IMPROVING HMM-BASED EXTRACTIVE SUMMARIZATION FOR MULTI-DOMAIN CONTACT CENTER DIALOGUES

Ryuichiro Higashinaka[†], *Yasuhiro Minami*[‡], *Hitoshi Nishikawa*[†], *Kohji Dohsaka*[‡],
Toyomi Meguro[‡], *Satoshi Kobashikawa*[†], *Hirokazu Masataki*[†],
Osamu Yoshioka[†], *Satoshi Takahashi*[†], *and Genichiro Kikui*[†]

† NTT Cyber Space Laboratories, NTT Corporation
‡NTT Communication Science Laboratories, NTT Corporation

ABSTRACT

This paper reports the improvements we made to our previously proposed hidden Markov model (HMM) based summarization method for multi-domain contact center dialogues. Since the method relied on Viterbi decoding for selecting utterances to include in a summary, it had the inability to control compression rates. We enhance our method by using the forward-backward algorithm together with integer linear programming (ILP) to enable the control of compression rates, realizing summaries that contain as many domain-related utterances and as many important words as possible within a predefined character length. Using call transcripts as input, we verify the effectiveness of our enhancement.

Index Terms— Natural languages, Natural language interfaces, Hidden Markov models

1. INTRODUCTION

Automatically summarizing calls is one of the important goals of contact centers because it allows human operators to look back on their exchanges with customers to improve service skills. It also allows supervisors to check the activity of individual operators for finding ways to improve customer satisfaction. Figure 1 shows a sample call (transcript) in the financial domain. Since such a dialogue can be quite long, our aim is to create a short extractive summary, such as the one shown in Fig. 2. Although there has been an emergence of work that uses natural language processing techniques to improve the productivity of contact centers [1, 2], little work has focused on the summarization of calls. One exception is [3], which uses manually created heuristic rules to extract key utterances, such as those related to caller intentions or the acceptance/loss of orders, and then uses those utterances to create call logs.

In that approach, however, experts must create rules and must do so for each business domain, which is costly and problematic in scalability. As an alternative, we have been promoting an approach based on hidden Markov models (HMMs) for extractive summarization of multi-domain contact center dialogues. With this approach, domain-specific operator-caller exchanges are automatically modeled from multi-domain dialogue data in order to perform extractive summarization of calls [4]. In our method, to summarize a dialogue of a given domain, we use Viterbi decoding [5] to select utterances that are most likely to be related to that domain in order to make a summary.

Since we used Viterbi decoding to extract all domain-related utterances, a shortcoming of our method was its inability to control compression rates, which has greatly limited the method's applicability. This paper tackles this problem by utilizing, instead of Viterbi decoding, the forward-backward algorithm [5] to assign utterances with posterior probabilities that indicate how likely they are to be related to a certain domain. Then, we regard such probabilities as the real-valued importance of utterances and formulate the summarization problem as the maximum coverage of important utterances and important words within a predefined length. Then, we solve the problem by using integer linear programming (ILP), which can find the optimal extractive summarization results under constraints [6].

In Section 2, we briefly describe our previously proposed HMM-based summarization method and explain the improvements we made to enable the control of compression rates in detail. In Section 3, we describe the dialogue data for training our HMMs as well as the reference data we created for testing. In Section 4, we describe the summarization experiment we performed and present the results. In Section 5, we conclude the paper.

2. HMM-BASED EXTRACTIVE SUMMARIZATION

2.1. Previous Method

Our previously proposed HMM-based extractive summarization method [4] used HMMs to model domain-specific sequences from multi-domain dialogue data. Although a method for HMM-based extractive summarization [7] had been reported before ours, it can only be applied to a single domain and training data have to be created for each domain.

978-1-4244-7903-0/10/$26.00 ©2010 IEEE

OPE: Thank you for waiting. This is Sakura at Wakaba Life Tokyo contact center.
CAL: Excuse me.
CAL: It's about my life insurance contract.
CAL: I'd like to change my plan.
OPE: Certainly.

⋮

OPE: From next month, the payment will be 25000 yen per month. Is that okay with you?
CAL: Yes.
OPE: Right.
OPE: The new contract will come into effect from next month.
CAL: From next month?
OPE: Is that all right?
CAL: Yes, that's fine.
OPE: Okay.
OPE: Okay then. Thank you for calling.
CAL: Thank you.

Fig. 1. Sample dialogue in the financial domain. OPE and CAL denote operator and caller, respectively. There are 70 utterances (880 characters) in this transcript when unabridged. The dialogue was originally in Japanese and was translated by the authors.

CAL: It's about my life insurance contract.
CAL: I'd like to change my plan.
OPE: Your current plan is the "Iki-Iki (lively) Life EX standard plan", is this right?
OPE: Sure. Then, could you tell me the plan you would like to change to?
CAL: I'd like to change to the "Iki-Iki (lively) Life EX premium plan".
OPE: Certainly.
OPE: The new contract will come into effect from next month.
CAL: Okay.
OPE: Okay then. Thank you for calling.

Fig. 2. Reference summary for the dialogue in Fig. 1.

In contrast, our method only requires dialogue data with domain labels and the training is done simultaneously for all domains.

Given the dialogue data of multiple domains, we first model utterance sequences of each domain using an HMM and combine such HMMs into a single HMM. Here, we assume that each HMM has a certain number of states that emit utterances of an operator and the same number of states for those of a caller. These states are connected ergodically and the training is done using the EM-algorithm. We use this type of HMM because it has been shown to be effective for modeling two-party conversations [8]. In combining the HMMs into a single HMM, we had three topologic variations:

ergodic: This variation connects all states in the HMMs ergodically with equal initial/transition probabilities (see Fig. 3 for the topology). In this paper, connecting HMMs means connecting all states in the HMMs.

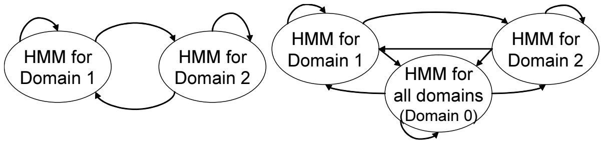

Fig. 3. Topologies of ergodic (left) and ergodic-cs (right).

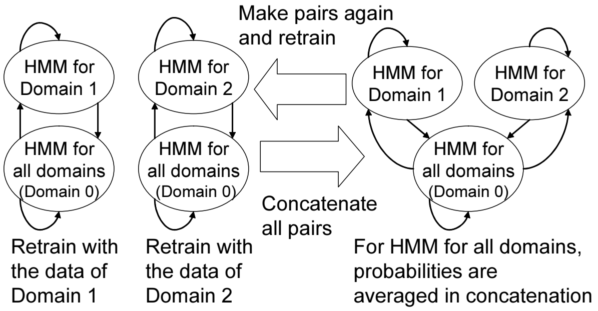
Make pairs again and retrain
HMM for Domain 1
HMM for Domain 2
HMM for all domains (Domain 0)
HMM for all domains (Domain 0)
HMM for Domain 1
HMM for Domain 2
HMM for all domains (Domain 0)
Concatenate all pairs
Retrain with the data of Domain 1
Retrain with the data of Domain 2
For HMM for all domains, probabilities are averaged in concatenation

Fig. 4. Process of concatenated training. For simplicity, the number of domains is depicted as two.

ergodic with common states (ergodic-cs): This variation bears similarity to the background model [9]. Here, in addition to 'ergodic', it additionally has an HMM trained from the utterances of all domains (see Fig. 3 for the topology). Hence, this additional HMM can represent common utterance sequences across all domains and makes it possible for the HMM to distinguish domain-specific sequences from common ones in the case of decoding. Since this additional HMM represents no specific domain, we call it 'an HMM for domain 0' and call its states 'common states'. Similarly to 'ergodic', all states in this HMM are connected to each other with equal initial/transition probabilities.

concat: This variation uses concatenated training [10] to optimize the transition probabilities between the states of different HMMs. Figure 4 illustrates its process. For K domains, an HMM for domain k ($1 \leq k \leq K$) is paired ergodically with a copy of an HMM for domain 0 and retrained with the data of domain k. This retraining reassigns the data, and, as a result, common utterance sequences become more likely to be generated from the HMM for domain 0 and domain-specific sequences in turn become more likely to be generated from the HMM for domain k. Then, the K paired HMMs are merged using the HMM for domain 0 as a pivot with the probabilities of the HMM for domain 0 averaged over K. If the fitting of all HMM pairs has not covered against training data, the HMMs are separated into pairs again and retrained. Here, the transition probabilities of the HMM for domain 0 is summed and redistributed between the pairs.

In our previous work [4], we trained these three types of HMMs using multi-domain contact center dialogues. Then, to summarize a dialogue known to belong to domain k, we used Viterbi decoding to decode an input utterance sequence into domain labels that indicate to which domain each utterance is most likely to be related. Finally, we selected all utterances in the dialogue that were given domain k labels. Since we selected all utterances having a certain domain label, a shortcoming of our previous method was its inability to control the compression rates. Note that, although concat achieved the best results in [4], we reconsider all three variations in this paper because we use a different decoding algorithm and different references (see Section 3).

2.2. Enabling to Control Compression Rates

One simple solution to enabling compression-rate control is to apply standard extractive summarization methods to the extracted utterances by Viterbi decoding. For example, using ILP [6], we can optimally select utterances that cover as many important words as possible. However, this may not be appropriate because the utterances not selected by the Viterbi decoding but that still contain important words would not be included in the summary. In addition, selecting utterances based only on its important words means that the importance of an utterance is determined solely by its words, which may be too simplistic because the importance of words is likely to depend on the context of the dialogue.

We argue that an utterance has certain real-valued importance (not a binary value as in Viterbi decoding) depending on the context and that the real-valued importance affects the importance of words in the utterance. For example, utterances concerning callers' requests and decisions should be valued more highly than others; therefore, the words in such utterances should be more important. For such importance of utterances, we propose to use the posterior probabilities of the states of the HMM for the target domain. The posterior probabilities can be calculated by using the forward-backward algorithm [5], and they represent how likely the utterances may have been generated from the HMM of the domain in question. We can regard the posterior probabilities as the importance of utterances because high-probability utterances should represent key utterances characterizing the domain.

Following this idea, we formulate the summarization problem as the maximum coverage of important utterances and important words. Although our formulation is similar to that in [11], which uses position information for weighting the words, our formulation is different in that we take into account the flow of dialogue that can be learned from data.

In our formulation, we perform the maximization using the following objective function:

$$\text{argmax} \sum_i \sum_j m_{ij} w_{ij} z_{ij} \tag{1}$$

where m_{ij} is a binary value representing whether the i-th utterance contains word j, w_{ij} is the weight of word j in the i-th utterance, and z_{ij} is a binary value representing whether word j in the i-th utterance is included in the summary. Here, w_{ij} is derived by

$$w_{ij} = \text{weight}(U_i) \cdot \text{weight}(w_j) \tag{2}$$

where 'weight' is a function that returns the importance of its argument, U_i is the i-th utterance, and w_j is the j-th word in the entire vocabulary within the dialogue. Here, for the weight of U_i, we use the posterior probabilities. For the weight of w_j, we use its term frequency within the dialogue, or as we show later, we can also employ an external dictionary to create a custom weight function. In maximizing the objective function, we have four constraints. First, we have

$$x_i, z_{ij} \in \{0, 1\} \ \ (\forall i, j) \tag{3}$$

which constrains the range of decision variables. Here, x_i is a binary value representing whether the i-th utterance is included in the summary. It is used in the second constraint

$$\sum_i l_i x_i \leq L \tag{4}$$

which limits the character length of the summary. Here, l_i is the character length of the i-th utterance and L is the length limitation. Our third constraint is

$$x_i \geq z_{ij} \ \ (\forall i, j) \tag{5}$$

which represents the inclusive relationship between utterances and the words in them; that is, if the i-th utterance is not included in the summary, words in the i-th utterance must not be included in the summary. Finally, to reduce the redundancy of the output summary, we have

$$\sum_i m_{ij} z_{ij} \leq 1 \ \ (\forall j) \tag{6}$$

which means that if the same words are included in the summary more than once, only one of their weights (i.e., the highest one for maximization) is used in the objective function.

2.3. Summarization Procedure

Using our three types of HMMs and our ILP formulation, our HMM-based extractive summarization method is performed as follows. See Fig. 5 for an illustration of the summarization flow.

Let D ($d_1 \ldots d_N$) be the entire set of contact center dialogues, DM^k ($DM^k \in DM, 1 \leq k \leq K$) be the domain assigned to domain k, and $U_{d_i,1} \ldots U_{d_i,H}$ be the utterances in d_i. Here, H is the number of utterances in d_i. To facilitate the treatment of utterances by HMMs, we assume that the utterances have been converted into topic labels $T_{d_i,1} \ldots T_{d_i,H}$ in advance. See [4] for how we do this by means of latent Dirichlet allocation (LDA). Basically, the conversion is the same as that used in [7], in which 'content topics' are assigned

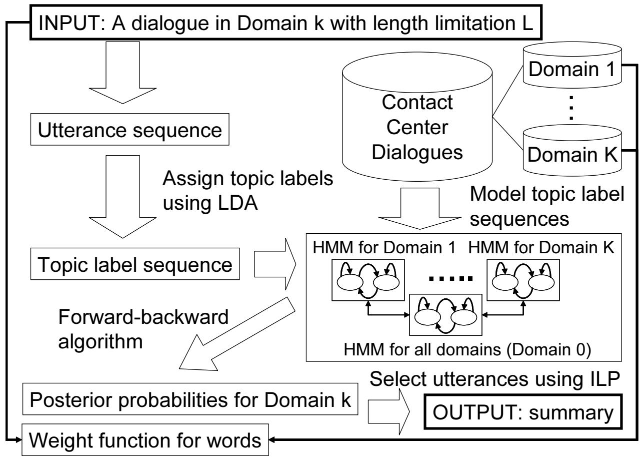
INPUT: A dialogue in Domain k with length limitation L
Utterance sequence
Contact Center Dialogues
Domain 1
:
Domain K
Assign topic labels using LDA
Model topic label sequences
Topic label sequence
HMM for Domain 1
HMM for Domain K
.....
HMM for all domains (Domain 0)
Forward-backward algorithm
Select utterances using ILP
Posterior probabilities for Domain k
OUTPUT: summary
Weight function for words

Fig. 5. Summarization flow.

to newswire sentences. From D, we train the HMMs to model the sequences of topic labels.

Let d_j ($\notin D$) be the input dialogue, $DM(d_j)$ ($\in DM$) be the table for obtaining the domain label of d_j, L the length limitation, and $U_{d_j,1} \ldots U_{d_j,H_{d_j}}$ be the utterances in d_j, where H_{d_j} is the number of the utterances. After converting the utterances into topic labels $T_{d_j,1} \ldots T_{d_j,H_{d_j}}$, we perform the forward-backward algorithm to derive the posterior probabilities $P_{d_j,1} \ldots P_{d_j,H_{d_j}}$ for $DM(d_j)$. At the same time, we instantiate the weight function for words, which returns the term frequency (TF) for a content word in d_j. Alternatively, we can use the following weight function for words:

$$\text{weight}(w_{i,DM^k}) = \frac{\log(\text{P}(w_i|DM^k))}{\log(\text{P}(w_i|DM \backslash DM^k))} \tag{7}$$

where $\text{P}(w_i|DM^k)$ denotes the occurrence probability of w_i in the dialogues of DM^k, and $\text{P}(w_i|DM \backslash DM^k)$ denotes the occurrence probability of w_i in all domains except DM^k. This log likelihood ratio estimates how much a word is characteristic of a given domain and therefore makes it possible to more accurately estimate the importance of words than the TF. Finally, using the posterior probabilities and the weight function, we select the utterances in d_j within L characters that maximize our objective function [cf. eq. (1)].

3. DIALOGUE DATA AND REFERENCES

We collected simulated contact center dialogues conducted between novice users (as callers) and expert operators who had experience of working at contact centers [4]. They talked over telephones in separate rooms. The operators were given manuals and the users were given realistic scenarios that showed why they were calling and what they needed. Each scenario had a description of approximately 300 characters. The dialogues ranged over six business domains: Finance (FIN), Internet Service Provider (ISP), Local Government Unit (LGU), Mail Order (MO), PC support (PC), and Telecommunication (TEL).

Table 1. Training data statistics.

	# dialogues	utts/dialogue	chars/dialogue
FIN	59	148.42	1868.37
ISP	64	108.27	1535.14
LGU	76	108.83	1179.75
MO	70	125.13	1438.33
PC	56	167.14	1891.20
TEL	66	139.55	1489.58
ALL	391	131.17	1542.32

We collected 391 and 307 dialogues in two separate occasions. All dialogues have been manually transcribed. Table 1 shows the statistics for the initial 391 dialogues. It can be seen that dialogues can be quite long with over 100 utterances and 1000-2000 characters per dialogue; hence the need for summarization. Reserving this initial set for training our HMMs, we created reference data using the remaining 307 dialogues, which had statistics similar to the training data. We randomly sampled 40 dialogues for each domain (240 dialogues in total) and had a text analyst make 250- and 500-character summaries by selecting utterances from the transcripts. See Fig. 2 for a sample reference.

The analyst was given the instruction: "Assume you are the operator and make a summary by selecting utterances so that both you and your supervisor would be able to easily grasp what took place in a dialogue". She was also instructed to choose as many utterances as necessary within a given length, although she was also allowed not to select utterances forcefully as long as the total summary length exceeded 80% of the target length. It took appropriately two weeks to create the reference summaries.

To investigate the inter-annotator agreement, we also had a second analyst create summaries for half the test dialogues (20 dialogues for each domain; 120 dialogues in total). The inter-annotator agreement that the two analysts chose the same utterance in Cohen's κ was 0.43 and 0.53 for 250- and 500-character summaries, respectively. Although we acknowledge that the agreement is not very high and we need more concrete instructions for making summaries, the kappa value of over 0.4 indicates that we have moderate agreement. We use the 240 summaries created by the first analyst as our reference data in the following experiment.

Note that in [4], we used the scenarios as references. Using the F-measure (the harmonic mean of the precision and recall), we performed the evaluation by assessing how accurately the output summaries contained the content words in the scenarios. We used scenarios as references under the assumption that they should contain the basic content exchanged between an operator and a caller. However, strictly speaking, this assumption may not hold. Therefore, in this work, we decided to prepare references manually and perform the most straightforward evaluation; that is, we calculate how

Table 2. The results in F-measure for 250-character summaries averaged over the 240 test dialogues. The letters next to F-measure values indicate statistical significance over others by a parametric paired t-test; 'aa' denotes statistical significance over (a) with $p < 0.01$ and 'a' with $p < 0.05$, and likewise for 'b' through 'f'. The best value is indicated by **bold** font.

	(a) even-TF	(b) viterbi-TF	(c) prob-TF	(d) even-DD	(e) viterbi-DD	(f) prob-DD
ergodic	0.190	0.236^{aa}	0.232^{aa}	0.249^{aac}	0.259^{aabbcc}	$\mathbf{0.260}^{aabbccd}$
ergodic-cs	0.190	0.241^{aa}	0.236^{aa}	0.249^{aa}	$\mathbf{0.268}^{aabbccdd}$	$0.265^{aabbccdd}$
concat	0.190	0.237^{aa}	0.258^{aabb}	0.249^{aa}	0.262^{aabbd}	$\mathbf{0.271}^{aabbcdd}$

Table 3. The results in F-measure for 500-character summaries. See the caption of Table 2 for the notations.

	(a) even-TF	(b) viterbi-TF	(c) prob-TF	(d) even-DD	(e) viterbi-DD	(f) prob-DD
ergodic	0.379	0.405^{aa}	0.406^{aa}	$0.423^{aabbcce}$	0.414^{aab}	$\mathbf{0.430}^{aabbccee}$
ergodic-cs	0.379	0.411^{aa}	0.412^{aa}	0.423^{aabce}	0.414^{aa}	$\mathbf{0.435}^{aabbccddee}$
concat	0.379	0.400^{aa}	$\mathbf{0.437}^{aabbdee}$	0.423^{aabbee}	0.409^{aa}	$0.433^{aabbdee}$

accurately correct utterances are selected from dialogues by using F-measure.

4. EXPERIMENT

We performed an experiment to verify our summarization method. Using the transcripts of 391 dialogues (cf. Table 1), we trained three types of HMMs; namely, ergodic, ergodic-cs, and concat. Here, the HMM for each domain has two states (one state each for an operator and a caller), and, for ergodic-cs and concat, we set the number of common states to six (three states each for an operator and a caller), which led to good performance in [4]. The number of topics for LDA was 100.

For comparison, we also prepared three variations for assigning the weights for utterances. They are '**even**', '**viterbi**', and '**prob**'. Here, 'even' gives equal weights to all utterances and simulates the case when we apply a standard ILP-based summarization to all utterances in a dialogue. The second variation, 'viterbi', gives 1.0 to the utterances selected by Viterbi decoding and a very small flooring value (1.0^{-6}) to unselected ones. This simulates our previous work [4] based on Viterbi decoding. A flooring value was added here to avoid having zero weights, which can result in too short summaries. Our third variation, 'prob', is our newly proposed one and uses the posterior probabilities for the weights of utterances. Here, zero probabilities are also floored. In addition, we have two variations for assigning the weights for words; namely, one that uses the TF and another that uses the weight function of eq. (7). We call the latter variation the domain dictionary (DD). As a result, we have six variations of weight functions: (a) even-TF, (b) viterbi-TF, (c) prob-TF, (d) even-DD, (e) viterbi-DD, and (f) prob-DD.

4.1. Results

Tables 2 and 3 show the summarization results for the 240 test dialogues (transcripts) in F-measure for 250- and 500-character summaries, respectively, for all variations. It can be seen that the variations that use the posterior probabilities (i.e., prob and prob-DD) outperform others in most cases, especially for 500-character summaries, although there is no statistical significance between viterbi/prob variations for 250-character summaries. This is reasonable when considering that only a small number of strongly domain-related utterances can be included in such short summaries and that the utterances given high weights by the forward-backward algorithm are also selected by Viterbi decoding. When we focus on how the weights of words affect the results, we clearly see that the incorporation of the domain-dependent weight function is effective. This fact also confirms our assumption that having domain-related elements makes good summaries. Overall, our new summarization method (prob/prob-DD) produced good summaries with different compression rates and that using posterior probabilities made solid improvements.

4.2. Impact of HMM Topologies

When we look into the impact of HMM topologies, we see a rather interesting phenomenon: when the common states are introduced (i.e., ergodic-cs), the performance goes up for both 'viterbi' and 'prob' variations. However, when concatenated training is applied, the performance for viterbi and viterbi-DD drops, whereas that for prob and prob-DD generally improves.

To investigate why this is, we made a breakdown of correctly selected utterances categorized by which HMM is responsible for selecting them in Viterbi decoding (see Fig. 4). We found that, in concat, many correct utterances were generated from domain 0 compared to ergodic-cs. We consider that because too many utterances were labeled as domain 0 (i.e., having only the flooring values), concat could only create its summaries using a limited number of domain-related utterances, leading to its degraded performance. More utterances had to be considered for inclusion. Following this result, we reviewed our reference summaries and found that there are a fair number of utterances that look common across domains, such as 'yes' and operators' thanking and acknowledging;

Table 4. Breakdown (in ratios) of correctly selected utterances categorized by which HMM (HMM for domain 0, HMM for the target domain, or the HMMs for other domains) is estimated to have generated them in Viterbi decoding.

250-character summary			
	domain 0	target-domain	other-domain
ergodic-cs	0.384	0.411	0.205
concat	0.491	0.372	0.137
500-character summary			
	domain 0	target-domain	other-domain
ergodic-cs	0.422	0.377	0.200
concat	0.522	0.343	0.135

Table 5. Ratio of utterances not selected by Viterbi decoding and whose posterior probabilities remained 0 or became more than 0 when the forward-backward algorithm is used.

	remained 0	became more than 0
ergodic	0.000	1.000
ergodic-cs	0.000	1.000
concat	0.486	0.514

however, such utterances were nonetheless selected by the analyst because they appear close to important utterances, such as requests and decisions. Since concat is trained so as to discriminate common utterance sequences from domain-specific sequences as much as possible, when we use Viterbi decoding, such common-looking utterances were forcefully categorized as common (domain 0) and were not selected. In contrast, prob and prob-DD performed well with concat because such utterances were given certain (i.e., more than the flooring value) weights because they are common but not totally so. The combination of prob and concat can perform such delicate utterance selection, leading to its good performance.

Table 5 shows the ratio of utterances that are not selected by Viterbi decoding and whose posterior probabilities remained 0 or became more than 0 by using the forward-backward algorithm. The table indicates that, for ergodic and ergodic-cs, the forward-backward algorithm gave some posterior probabilities for all utterances. This is reasonable because the HMM for domain 0 was trained using the data of all domains. However, for concat, about a half of the utterances still had zero posterior probabilities, which indicates that concat is successfully distinguishing totally common utterances from less common ones, enabling it to successfully create summaries when used with prob and prob-DD.

5. CONCLUDING REMARKS

This paper reported improvements we made to our previously proposed HMM-based summarization method for multi-domain contact center dialogues. For the control of compression rates, we formulated the summarization problem as the maximum coverage of important utterances and important words and used the forward-backward algorithm to derive posterior probabilities that represent how likely utterances in a dialogue belong to a particular domain and used those probabilities to represent the importance of utterances. Experimental results showed that our revised method (prob/prob-DD) successfully creates summaries of different compression rates and outperforms existing methods, including our previous one. The improvement was significant for 500-character summaries. By analyzing the trained models, we also found supporting evidence that concatenated training successfully models common utterance sequences. Our future work includes improving the utterance selection accuracy and applying our method to speech recognition results.

6. REFERENCES

[1] Hironori Takeuchi, L Venkata Subramaniam, Tetsuya Nasukawa, Shourya Roy, and Sreeram Balakrishnan, “A conversation-mining system for gathering insights to improve agent productivity,” in *Proc. CEC-EEE*, 2007, pp. 465–468.

[2] L. Venkata Subramaniam, Tanveer A. Faruquie, Shajith Ikbal, Shantanu Godbole, and Mukesh K. Mohania, “Business intelligence from voice of customer,” in *Proc. ICDE*, 2009, pp. 1391–1402.

[3] Roy J. Byrd, Mary S. Neff, Wilfried Teiken, Youngja Park, Keh-Shin F. Cheng, Stephen C. Gates, and Karthik Visweswariah, “Semi-automated logging of contact center telephone calls,” in *Proc. CIKM*, 2008, pp. 133–142.

[4] Ryuichiro Higashinaka, Yasuhiro Minami, Hitoshi Nishikawa, Kohji Dohsaka, Toyomi Meguro, Satoshi Takahashi, and Genichiro Kikui, “Learning to model domain-specific utterance sequences for extractive summarization of contact center dialogues,” in *Proc. COLING*, 2010, vol. Poster, pp. 400–408.

[5] Lawrence R. Rabiner and Biing-Hwang Juang, “An introduction to hidden Markov models,” *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[6] Dan Gillick and Benoit Favre, “A scalable global model for summarization,” in *Proc. the Workshop on Integer Linear Programming for Natural Language Processing*, 2009, pp. 10–18.

[7] Regina Barzilay and Lillian Lee, “Catching the drift: Probabilistic content models, with applications to generation and summarization,” in *Proc. HLT-NAACL*, 2004, pp. 113–120.

[8] Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami, and Hideki Isozaki, “Analysis of listening-oriented dialogue for building listening agents,” in *Proc. SIGDIAL*, 2009, pp. 124–127.

[9] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19 – 41, 2000.

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

[11] Wen tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki, “Multi-document summarization by maximizing informative content-words,” in *Proc. IJCAI*, 2007, pp. 1776–1782.