Automatic Generation of Domain Models for Call Centers from Noisy Transcriptions

Shourya Roy and **L Venkata Subramaniam**
IBM Research
India Research Lab
IIT Delhi, Block-1
New Delhi 110016
India
rshourya,lvsubram@in.ibm.com

Abstract

Call centers handle customer queries from various domains such as computer sales and support, mobile phones, car rental, etc. Each such domain generally has a *domain model* which is essential to handle customer complaints. These models contain common problem categories, typical customer issues and their solutions, greeting styles. Currently these models are manually created over time. Towards this, we propose an unsupervised technique to generate domain models automatically from call transcriptions. We use a state of the art Automatic Speech Recognition system to transcribe the calls between agents and customers, which still results in high word error rates (40%) and show that even from these noisy transcriptions of calls we can automatically build a domain model. The domain model is comprised of primarily a *topic taxonomy* where every node is characterized by *topic(s)*, *typical Questions-Answers (Q&As)*, *typical actions* and *call statistics*. We show how such a domain model can be used for topic identification of unseen calls. We also propose applications for aiding agents while handling calls and for agent monitoring based on the domain model.

1 Introduction

Call center is a general term for help desks, information lines and customer service centers. Many companies today operate call centers to handle customer issues. It includes dialog-based (both voice and online chat) and email support a user receives from a professional agent. Call centers have become a central focus of most companies as they allow them to be in direct contact with their customers to solve product-related and services-related issues and also for grievance redress. A typical call center agent handles over a hundred calls in a day. Gigabytes of data is produced every day in the form of speech audio, speech transcripts, email, etc. This data is valuable for doing analysis at many levels, e.g., to obtain statistics about the type of problems and issues associated with different products and services. This data can also be used to evaluate agents and train them to improve their performance.

Today's call centers handle a wide variety of domains such as computer sales and support, mobile phones and apparels. To analyze the calls in any domain, analysts need to identify the key issues in the domain. Further, there may be variations within a domain, say mobile phones, based on the service providers. The analysts generate a *domain model* through inspection of the call records (audio, transcripts and emails). Such a model can include a listing of the call categories, types of problems solved in each category, listing of the customer issues, typical questions-answers, appropriate call opening and closing styles, etc. In essence, these models provide a structured view of the domain. Manually building such models for various domains may become prohibitively resource intensive. Another important point to note is that these models are *dynamic* in nature and change over time. As a new version of a mobile phone is introduced, software is launched in a country, a sudden attack of a virus, the model may need to be refined. Hence, an automated way of creating and maintaining such a model is important.

In this paper, we have tried to formalize the essential aspects of a domain model. It comprises of primarily a *topic taxonomy* where every node is characterized by *topic(s)*, *typical Questions-*

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 737–744,
Sydney, July 2006.

Answers (Q&As), *typical actions* and *call statistics*. To build the model, we first automatically transcribe the calls. Current automatic speech recognition technology for telephone calls have moderate to high word error rates (Padmanabhan et al., 2002). We applied various feature engineering techniques to combat the noise introduced by the speech recognition system and applied text clustering techniques to group topically similar calls together. Using clustering at different granularity and identifying the relationship between groups at different granularity we generate a taxonomy of call types. This taxonomy is augmented with various meta information related to each node as mentioned above. Such a model can be used for identification of topics of unseen calls. Towards this, we envision an *aiding tool* for agents to increase agent effectiveness and an administrative tool for agent appraisal and training.

Organization of the paper: We start by describing related work in relevant areas. Section 3 talks about the call center dataset and the speech recognition system used. The following section contains the definition and describes an unsupervised mechanism for building a topical model from automatically transcribed calls. Section 5 demonstrates the usability of such a topical model and proposes possible applications. Section 6 concludes the paper.

2 Background and Related Work

In this work, we are trying to bridge the gap between a few seemingly unrelated research areas viz. (1) Automatic Speech Recognition(ASR), (2) Text Clustering and Automatic Taxonomy Generation (ATG) and (3) Call Center Analytics. We present some relevant work done in each of these areas.

Automatic Speech Recognition(ASR): Automatic transcription of telephonic conversations is proven to be more difficult than the transcription of read speech. According to (Padmanabhan et al., 2002), word-error rates are in the range of 7-8% for read speech whereas for telephonic speech it is more than 30%. This degradation is due to the spontaneity of speech as well as the telephone channel. Most speech recognition systems perform well when trained for a particular accent (Lawson et al., 2003). However, with call centers now being located in different parts of the world, the requirement of handling different accents by the same speech recognition system further increases word error rates.

Automatic Taxonomy Generation (ATG): In recent years there has been some work relating to mining domain specific documents to build an ontology. Mostly these systems rely on parsing (both shallow and deep) to extract relationships between key concepts within the domain. The ontology is constructed from this by linking the extracted concepts and relations (Jiang and Tan, 2005). However, the documents contain well formed sentences which allow for parsers to be used.

Call Center Analytics: A lot of work on automatic call type classification for the purpose of categorizing calls (Tang et al., 2003), call routing (Kuo and Lee, 2003; Haffner et al., 2003), obtaining call log summaries (Douglas et al., 2005), agent assisting and monitoring (Mishne et al., 2005) has appeared in the past. In some cases, they have modeled these as *text classification* problems where topic labels are manually obtained (Tang et al., 2003) and used to put the calls into different buckets. Extraction of key phrases, which can be used as features, from the noisy transcribed calls is an important issue. For manually transcribed calls, which do not have any noise, in (Mishne et al., 2005) a phrase level significance estimate is obtained by combining word level estimates that were computed by comparing the frequency of a word in a domain-specific corpus to its frequency in an open-domain corpus. In (Wright et al., 1997) phrase level significance was obtained for noisy transcribed data where the phrases are clustered and combined into finite state machines. Other approaches use n-gram features with stop word removal and minimum support (Kuo and Lee, 2003; Douglas et al., 2005). In (Bechet et al., 2004) call center dialogs have been clustered to learn about dialog traces that are similar.

Our Contribution: In the call center scenario, the authors are not aware of any work that deals with automatically generating a taxonomy from transcribed calls. In this paper, we have tried to formalize the essential aspects of a domain model. We show an unsupervised method for building a domain model from noisy unlabeled data, which is available in abundance. This hierarchical domain model contains summarized topic specific details for topics of different granularity. We show how such a model can be used for topic identification of unseen calls. We propose two applications for

aiding agents while handling calls and for agent monitoring based on the domain model.

3 Issues with Call Center Data

We obtained telephonic conversation data collected from the internal IT help desk of a company. The calls correspond to users making specific queries regarding problems with computer software such as *Lotus Notes, Net Client, MS Office, MS Windows*, etc. Under these broad categories users faced specific problems e.g. in *Lotus Notes* users had problems with their *passwords, mail archiving, replication, installation*, etc. It is possible that many of the sub problem categories are similar, e.g. *password* issues can occur with *Lotus Notes, Net Client* and *MS Windows*.

We obtained automatic transcriptions of the dialogs using an Automatic Speech Recognition (ASR) system. The transcription server, used for transcribing the call center data, is an IBM research prototype. The speech recognition system was trained on 300 hours of data comprising of help desk calls sampled at 6KHz. The transcription output comprises information about the recognized words along with their durations, i.e., beginning and ending times of the words. Further, speaker turns are marked, so the agent and customer portions of speech are demarcated without exactly naming which part is the agent and which the customer. It should be noted that the call center agents and the customers were of different nationalities having varied accents and this further made the job of the speech recognizer hard. The resultant transcriptions have a word error rate of about 40%. This high error rate implies that many wrong deletions of actual words and wrong insertion of dictionary words have taken place. Also often speaker turns are not correctly identified and voice portions of both speakers are assigned to a single speaker. Apart from speech recognition errors there are other issues related to spontaneous speech recognition in the transcriptions. There are no punctuation marks, silence periods are marked but it is not possible to find sentence boundaries based on these. There are repeats, false starts, a lot of pause filling words such as *um* and *uh*, etc. Portion of a transcribed call is shown in figure 1. Generally, at these noise levels such data is hard to interpret by a human. We used over 2000 calls that have been automatically transcribed for our analysis. The average duration of a call is about 9 minutes. For 125 of these calls, *call topics* were manually assigned.

```
SPEAKER 1:  windows thanks for calling and you can
learn yes i don't mind it so then i went to
SPEAKER 2:  well and ok bring the machine front
end loaded with a standard um and that's um it's
a desktop machine and i did that everything was
working wonderfully um I went ahead connected
into my my network um so i i changed my network
settings to um to my home network so i i can you
know it's showing me for my workroom um and then
it is said it had to reboot in order for changes
to take effect so i rebooted and now it's asking
me for a password which i never i never said
anything up
SPEAKER 1:  ok just press the escape key i can
doesn't do anything can you pull up so that i mean
```

Figure 1: Partial transcript of a help desk dialog

4 Generation of Domain Model

Fig 2 shows the steps for generating a domain model in the call center scenario. This section explains different modules shown in the figure.

4.1 Description of Model

We propose the *Domain Model* to be comprised of primarily a *topic taxonomy* where every node is characterized by *topic(s), typical Questions-Answers (Q&As), typical actions* and *call statistics*. Generating such a taxonomy manually from scratch requires significant effort. Further, the changing nature of customer problems requires frequent changes to the taxonomy. In the next subsection, we show that meaningful taxonomies can be built without any manual supervision from a collection of noisy call transcriptions.

4.2 Taxonomy Generation

As mentioned in section 3, automatically transcribed data is noisy and requires a good amount of *feature engineering* before applying any text analytics technique. Each transcription is passed through a *Feature Engineering Component* to perform noise removal. We performed a sequence of cleansing operations to remove *stopwords* such as *the, of, seven, dot, january, hello*. We also remove *pause filling words* such as *um, uh, huh* . The remaining words in every transcription are passed through a *stemmer* (using Porter's stemming algo-

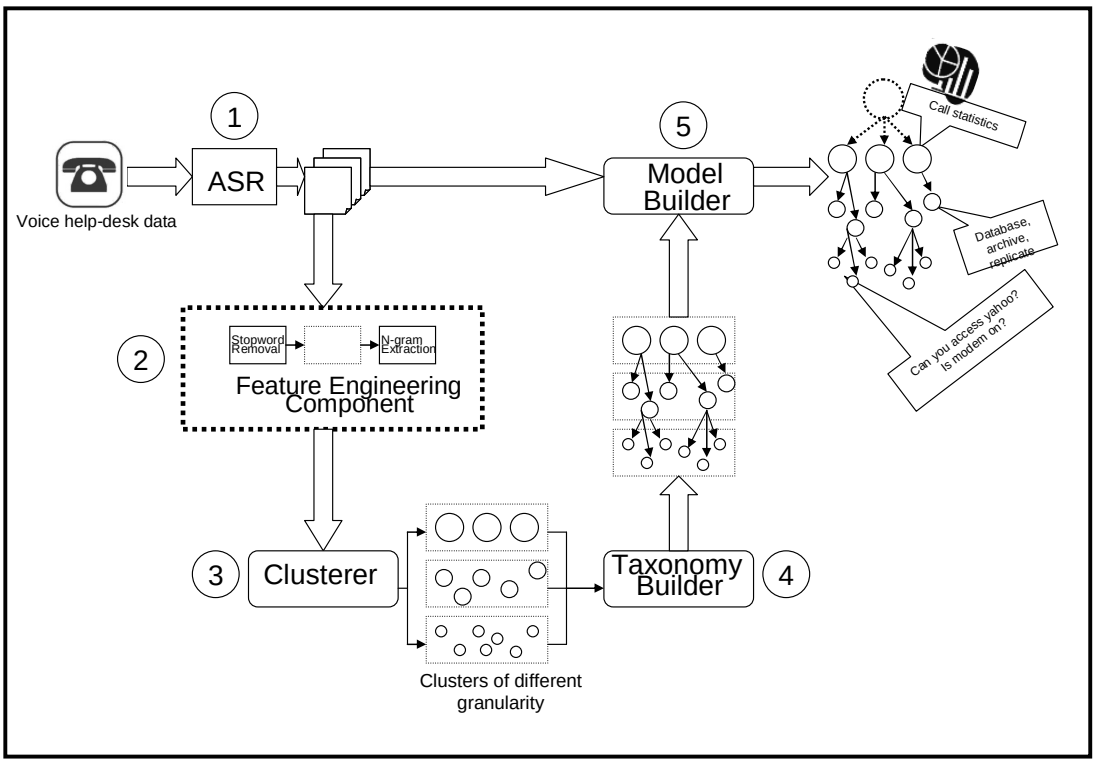

Figure 2: 5 Steps to automatically build domain model from a collection of telephonic conversation recordings

rithm [1]) to extract the root form of every word e.g. *call* from *called*. We extract all n-grams which occur more frequently than a threshold and do not contain any stopword. We observed that using all n-grams without thresholding deteriorates the quality of the generated taxonomy. *a t & t, lotus notes,* and *expense reimbursement* are some examples of extracted n-grams.

The *Clusterer* generates individual levels of the taxonomy by using text clustering. We used CLUTO package [2] for doing text clustering. We experimented with all the available clustering functions in CLUTO but no one clustering algorithm consistently outperformed others. Also, there was not much difference between various algorithms based on the available goodness metrics. Hence, we used the default *repeated bisection* technique with *cosine* function as the similarity metric. We ran this algorithm on a collection of 2000 transcriptions multiple times. First we generate 5 clusters from the 2000 transcriptions. Next we generate 10 clusters from the same set of transcriptions and so on. At the finest level we split them into 100 clusters. To generate the topic taxonomy, these sets containing 5 to 100 clusters are passed through the *Taxonomy Builder* component. This component (1) removes clusters containing less than n documents (2) introduces directed edges from cluster v_1 to v_2 if v_1 and v_2 share at least one document between them, and where v_2 is one level finer than v_1. Now v_1 and v_2 become nodes in adjacent layers in the taxonomy. Here we found the taxonomy to be a tree but in general it can be a DAG. Now onwards, each node in the taxonomy will be referred to as a *topic*.

This kind of top-down approach was preferred over a bottom-up approach because it not only gives the linkage between clusters of various granularity but also gives the most *descriptive* and *discriminative* set of features associated with each node. CLUTO defines descriptive (and discriminative) features as the set of features which contribute the most to the average similarity (dissimilarity) between documents belonging to the same cluster (different clusters). In general, there is a large overlap between descriptive and discriminative features. These features, *topic features*, are later used for generating topic specific information. Figure 3 shows a part of the taxonomy obtained from the IT help desk dataset. The labels

[1] http://www.tartarus.org/˜martin/PorterStemmer
[2] http://glaros.dtc.umn.edu/gkhome/views/cluto

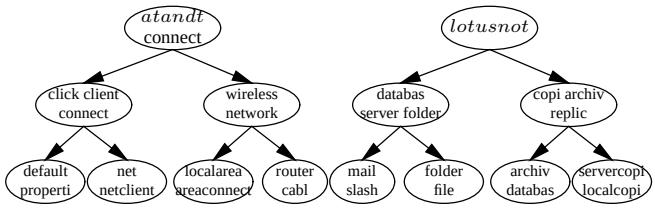

Figure 3: A part of the automatically generated ontology along with descriptive features.

shown in Figure 3 are the most descriptive and discriminative features of a node given the labels of its ancestors.

4.3 Topic Specific Information

The *Model Builder* component in Figure 2 creates an augmented taxonomy with topic specific information extracted from noisy transcriptions. Topic specific information includes phrases that describe *typical actions*, *typical Q&As* and *call statistics* (for each topic in the taxonomy).

Typical Actions: Actions correspond to typical issues raised by the customer, problems and strategies for solving them. We observed that action related phrases are mostly found around topic features. Hence, we start by searching and collecting all the phrases containing topic words from the documents belonging to the topic. We define a 10-word window around the topic features and harvest all phrases from the documents. The set of collected phrases are then searched for n-grams with support above a preset threshold. For example, both the 10-grams *note in click button to set up for all stops* and *to action settings and click the button to set up* increase the support count of the 5-gram *click button to set up*.

The search for the n-grams proceeds based on a threshold on a distance function that counts the insertions necessary to match the two phrases. For example *can you* is closer to *can* $<$ ... $>$ *you* than to *can* $<$ ... $><$ ... $>$ *you*. Longer n-grams are allowed a higher distance threshold than shorter n-grams. After this stage we extracted all the phrases that frequently occur within the cluster.

In the second step, *phrase tiling and ordering*, we prune and merge the extracted phrases and order them. Tiling constructs longer n-grams from sequences of overlapping shorter n-grams. We noted that the phrases have more meaning if they are ordered by their appearance. For example, if *go to the program menu* typically appears before *select options from program menu* then it is more useful to present them in the order of their appearance. We establish this order based on the average turn number where a phrase occurs.

```
thank you for calling this is
problem with our serial number software
Q: may i have your serial number
Q: how may i help you today
A: i'm having trouble with my at&t network
............
............
click on advance log in properties
i want you to right click
create a connection across an existing internet
connection
in d.  n.  s.  use default network
............
............
Q: would you like to have your ticket
A: ticket number is two
thank you for calling and have a great day
thank you for calling bye bye
anything else i can help you with
have a great day you too
```

Figure 4: Topic specific information

Typical Questions-Answers: To understand a customer's issue the agent needs to ask the right set of questions. Asking the right questions is the key to effective call handling. We search for all the questions within a topic by defining question templates. The question templates basically look for all phrases beginning with *how, what, can I, can you, were there*, etc. This set comprised of 127 such templates for questions. All 10-word phrases conforming to the question templates are collected and phrase harvesting, tiling and ordering is done on them as described above. For the answers we search for phrases in the vicinity immediately following the question.

Figure 4 shows a part of the topic specific information that has been generated for the *default properti* node in Fig 3. There are 123 documents in this node. We have selected phrases that occur at least 5 times in these 123 documents. We have captured the general opening and closing styles used by the agents in addition to typical actions and Q&As for the topic. In this node the documents pertain to queries on setting up a new A T & T network connection. Most of the topic specific issues that have been captured relate to the agent

leading the customer through the steps for setting up the connection. In the absence of tagged dataset we could not quantify our observation. However, when we compared the automatically generated topic specific information to the extracted information from the hand labeled calls, we noted that almost all the issues have been captured. In fact there are some issues in the automatically generated set that are missing from the hand labeled set. The following observations can be made from the topic specific information that has been generated:

- The phrases that have been captured turn out to be quite well formed. Even though the ASR system introduces a lot of noise, the resulting phrases when collected over the clusters are clean.
- Some phrases appear in multiple forms *thank you for calling how can i help you*, *how may i help you today*, *thanks for calling can i be of help today*. While tiling is able to merge matching phrases, semantically similar phrases are not merged.
- The list of topic specific phrases, as already noted, matched and at times was more exhaustive than similar hand generated sets.

Call Statistics: We compute various aggregate statistics for each node in the topic taxonomy as part of the model viz. (1) *average call duration*(in seconds), (2) *average transcription length*(number of words) (3) *average number of speaker turns* and (4) *number of calls*. We observed that call durations and number of speaker turns varies significantly from one topic to another. Figure 5 shows average call duration and corresponding average transcription lengths for a few interesting topics. It can be seen that in topic *cluster-1*, which is about *expense reimbursement* and related stuff, most of the queries can be answered quickly in standard ways. However, some *connection* related issues (topic *cluster-5*) require more information from customers and are generally longer in duration. Interestingly, topic *cluster-2* and topic *cluster-4* have similar average call durations but quite different average transcription lengths. On investigation we found that *cluster-4* is primarily about *printer* related queries where the customer many a times is not ready with details like printer name, ip address of the printer, resulting in long hold time whereas for *cluster-2*, which is about *online courses*, users

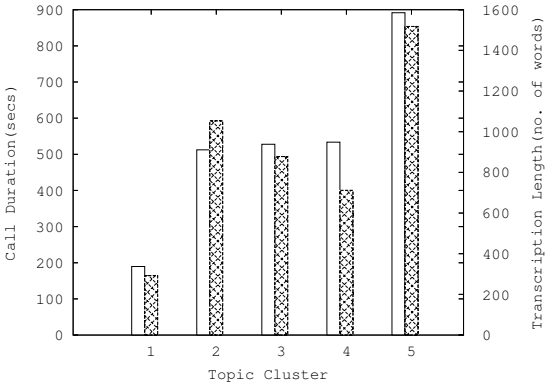

Figure 5: Call duration and transcription length for some topic clusters

generally have details like course name, etc. ready with them and are interactive in nature.

We build a hierarchical index of type {*topic*→*information*} based on this automatically generated model for each topic in the *topic taxonomy*. An entry of this index contains topic specific information viz. (1) *typical Q&As*, (2) *typical actions*, and (3) *call statistics*. As we go down this hierarchical index the information associated with each topic becomes more and more specific. In (Mishne et al., 2005) a manually developed collection of issues and their solutions is indexed so that they can be matched to the call topic. In our work the indexed collection is automatically obtained from the call transcriptions. Also, our index is more useful because of its hierarchical nature where information can be obtained for topics of various granularity unlike (Mishne et al., 2005) where there is no concept of topics at all.

5 Application of Domain Model

Information retrieval from spoken dialog data is an important requirement for call centers. Call centers constantly endeavor to improve the call handling efficiency and identify key problem areas. The described model provides a comprehensive and structured view of the domain that can be used to do both. It encodes three levels of information about the domain:

- **General:** The taxonomy along with the labels gives a general view of the domain. The general information can be used to monitor trends on how the number of calls in different categories change over time e.g. daily, weekly, monthly.

- **Topic level:** This includes a listing of the specific issues related to the topic, typical customer questions and problems, usual strategies for solving the problems, average call durations, etc. It can be used to identify primary issues, problems and solutions pertaining to any category.

- **Dialog level:** This includes information on how agents typically open and close calls, ask questions and guide customers, average number of speaker turns, etc. The dialog level information can be used to monitor whether agents are using courteous language in their calls, whether they ask pertinent questions, etc.

The {*topic→information*} index requires identification of the topic for each call to make use of information available in the model. Below we show examples of the use of the model for topic identification.

5.1 Topic Identification

Many of the customer complaints can be categorized into coarse as well as fine topic categories by listening to only the initial part of the call. Exploiting this observation we do *fast topic identification* using a simple technique based on distribution of topic specific *descriptive* and *discriminative* features (Sec 4.2) within the initial portion of the call. Figure 6 shows variation in prediction accuracy using this technique as a function of the fraction of a call observed for 5, 10 and 25 clusters verified over the 125 hand-labeled transcriptions. It can be seen, at coarse level, nearly 70% prediction accuracy can be achieved by listening to the initial 30% of the call and more than 80% of the calls can be correctly categorized by listening only to the first half of the call. Also calls related to some categories can be quickly detected compared to some other clusters as shown in Figure 7.

5.2 Aiding and Administrative Tool

Using the techniques presented in this paper so far it is possible to put together many applications for a call center. In this section we give some example applications and describe ways in which they can be implemented. Based on the hierarchical model described in Section 4 and topic identification mentioned in the last sub-section we describe

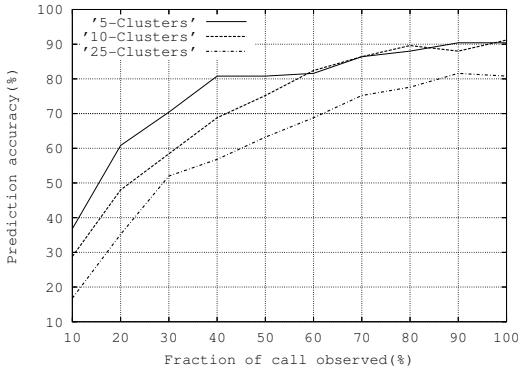

Figure 6: Variation in prediction accuracy with fraction of call observed for 5, 10 and 25 clusters

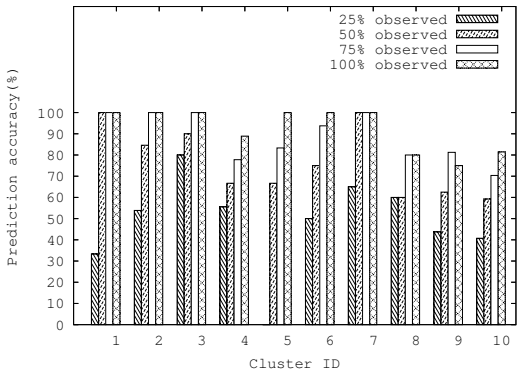

Figure 7: Cluster wise variation in prediction accuracy for 10 clusters

(1) a tool capable of aiding agents for efficient handling of calls to improve customer satisfaction as well as to reduce call handling time, (2) an administrative tool for agent appraisal and training.

Agent aiding is done based on the automatically generated domain model. The hierarchical nature of the model helps to provide generic to specific information to the agent as the call progresses. During call handling the agent can be provided the automatically generated taxonomy and the agent can get relevant information associated with different nodes by say clicking on the nodes. For example, once the agent identifies a call to be about {*lotusnot*} in Fig 3 then he can see the generic *Lotus Notes* related Q&As and actions. By interacting further with the customer the agent identifies it to be of {*copi archiv replic*} topic and typical Q&As and actions change accordingly. Finally, the agent narrows down to the topic as {*servercopi localcopi*} and suggest solution for replication problem in *Lotus Notes*.

The concept of **administrative tool** is primarily driven by Dialog and Topic level information. We envision this post-processing tool to be used

for comparing completed individual calls with corresponding topics based on the distribution of Q&As, actions and call statistics. Based on the topic level information we can check whether the agent identified the issues and offered the known solutions on a given topic. We can use the dialog level information to check whether the agent used courteous opening and closing sentences. Calls that deviate from the topic specific distributions, can be identified in this way and agents handling these calls can be offered further training on the *subject matter*, *courtesy*, etc. This kind of post-processing tool can also help us to catch *abnormally long calls*, *agents with high average call handle time*, etc.

6 Discussion and Future Work

We have shown that it is possible to retrieve useful information from noisy transcriptions of call center voice conversations. We have shown that the extracted information can be put in the form of a model that succinctly captures the domain and provides a comprehensive view of it. We briefly showed through experiments that this model is an accurate description of the domain. We have also suggested useful scenarios where the model can be used to aid and improve call center performance.

A call center handles several hundred-thousand calls per year in various domains. It is very difficult to monitor the performance based on manual processing of the calls. The framework presented in this paper, allows a large part of this work to be automated. A domain specific model that is automatically learnt and updated based on the voice conversations allows the call center to identify problem areas quickly and allocate resources more effectively.

In future we would like to semantically cluster the topic specific information so that redundant topics are eliminated from the list. We can use Automatic Taxonomy Generation(ATG) algorithms for document summarization (Kummamuru et al., 2004) to build topic taxonomies. We would also like to link our model to technical manuals, catalogs, etc. already available on the different topics in the given domain.

Acknowledgements: We thank our colleagues Raghuram Krishnapuram and Sreeram Balakrishnan for helpful discussions. We also thank Olivier Siohan from the IBM T. J. Watson Research Center for providing us with call transcriptions.

References

F. Bechet, G. Riccardi and D. Hakkani-Tur 2004. Mining Spoken Dialogue Corpora for System Evaluation and Modeling. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. July, Barcelona, Spain.

S. Douglas, D. Agarwal, T. Alonso, R. M. Bell, M. Gilbert, D. F. Swayne and C. Volinsky. 2005. Mining Customer Care Dialogs for "Daily News". *IEEE Trans. on Speech and Audio Processing*, 13(5):652–660.

P. Haffner, G. Tur and J. H. Wright 2003. Optimizing SVMs for Complex Call Classification. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. April 6-10, Hong Kong.

X. Jiang and A.-H. Tan. 2005. Mining Ontological Knowledge from Domain-Specific Text Documents. *IEEE International Conference on Data Mining*, November 26-30, New Orleans, Louisiana, USA.

K. Kummamuru, R. Lotlikar, S. Roy, K. Singal and R. Krishnapuram. 2004. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. *International Conference on World Wide Web*. New York, NY, USA.

H.-K J. Kuo and C.-H. Lee. 2003. Discriminative Training of Natural Language Call Routers. *IEEE Trans. on Speech and Audio Processing*, 11(1):24–35.

A. D. Lawson, D. M. Harris, J. J. Grieco. 2003. Effect of Foreign Accent on Speech Recognition in the NATO N-4 Corpus. *Eurospeech*. September 1-4, Geneva, Switzerland.

G. Mishne, D. Carmel, R. Hoory, A. Roytman and A. Soffer. 2005. Automatic Analysis of Call-center Conversations. *Conference on Information and Knowledge Management*. October 31-November 5, Bremen, Germany.

M. Padmanabhan, G. Saon, J. Huang, B. Kingsbury and L. Mangu.. 2002. Automatic Speech Recognition Performance on a Voicemail Transcription Task. *IEEE Trans. on Speech and Audio Processing*, 10(7):433–442.

M. Tang, B. Pellom and K. Hacioglu. 2003. Call-type Classification and Unsupervised Training for the Call Center Domain. *Automatic Speech Recognition and Understanding Workshop*. November 30-December 4, St. Thomas, U S Virgin Islands.

J. Wright, A. Gorin and G. Riccardi. 1997. Automatic Acquisition of Salient Grammar Fragments for Call-type Classification. *Eurospeech*. September, Rhodes, Greece.