHMM learns to distinguish sequences of individual domains and common sequences in all domains at the same time. This approach makes it possible to accurately distinguish utterances specific to a certain domain and thereby has the potential to generate accurate extractive summaries.

In Section 2, we review recent work on automatic summarization, including its application to contact center dialogues. In Section 3, we describe the CSHMM. In Section 4, we describe our automatic summarization method in detail. In Section 5, we describe the experiment we performed to verify our method and present the results. In Section 6, we summarize and mention future work.

2 Related Work

There is an abundance of research in automatic summarization. It has been successfully applied to single documents (Mani, 2001) as well as to multiple documents (Radev et al., 2004), and various summarization methods, such as the conventional LEAD method, machine-learning based sentence selection (Kupiec et al., 1995; Osborne, 2002), and integer linear programming (ILP) based sentence extraction (Gillick and Favre, 2009), have been proposed. Recent years have seen work on summarizing broadcast news speech (Hori and Furui, 2003), multi-party meetings (Murray et al., 2005), and contact center dialogues (Byrd et al., 2008). However, despite the large amount of previous work, little work has tackled the automatic summarization of multi-domain data.

Coling 2010: Poster Volume, pages 400–408,
Beijing, August 2010

In the past few decades, contact center dialogues have been an active research focus (Gorin et al., 1997; Chu-Carroll and Carpenter, 1999). Initially, the primary aim of such research was to transfer calls from answering agents to operators as quickly as possible in the case of problematic situations. However, real-time processing of calls requires a tremendous engineering effort, especially when customer satisfaction is at stake, which led to recent work on the offline processing of calls, such as call mining (Takeuchi et al., 2007) and call summarization (Byrd et al., 2008).

The work most related to ours is (Byrd et al., 2008), which maps operator/caller utterances to an ontology in the automotive domain by using support vector machines (SVMs) and creates a structured summary by heuristic rules that assign the mapped utterances to appropriate summary sections. Our work shares the same motivation as theirs in that we want to make it easier for quality analysts to analyze the massive number of calls. However, we tackle the problem differently in that we propose a new modeling of utterance sequences for extractive summarization that makes it unnecessary to create heuristics rules by hand and facilitates the porting of a summarization system.

HMMs have been successfully applied to automatic summarization (Barzilay and Lee, 2004). In their work, an HMM was used to model the transition of *content topics*. The Viterbi decoding (Rabiner, 1990) was performed to find content topics that should be incorporated into a summary. Their approach is similar to ours in that HMMs are utilized to model topic sequences, but they did not use data of multiple domains in creating their model. In addition, their method requires training data (original articles with their reference summaries) in order to find which content topics should be included in a summary, whereas our method requires only the raw sequences with their domain labels.

3 Class Speaker HMM

A Class Speaker HMM (CSHMM) is an extension of Speaker HMM (SHMM), which has been utilized to model two-party conversations (Meguro et al., 2009). In an SHMM, there are two states, and each state emits utterances of one of the two conversational participants. The states are connected ergodically and the emission/transition probabilities are learned from training data by using the EM-algorithm. Although Meguro et al., (2009) used SHMMs to analyze the flow of listening-oriented dialogue, we extend their idea to make it applicable to classification tasks, such as dialogue segmentation.

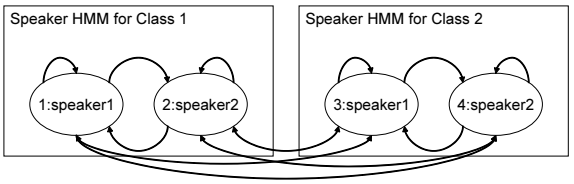

Figure 1: Topology of an ergodic CSHMM. Numbers before 'speaker1' and 'speaker2' denote state IDs.

A CSHMM is simply a concatenation of SHMMs, each of which is trained by using utterance sequences of a particular dialogue class. After such SHMMs are concatenated, the Viterbi algorithm is used to decode an input utterance sequence into class labels by estimating from which class each utterance has most likely to have been generated. Figure 1 illustrates the basic topology of a CSHMM where two SHMMs are concatenated ergodically. When the most likely state sequence for an input utterance sequence is <1,3,4,2>, we can convert these state IDs into their corresponding classes; that is, <1,2,2,1>, which becomes the result of utterance classification.

We have conceived three variations of CSHMM as we describe below. They differ in how we treat utterance sequences that appear commonly in all classes and how we train the transition probabilities between independently trained SHMMs.

3.1 Ergodic CSHMM

The most basic CSHMM is the ergodic CSHMM, which is a simple concatenation of SHMMs in an ergodic manner as shown in Fig. 1. For K classes, K SHMMs are combined with the initial and transition probabilities all set to *equal*. In this CSHMM, the assignment of class labels solely depends on the output distributions of each class.

3.2 Ergodic CSHMM with Common States

This type of CSHMM is the same as the ergodic CSHMM except that it additionally has a SHMM trained from all dialogues of all classes. There-

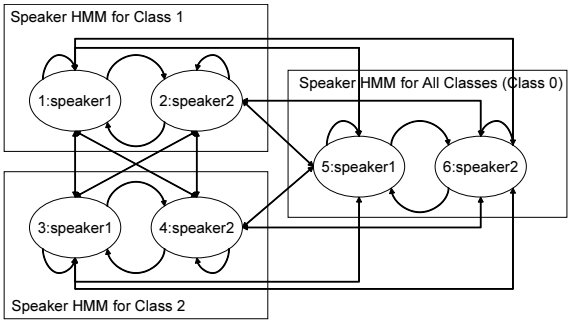

Figure 2: CSHMM with common states.

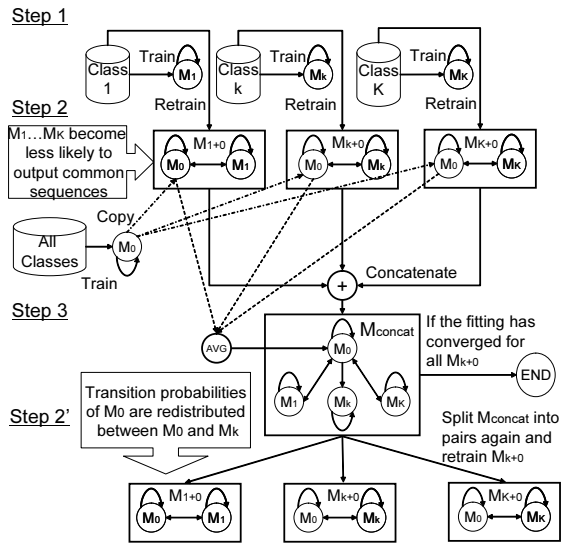

Figure 3: Three steps to create a CSHMM using concatenated training.

fore, for K classes, this CSHMM has $K + 1$ SHMMs. Figure 2 shows the model topology. This newly added SHMM works in a manner similar to the background model (Reynolds et al., 2000) representing sequences that are common to all classes. By having these *common states*, common utterance sequences can be classified as 'common', making it possible to avoid forcefully classifying common utterance sequences into one of the given classes.

Detecting common sequences is especially helpful when several classes overlap in nature. For example, most dialogues commonly start and end with greetings, and many calls at contact centers commonly contain exchanges in which the operator requests personal information about the caller for confirmation. Regarding the model topology in Fig. 2, if the most likely state sequence by the Viterbi decoding is <1,4,5,6,3,2>, we obtain a class label sequence <1,2,0,0,2,1> where the third and fourth utterances are classified as 'zero', meaning that they do not belong to any class.

3.3 CSHMM using Concatenated Training

The CSHMMs presented so far have two problems: one is that the order of utterances of different classes cannot be taken into account because of the equal transition probabilities. As a result, the very merit of HMMs, their ability to model time series data, is lost. The other is that the output distributions of common states may be overly broad because they are the averaged distributions over all classes; that is, the best path determined by the Viterbi decoding may not go through the common states at all.

Our solution to these problems is to apply concatenated training (Lee, 1989), which has been successfully used in speech recognition to model phoneme sequences in an unsupervised manner. The procedure for concatenated training is illustrated in Fig. 3 and has three steps.

step 1 Let M_k ($M_k \in M, 1 \leq k \leq K$) be the SHMM trained using dialogues D_k where $D_k = \{\forall d_j | c(d_j) = k\}$, and M_0 be the SHMM trained using all dialogues; i.e., D. Here, K means the total number of classes and $c(d_j)$ the class assigned to a dialogue d_j.

step 2 Connect each $M_k \in M$ with a copy of M_0 using equal initial and transition probabilities (we call this connected model M_{k+0}) and retrain M_{k+0} with $\forall d_j \in D_k$ where $c(d_j) = k$.

step 3 Merge all models M_{k+0} ($1 \leq k \leq K$) to produce one concatenated HMM (M_{concat}). Here, the output probabilities of the copies of M_0 are averaged over K when all models are merged to create a combined model. If the fitting of all M_{k+0} models has converged against the training data, exit this procedure; otherwise, go to step 2 by connecting a copy of M_0 and M_k for all k. Here, the transition probabilities from M_0 to $M_l (l \neq k)$ are summed and equally distributed between the copied M_0's self-loop and transitions to the states in M_k.

In concatenated training, the transition and output probabilities can be optimized between M_0 and

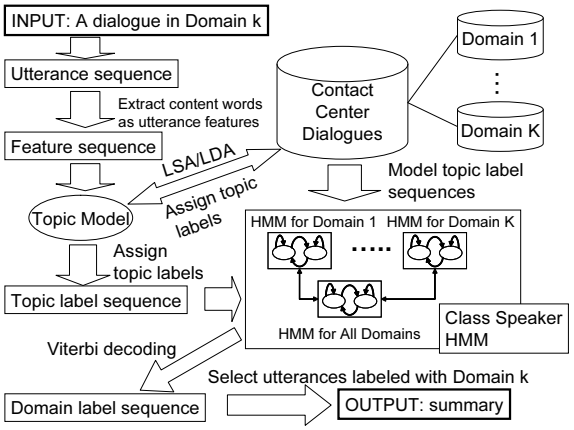

Figure 4: Overview of our summarization method.

M_k, meaning that the output probabilities of utterance sequences that are common and also found in M_k can be moved from M_k to M_0. This makes the distribution of M_k sharp (not broad/uniform), making it likely to output only the utterances representative of a class k. As regards M_0, its distribution of output probabilities can also be sharpened for utterances that occur commonly in all classes. This sharpening of distributions is likely to be helpful for class discrimination.

4 Summarization Method

We apply CSHMMs to extractive summarization of contact center dialogues because such dialogues are two-party, can be categorized into multiple classes by their call domains (e.g., inquiry types), and are likely contain many overlapping exchanges between an operator and a caller across domains, such as greetings, the confirmation of personal information, and other cliches in business (e.g., name exchanges, thanking/apologizing phrases, etc.), making them the ideal target for CSHMMs.

In our method, summarization is performed by decoding a sequence of utterances of a domain DM^k into domain labels and selecting those utterances that have domain labels DM^k. This makes it possible to extract utterances that are characteristic of DM^k in relation to other domains. Our assumption is that extracting characteristic sequences of a given domain provides a good summary for that domain because such sequences should contain important information necessitated by the domain.

Figure 4 outlines our extractive summarization process. The process consists of a training phase and a decoding phase as described below.

Training phase: Let D ($d_1 \ldots d_N$) be the entire set of contact center dialogues, DM^k ($DM^k \in DM, 1 \leq k \leq K$) the domain assigned to domain k, and $U_{d_i,1} \ldots U_{d_i,H}$ the utterances in d_i. Here, H is the number of utterances in d_i. From D, we create two models: a topic model (TM) and a CSHMM.

The topic model is used to assign a single topic to each utterance so as to facilitate the training of the CSHMM by reducing the dimensions of the feature space. The same approach has been taken in (Barzilay and Lee, 2004). The topic model can be created by such techniques as probabilistic latent semantic analysis (PLSA) (Šingliar and Hauskrecht, 2006) and latent Dirichlet allocation (LDA) (Tam and Schultz, 2005). PLSA models the latent topics of the documents and its Baysian extension is LDA, which also models the co-occurrence of topics using the Dirichlet prior. We first derive features $F_{d_1} \ldots F_{d_N}$ for the dialogues. Here, we assume a bag-of-words representation for the features; therefore, F_{d_i} is represented as $\{< w_1, c_1 > \ldots < w_V, c_V >\}$, where V means the total number of content words in the vocabulary and $< w_i, c_i >$ denotes that a content word w_i appears c_i times in a dialogue. Note that we derive the features for dialogues, not for utterances, because utterances in dialogue can be very short, often consisting of only one or two words and thus making it hard to calculate the word co-occurrence required for creating a topic model. From the features, we build a topic model that includes $\mathrm{P}(z|w)$, where w is a word and z is a topic. Using the topic model, we can assign a single topic label to every utterance in D by finding its likely topic; i.e., $\underset{z}{\mathrm{argmax}} \sum_{w \in \mathrm{words}(U_{d_i})} \mathrm{P}(z|w)$.

After labeling all utterances in D with topic labels, we train a CSHMM that learns characteristic topic label sequences in each domain as well as common topic label sequences across domains.

Decoding phase: Let d_j be the input dialogue, $DM(d_j)$ ($\in DM$) the table for obtaining the domain label of d_j, and $U_{d_j,1} \ldots U_{d_j,H_{d_j}}$ the utterances in d_j, where H_{d_j} is the number of the utterances. We use TM to map the utterances to topic

Domain	# Tasks	Sentences	Characters
FIN	15	8.93	289.93
ISP	15	7.20	259.53
LGU	20	9.85	328.55
MO	15	10.07	326.20
PC	15	9.40	354.07
TEL	18	8.44	322.22
ALL	98	9.01	314.46

Table 1: Scenario statistics: the number of tasks and averaged number of sentences/characters in a task scenario in the six domains.

labels $T_{d_j,1} \ldots T_{d_j,H_{d_j}}$ and convert them into domain label sequences $DM_{d_j,1} \ldots DM_{d_j,H_{d_j}}$ using the trained CSHMM by the Viterbi decoding. Then, we select $U_{d_j,h}$ $(1 \leq h \leq H_{d_j})$ whose corresponding domain label $DM_{d_j,h}$ equals $DM(d_j)$ and output the selected utterances in the order of appearance in the original dialogue as a summary.

5 Experiment

We performed an experiment to verify our summarization method. We first collected simulated contact center dialogues using human subjects. Then, we compared our method with baseline systems. Finally, we analyzed the created summaries to investigate what had been learned by our CSHMMs.

5.1 Dialogue Data

Since we do not have access to actual contact center data, we recruited human subjects to collect simulated contact center dialogues. A total of 90 participants (49 males and 41 females) took the roles of operator or a caller and talked over telephones in separate rooms. The callers were given realistic scenarios that included their motivation for a call as well as detailed instructions about what to ask. The operators, who had experience of working at contact centers, were given manuals containing the knowledge of the domain and explaining how to answer questions in specific scenarios.

The dialogues took place in six different domains: Finance (**FIN**), Internet Service Provider (**ISP**), Local Government Unit (**LGU**), Mail Order (**MO**), PC support (**PC**), and Telecommunication (**TEL**). In each domain, there were 15–20 tasks. Table 1 shows the statistics of the task scenarios used by the callers. We cannot describe the details of each domain for lack of space, but examples of the tasks for FIN are inquiries about insurance, notifications of the loss of credit cards, and applications for finance loans, and those for ISP are inquiries about fees for Internet access, requests to forward emails, and reissuance of passwords. Figure 5 shows one of the task scenarios in the MO domain.

MO task No. 3: It is becoming a good season for the Japanese Nabe (pan) cuisine. You own a Nabe restaurant and it is going well. When you were searching on the Internet, thinking of creating a new dish, you saw that drop-shipped Shimonoseki puffer fish was on sale. Since you thought the puffer fish cuisine would become hot in the coming season, you decided to order it as a trial. ... You ordered a puffer fish set on the Internet, but you have not received the confirmation email that you were supposed to receive. ... You decided to call the contact center to make an inquiry, ask them whether the order has been successful, and request them to send you the confirmation email.

Figure 5: Task scenario in the MO domain. The scenario was originally in Japanese and was translated by the authors.

We collected data on two separate occasions using identical scenarios but different participants, which gave us two sets of dialogue data. We used the former for training our summarization system and the latter for testing. We only use the transcriptions in this paper so as to avoid particular problems of speech. All dialogues were in Japanese. Tables 2 and 3 show the statistics of the training data and the test data, respectively. As can be seen from the tables, each dialogue is quite long, which attests to the complexity of the tasks.

5.2 Training our Summarization System

For training our system, we first created a topic model using LDA. We performed a morphological analysis using ChaSen[1] to extract content words from each dialogue and made its bag-of-words features. We defined content words as nouns, verbs, adjectives, unknown words, and interjections (e.g., "yes", "no", "thank you", and "sorry"). We included interjections because they occur very frequently in dialogues and often possess important content, such as agreement and refusal, in transactional communication. We use this definition of content words throughout the paper.

Then, using an LDA software package[2], we built a topic model. We tentatively set the number

[1] http://chasen-legacy.sourceforge.jp/

[2] http://chasen.org/~daiti-m/dist/lda/

		Utterances/Dial.			Characters/Utt.		
Domain	# dial.	OPE	CAL	Both	OPE	CAL	Both
FIN	59	75.73	72.69	148.42	17.44	7.54	12.59
ISP	64	55.09	53.17	108.27	20.11	8.03	14.18
LGU	76	58.28	50.55	108.83	12.83	8.55	10.84
MO	70	66.39	58.74	125.13	15.09	7.43	11.49
PC	56	89.34	77.80	167.14	15.48	6.53	11.31
TEL	66	75.58	63.97	139.55	12.74	8.24	10.67
ALL	391	69.21	61.96	131.17	15.40	7.69	11.76

Table 2: Training data statistics: Averaged number of utterances per dialogue and characters per utterance for each domain. OPE and CAL denote operator and caller, respectively. See Section 5.1 for the full domain names.

		Utterances/Dial.			Characters/Utt.		
Domain	# dial.	OPE	CAL	Both	OPE	CAL	Both
FIN	60	73.97	61.05	135.02	14.53	7.50	11.35
ISP	59	76.08	61.24	137.32	15.43	6.94	11.65
LGU	56	66.55	51.59	118.14	14.54	7.53	11.48
MO	47	75.53	64.87	140.40	10.53	6.79	8.80
PC	44	124.02	94.16	218.18	14.23	7.79	11.45
TEL	41	93.71	68.54	162.24	13.94	7.85	11.37
ALL	307	83.07	65.69	148.76	13.98	7.41	11.08

Table 3: Test data statistics.

of topics to 100. Using this topic model, we labeled all utterances in the training data using these 100 topic labels.

We trained seven different CSHMMs in all: one ergodic CSHMM (**ergodic0**), three variants of ergodic CSHMMs with common states (**ergodic1**, **ergodic2**, **ergodic3**), and three variants of CSHMMs with concatenated training (**concat1**, **concat2**, **concat3**). The difference within the variants is in the number of common states. The numbers 0–3 after 'ergodic' and 'concat' indicate the number of SHMMs containing common states. For example, ergodic3 has nine SHMMs (six SHMMs for the six domains plus three SHMMs containing common states). Since more states would enable more minute modeling of sequences, we made such variants in the hope that common sequences could be more accurately modeled. We also wanted to examine the possibility of creating sharp output distributions in common states without the concatenated training by such minute modeling. These seven CSHMMs make seven different summarization systems.

5.3 Baselines

Baseline-1: BL-TF We prepared two baseline systems for comparison. One is a simple summarizer based on the maximum coverage of high term frequency (TF) content words. We call this baseline BL-TF. This baseline summarizes a dialogue by maximizing the following objective function:

$$\max \sum_{z_i \in Z} \text{weight}(w_i) \cdot z_i$$

where 'weight' returns the importance of a content word w_i and z_i is a binary value indicating whether to include w_i in the summary. Here, 'weight' returns the count of w_i in a given dialogue. The maximization is done using ILP (we used an off-the-shelf solver `lp_solve`[3]) with the following three constraints:

$$x_i, z_i \in \{0, 1\}$$

$$\sum_{x_i \in X} l_i x_i \leq K$$

$$\sum_i m_{ij} x_i \geq z_j \quad (\forall z_j \in Z)$$

where x_i is a binary value that indicates whether to include the i-th utterance in the summary, l_i is the length of the i-th utterance, K is the maximum number of characters to include in a summary, and m_{ij} is a binary value that indicates whether w_i is included in the j-th utterance. The last constraint means that if a certain utterance is included in the summary, all words in that utterance have to be included in the summary.

Baseline-2: BL-DD Although BL-TF should be a very competitive baseline because it uses the state-of-the-art formulation as noted in (Gillick and Favre, 2009), having only this baseline is rather unfair because it does not make use of the training data, whereas our proposed method uses them. Therefore, we made another baseline that learns domain-specific dictionaries (DDs) from the training data and incorporates them into the weights of content words of the objective function of BL-TF. We call this baseline BL-DD. In this baseline, the weight of a content word w_i in a domain DM^k is

$$\text{weight}(w_{i,DM^k}) = \frac{\log(\text{P}(w_i|DM^k))}{\log(\text{P}(w_i|DM \backslash DM^k))}$$

[3] http://lpsolve.sourceforge.net/5.5/

	Metric	ergodic0	ergodic1	ergodic2	ergodic3	concat1	concat2	concat3
PROPOSED	F	0.177	0.177	0.177	0.177	0.187^{*e0e1}_{e2e3}	0.198^{*+e0e1}_{e2e3c1}	$\mathbf{0.199}^{*+e0e1}_{e2e3c1}$
	precision	0.145	0.145	0.145	0.145	0.161^{*}	0.191^{*+}	$\mathbf{0.195}^{*+}$
	recall	**0.294**	**0.294**	**0.294**	**0.294**	0.280^{*}	0.259^{*+}	0.259^{*+}
(Same-length) BL-TF	F	**0.171**	**0.171**	**0.171**	**0.171**	0.168	0.164	0.163
	precision	0.132	0.132	0.132	0.132	0.135	**0.140**	0.140
	recall	**0.294**	**0.294**	**0.294**	**0.294**	0.270	0.241	0.240
(Same-length) BL-DD	F	0.189	0.189	0.189	0.189	**0.189**	0.187	0.187
	precision	0.155	0.155	0.155	0.155	0.162	0.170	**0.172**
	recall	**0.287**	**0.287**	**0.287**	**0.287**	0.273	0.250	0.248
Compression Rate		0.42	0.42	0.42	0.42	0.37	0.30	0.30

Table 4: F-measure, precision, and recall averaged over all 307 dialogues (cf. Table 3) in the test set for the proposed methods and baselines BL-TF and BL-DD configured to output the same-length summaries as the proposed systems. The averaged compression rate for each proposed system is shown at the bottom. The columns (ergodic0–concat3) indicate our methods as well as the character lengths used by the baselines. Asterisks, '+', e0–e3, and c1–c3 indicate our systems' statistical significance by the Wilcoxon signed-rank test ($p<0.01$) over BL-TF, BL-DD, ergodic0–3, and concat1–3, respectively. Statistical tests for the precision and recall were only performed between the proposed systems and their same-length baseline counterparts. **Bold font** indicates the best score in each row.

where $\mathrm{P}(w_i|DM^k)$ denotes the occurrence probability of w_i in the dialogues of DM^k, and $\mathrm{P}(w_i|DM\backslash DM^k)$ the occurrence probability of w_i in all domains except for DM^k. This log likelihood ratio estimates how much a word is characteristic of a given domain. Incorporating such weights would make a very competitive baseline.

5.4 Evaluation Procedure

We made our seven proposed systems and two baselines (BL-TF and BL-DD) output extractive summaries for the test data. Since one of the shortcomings of our proposed method is its inability to set the compression rate, we made our systems output summaries first and made the baseline systems output their summaries within the character lengths of our systems' summaries.

We used scenario texts (See Fig. 5) as reference data; that is, a dialogue dealing with a certain task is evaluated using the scenario text for that task. As an evaluation criterion, we used the F-measure (F1) to evaluate the retrieval accuracy on the basis of the recall and precision of retrieved content words. We used the scenarios as references because they contain the basic content exchanged between an operator and a caller, the retrieval accuracy of which should be important for quality analysts.

We could have used ROUGE (Lin and Hovy, 2003), but we did not because ROUGE does not correlate well with human judgments in conversational data (Liu and Liu, 2008). Another benefit of using the F-measure is that summaries of varying lengths can be compared.

5.5 Results

Table 4 shows the evaluation results for the proposed systems and the baselines. It can be seen that concat3 shows the best performance in F-measure among all systems, having a statistically better performance over all systems except for concat2. The CSHMMs with concatenated training were all better than ergodic0–3. Here, the performance (and output) of ergodic0–3 was exactly the same. This happened because of the broad distributions in their common states; no paths went through the common states and all paths went through the SHMMs of the six domains instead.

The evaluation results in Table 4 may be rather in favor of our systems because the summarization lengths were set by the proposed systems. Therefore, we performed another experiment to investigate the performance of the baselines with varying compression rates and compared their performance with the proposed systems in F-measure. We found that the best performance was achieved by BL-DD when the compression rate was 0.4 with the F-measure of 0.191, which concat3 significantly outperformed by the Wilcoxon signed-rank test ($p<0.01$). Note that the performance shown in Table 4 may seem low. However, we found that the maximum recall is 0.355 (cal-

CAL1	When I order a product from you, I get a confirmation email
CAL2	**Puffer fish**
CAL3	**Sets I have ordered, but I haven't received the confirmation email**
OPE1	**Order**
OPE2	**I will make a confirmation whether you have ordered**
OPE3	Ten sets of Shimonoseki puffer fish by dropship
OPE4	"Yoriai" (name of the product)
OPE5	**Two kilos of bony parts of tiger puffer fish**
OPE6	Baked fins for fin sake
OPE7	**600 milliliter of puffer fish soy sauce**
OPE8	**And, grated radish and red pepper**
OPE9	Your desired delivery date is the 13th of February
CAL4	Yes, all in small cases
CAL5	This is q in alphabet right?
CAL6	Hyphen g
CAL7	**You mean that the order was successful**
OPE10	Yes, it was Nomura at JDS call center

Figure 6: Example output of concat3 for MO task No. 3 (cf Fig. 5). The utterances were translated by the authors. The compression rate for this dialogue was 0.24.

culated by using summaries with no compression). This means that the maximum F-measure we could attain is lower than 0.524 (when the precision is ideal with 1). This is because of the differences between the scenarios and the actual dialogues. We want to pursue ways to improve our evaluation methodology in the future.

Despite such issues in evaluation, from the results, we conclude that our extractive summarization method is effective and that having the common states and training CSHMMs with concatenated training are useful in modeling domain-specific sequences of contact center dialogues.

5.6 Example of System Output

Figure 6 shows an example output of concat3 for the scenario MO task No. 3 (cf. Fig. 5). **Bold font** indicates utterances that were NOT included in the summary of concat3's same-length-BF-DD counterpart. It is clear that sequences related to the MO domain were successfully extracted. When we look at the summary of BF-DD, we see such utterances as "*Can I have your address from the postcode*" and "*Finally, can I have your email address*", which are obvious cliches in contact center dialogues. This indicates the usefulness of common states for ignoring such common exchanges.

6 Summary and Future Work

This paper proposed a novel extractive summarization method for contact center dialogues. We devised a particular type of HMM called CSHMM, which processes operator/caller utterance sequences of multiple domains simultaneously to model domain-specific utterance sequences and common sequences at the same time. We trained a CSHMM using the transcripts of simulated contact center dialogues and verified its effectiveness for the summarization of calls.

There still remain several limitations in our approach. One is its inability to change the compression rate, which we aim to solve in the next step using the forward-backward algorithm (Rabiner and Juang, 1986). This algorithm can calculate the posterior probability of each state at each time frame given an input dialogue sequence, enabling us to extract top-N domain-specific sequences. We also need to find the appropriate topic number for the topic model. In our implementation, we used a tentative value of 100, which may not be appropriate. In addition, we believe the topic model and the CSHMM can be unified because these models are fundamentally similar, especially when LDA is employed. Model topologies may also have to be reconsidered. In our CSHMM with concatenated training, the states in domain-specific SHMMs are only connected to the common states, which may be inappropriate because there could be a case where a domain changes from one to another without having a common sequence. Applying CSHMMs to speech and other NLP tasks is another challenge. As a near-term goal, we aim to apply our method to the summarization of meetings, where we will need to extend our CSHMMs to deal with more than two participants. Finally, we also want to build a contact center dialogue agent by extending the CSHMMs to partially observable Markov decision processes (POMDPs) (Williams and Young, 2007) by following the recent work on building POMDPs from dialogue data in the dynamic Bayesian network (DBN) framework (Minami et al., 2009).

Acknowledgments

We thank the members of the Spoken Dialog System Group, especially Noboru Miyazaki and Satoshi Kobashikawa, for their effort in dialogue data collection.

References

Barzilay, Regina and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 113–120.

Byrd, Roy J., Mary S. Neff, Wilfried Teiken, Youngja Park, Keh-Shin F. Cheng, Stephen C. Gates, and Karthik Visweswariah. 2008. Semi-automated logging of contact center telephone calls. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM)*, pages 133–142.

Chu-Carroll, Jennifer and Bob Carpenter. 1999. Vector-based natural language call routing. *Computational Linguistics*, 25(3):361–388.

Gillick, Dan and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18.

Gorin, Allen L., Giuseppe Riccardi, and Jerry H. Wright. 1997. How may I help you? *Speech Communication*, 23(1-2):113–127.

Hori, Chiori and Sadaoki Furui. 2003. A new approach to automatic speech summarization. *IEEE Transactions on Multimedia*, 5(3):368–378.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 68–73.

Lee, Kai-Fu. 1989. *Automatic speech recognition: the development of the SPHINX system*. Kluwer Academic Publishers.

Lin, Chin-Yew and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 71–78.

Liu, Feifan and Yang Liu. 2008. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (HLT)*, pages 201–204.

Mani, Inderjeet. 2001. *Automatic summarization*. John Benjamins Publishing Company.

Meguro, Toyomi, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami, and Hideki Isozaki. 2009. Analysis of listening-oriented dialogue for building listening agents. In *Proceedings of the SIGDIAL 2009 conference*, pages 124–127.

Minami, Yasuhiro, Akira Mori, Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, and Eisaku Maeda. 2009. Dialogue control algorithm for ambient intelligence based on partially observable Markov decision processes. In *Proceedings of the 1st international workshop on spoken dialogue systems technology (IWSDS)*, pages 254–263.

Murray, Gabriel, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 593–596.

Osborne, Miles. 2002. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 1–8.

Rabiner, Lawrence R. and Biing-Hwang Juang. 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16.

Rabiner, Lawrence R. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296.

Radev, Dragomir R., Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Reynolds, Douglas A., Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19 – 41.

Subramaniam, L. Venkata, Tanveer A. Faruquie, Shajith Ikbal, Shantanu Godbole, and Mukesh K. Mohania. 2009. Business intelligence from voice of customer. In *Proceedings of the 2009 IEEE International Conference on Data Engineering (ICDE)*, pages 1391–1402.

Takeuchi, Hironori, L Venkata Subramaniam, Tetsuya Nasukawa, Shourya Roy, and Sreeram Balakrishnan. 2007. A conversation-mining system for gathering insights to improve agent productivity. In *Proceedings of the IEEE International Conference on E-Commerce Technology and the IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, pages 465–468.

Tam, Yik-Cheung and Tanja Schultz. 2005. Dynamic language model adaptation using variational Bayes inference. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 5–8.

Šingliar, Tomas and Milos Hauskrecht. 2006. Noisy-OR component analysis and its application to link analysis. *The Journal of Machine Learning Research*, 7:2189–2213.

Williams, Jason D. and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.