Mining Customer Care Dialogs for "Daily News"

Shona Douglas, Deepak Agarwal, Tirso Alonso, *Member, IEEE*, Robert M. Bell, Mazin Gilbert, *Senior Member, IEEE*, Deborah F. Swayne, and Chris Volinsky

***Abstract*—As large-scale deployments of spoken dialog systems in call centers become more common, a wealth of information is gathered about the call center business as well as the operation of these systems from their daily logs. This paper describes the "VoiceTone Daily News" data mining tool for analyzing this information and presenting it in a readily comprehensible and customizable form that is suitable for use by anyone from system designers to call center businesses. Relevant business and dialog features are extracted from the speech logs of caller-system interactions and tracked by trend analysis algorithms. We describe novel techniques for generating alerts on multiple data streams while avoiding redundant "knock-on" alerts. Some initial experiments with automated measures of dialog success are described as possible additional features to track. Features that move outside their expected bounds on a given day generate headlines as part of a website generated completely automatically from each day's logs. A "drill-down" facility allows headlines to be investigated all the way to viewing logs of individual interactions behind the headline and listening to the audio for individual turns.**

***Index Terms*—Alerts, business intelligence, data mining, dialog success, speech mining, speech recognition, spoken dialog systems, trend analysis.**

I. Introduction

OVER the past decade, there has been a significant rise in the number of speech services used across a wide variety of market segments to provide electronic contact mechanisms between people and organizations. Companies such as AT&T, Nuance, and Scansoft have deployed large-scale spoken dialog systems that receive several million calls per year and produce terabytes of data in the form of operation logs and speech audio [1]–[3]. Traditionally, this data is used for standard reporting, such as providing statistics on the number of calls per day, percentage of completed calls, hang-up rate, etc.

As the volume of data grows and the number of deployments rises, we face the major technical challenge of mining these interactive spoken dialogs to quickly extract critical application information and business intelligence. The basic question is whether one can explain not only what is happening but why. For example, when a spoken dialog system is first deployed, it is sometimes the case that a large percentage of callers hang up or insist on speaking to a live agent. It is not only important to discover how many people are declining to communicate with the system but to learn why. Can caller behavior be explained by geographical location, age, gender, or previous interactions? Such discoveries can help to improve both user experience and system performance. Even better, can these discoveries be generated automatically by algorithms that automatically examine callers' profiles and behavior?

Mining spoken dialog data for the purpose of extracting business intelligence is a new research area that has not been well explored in the past. It poses several interesting and new theoretical challenges including a) identifying relevant features of the dialogs, b) creating robust statistical and learning methods for capturing trends, discovering patterns automatically, and identifying critical service and business information, and c) applying language generation and visualization methods for presentation of key information.

In this paper, we describe the *VoiceTone Daily News* [4], which is a data mining tool that supplements AT&T's VoiceTone system by *automatically* extracting critical service information and business intelligence from records of dialogs resulting from a customer calling an automated help desk for some product or service supplier. VoiceTone is a call center service that creates, deploys, and hosts natural-language spoken dialog applications for large business customers [5]. The *Daily News* uses the spoken dialog interaction logs to automatically detect interesting and unexpected patterns and to present them in a daily web-based newsletter intended to resemble online news sites such as CNN.com or BBC.co.uk.

This paper describes the statistical and linguistic methods used to generate the *Daily News*. Section II describes the AT&T VoiceTone system. Section III describes the architecture of the system. In Section IV, we describe feature extraction and computation. The statistical methods for tracking and identifying trends are described in Section V. Section VI describes how the web site presents the analyses to the user. In Section VII, we summarize and give some future plans.

II. AT&T VoiceTone

VoiceTone is an AT&T hosted solution for contact center automation. AT&T VoiceTone applications are based on systematically mixing both open (or user-initiative) and directed (or system-initiative) dialog strategies to provide users the ability to speak naturally and complete their transactional requests. They represent a new generation of automated contact center services that greet the caller with the open ended prompt "*How may I help you?*". The main speech technologies behind VoiceTone services include Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialog Management (DM), Spoken Language Generation (SLG), and Text to Speech (TTS) Synthesis. Speech is recognized by AT&T's Watson ASR engine, which uses continuous-density hidden Markov models

Manuscript received August 13, 2004; revised April 14, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Isabel Trancoso.

The authors are with the AT&T Labs Research, Florham Park, NJ 07932 USA (e-mail: shona@research.att.com; dagarwal@research.att.com; tma@research.att.com; rbell@research.att.com; mazin@research.att.com; dfs@research.att.com; volinsky@research.att.com).

Digital Object Identifier 10.1109/TSA.2005.851878

System: Hello, this is Barney health care, how may I help you?
User: My medicine ran out last week ⟶ *Request(Refill)*
System: Would you like to refill your prescription?
User: yes uh my prescription number is PB14720. ⟶ *Yes, pres_num=PB14720*
System: Okay, I have ordered your new refill. Is there anything else I can help you with today?
User: No thanks ⟶ *No*
System: Thank you for calling Barney Health Care. Goodbye.

Fig. 1. Example of a VoiceTone dialog from a healthcare application.

and statistical language models. The Watson engine is based on acoustic and network optimization algorithms for robust and efficient speech recognition on large vocabulary tasks [6]. The recognized speech is processed by the SLU and is tagged for the presence of domain-relevant "named entities" using finite state automata and mapped to one of a set of **call types** using BoosTexter—a member of the AdaBoost family of large-margin classifiers [7]. The DM processes the output of the SLU along with the dialog history and determines an action. This is converted into words and then played out to the user. A dialog ends when a goal is achieved by completing a transaction, for instance, or routing the user to an appropriate destination. An example of a VoiceTone dialog is shown in Fig. 1, illustrating the system's classifications (*Request(Refill)*, *Yes*, *No*) and named-entity extraction (*pres_num=PB14720*).

The system logs each dialog interaction to a database as a set of time-stamped events generated by the various system components. Among the recorded events are prompts played, recognition output, classifier output, and call routing behavior. An XML document is generated daily, consisting of a sequence of dialog elements, one for each call, each consisting of a time-ordered sequence of the logged events. All VoiceTone applications share the same log file format but differ in their particular ASR grammars, SLU models, dialog design, classification output values and models, and final routing destinations.

III. VoiceTone Daily News System

The overall system architecture for the Daily News is shown in Fig. 2. Each night, the previous day's dialog logs are processed to extract dialog and turn level features, which are loaded into a database. (A turn is a segment of a dialog, which is defined in the next section.) A set of statistical analyses is run on these features to update cumulative statistics and to track and identify trends.

The numeric results of the statistical analyses are generated in a general format that allows them to be flexibly structured into a layered web site that presents high-level natural language descriptions in addition to detailed tabular and graphical reports. The web site integrates this information with dynamic query access to the feature database and to the SeeCalls visualization tool.

IV. Dialog Features

The first part of the feature extraction process is the application of XSLT templates to divide the stream of events in each XML dialog into a sequence of turns. (XSLT [8] is a template-based language for transforming XML documents into other XML formats, HTML, or plain text output.) A turn consists of a system prompt, the user response as recognized by the system, and any records associated with the system's processing of that response. Many kinds of analysis and evaluation relevant to spoken language systems naturally operate at the turn level.

From this representation, a set of features is extracted and/or computed for each dialog and for each turn within a dialog. Feature values are deposited in the database for run-time use and sent to the statistical analysis module. We use AT&T's Daytona database technology [9], which combines huge capacity and fast operation with simplicity of administration and the flexibility of using either its own 4GL Cymbal, SQL, or normal UNIX text processing tools on the database tables, which are simply flat files.

A. Extracted Features

Dialog features currently extracted directly from the logs include the originating telephone number for the call (**ANI**), the number of turns, the length of the call, any final routing destination (**RD**), and what is termed the **final actionable call type** (**FACT**). This is the last call type the classifier obtained in the course of the system's dialog with the user before routing or hangup, which is also "actionable" in the sense of not being vague or uninformative (if none are actionable, the last is used). The FACT and RD are primary features tracked by the "Daily News" alert system. The FACT is our closest approximation to the caller's intent. This is of particular interest to VoiceTone clients (banks, pharmacies, etc.), who want to know what their customers are calling about and how that is changing. The RD, particularly together with time of day information and geographic information derived from the ANI, provides information on call center load to support decision-making about provisioning and automation.

B. Deriving a Dialog Success Feature

We also obtain a dialog-level feature that is intended to capture the notion of dialog success. In terms of Paek's discussion of the purposes of dialog system evaluation [10], our requirement here is not for comparability of performance across difference dialog applications but, rather, to identify changes in one application over time and possibly correlate these with other factors we are tracking. For example, we would be interested in alerts like "The number of successful calls originating in Ohio has dropped by 54%," or "The number of calls about voicemail that are successful has risen by 34%." This success feature cannot be directly extracted from the logs but instead is generated by a machine learning process, which uses dialog and turn-level features extracted from the logs.

Target Success Measure: In order to train a classifier, we require a training corpus of dialogs labeled with their true success value. Previous work on dialog evaluation makes use of objective measures such as task completion rates and/or subjective ones such as user satisfaction [11]. In the PARADISE framework [12], multivariate linear regression is used to model the contribution of various objective and subjective dialog metrics to the target success measure of user satisfaction. Task completion has typically been found to be a significant predictor of user

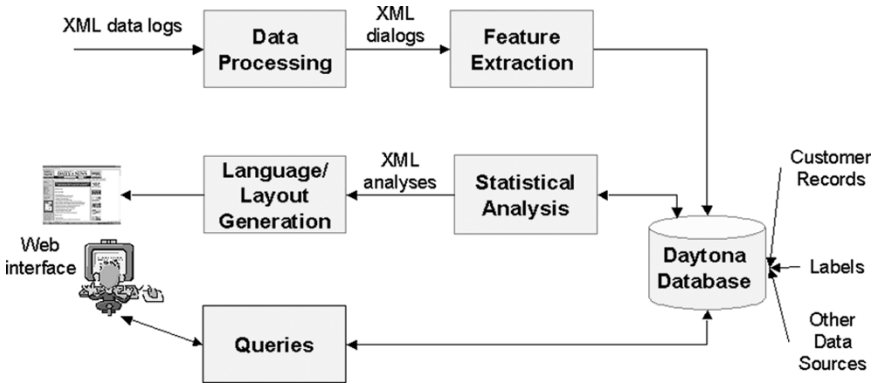


Fig. 2. VoiceTone Daily News system architecture.

satisfaction [13], [14], and of course, it is itself a important measure of success. However, for the present dataset, we have no human judgments at the dialog level: neither user satisfaction judgments nor task completion judgments. Instead, we use a dialog success measure determined automatically from human labeling of the individual dialog turns.

A small amount of the turn audio data is labeled with call classification types by hand, as part of the ongoing collection of training material for the application's classification module. We currently use these labels to generate a **turn label based (tlb)** success value for each dialog. Recall that we earlier defined the **final actionable call type** to be the last call type the classifier obtained before routing or hangup, which does not fall into a predetermined set of vague or uninformative call types, or simply the last if none exist. A dialog is considered successful if the final actionable call type is in the set of most actionable human-assigned labels for the dialog; this set consists of any actionable call types assigned to any of the turns, or if there are none of them, simply the set of call types for all the turns. That is, the system's final best guess as to the user's intention must be one of the most specific intentions the user has actually expressed. Otherwise, the dialog is considered to be a failure. For example, if the call types supplied for a given dialog by human labeling are "Request(Info)" (a vague call type) and "Request(Repair)" (an actionable call type), and the dialog manager's final actionable call type is "Request(Info)", that dialog is considered a failure since a more specific intention was expressed.

We hope in future work to validate this semi-automatic dialog-level success metric using human judgments of task completion and/or user satisfaction.

Input Features: The input features we use to predict dialog success must be fully automatic by the nature of the Daily News. Previous work on predicting dialog success has typically found a combination of automatic and human-labeled metrics to be useful, including task completion, time-related features, and speech recognition scores. In work in an air travel domain, Hastie *et al.* [14] additionally use features based on the DATE dialog act tagging system [13] (which aims to capture features of the conversational domain, the speech act dimension, and the task) to build a classification and regression model for predicting dialog success. Much of the benefit from the DATE features is obtained by using a special task feature ("GroundCheck") that is in effect a domain-specific surrogate for task completion since only relatively successful dialogs in the air travel domain reach the point of discussing ground transportation.

In the present work, we do not have access to such a clear indicator of task completion—nor would it provide a feature that would generalize over VoiceTone applications, as we would like. We also do not have speech recognition accuracy values (nor even, in the current version, recognizer confidence scores, which would be a useful substitute). However, some of the Routing Destination values indicate that specific caller intents have been determined with the above-threshold confidence, whereas others indicate routing to more general destinations, which are associated with uncertainty about the caller's intent. Thus, using Routing Destination as an input feature might capture some of the task completion and quality information that has been found to be useful in other work.

We have tested the following sets of turn and dialog features directly extracted from the logs:

- C (dialog contextual features): final actionable call type; routing destination.
- T (time features): number of turns; dialog duration; average turn duration; duration of first turn.
- N (turn features): sequence of turn call types; sequence of prompt ids. These are treated as gappy (sparse) n-gram features up to size 3, that is, up to three consecutive tokens including possible wild cards matching any token.

The n-gram turn features are interesting because they concisely designate sets of dialog trajectories, taking advantage of BoosTexter capabilities intended for natural language sequences [7]. For instance, the gappy n-gram $\texttt{p123} * \texttt{p56}$, which is a possible value for the turns prompt sequence feature, designates all dialogs containing a sequence of three turns where the first has prompt p123, the second can have any prompt, and the third has prompt p56. Finding which trajectory fragments are associated with successful and unsuccessful dialogs should provide useful insight for the designers of the dialog structure.

Predicting Dialog Success: We have trained a number of classifiers to predict dialog success for unlabeled dialogs, with the **tlb** success measure as the target, using various combinations of features in the unlabeled dialogs. We use the BoosTexter text classifier [7], formulating our essentially binary classification problem as a two-class problem with classes Y (success) and N (failure).

The training data consists of a set of example dialogs, each accompanied by a **tlb** label $\ell \in \{Y, N\}$ derived as described above. Each dialog is represented as a set of textual or real-

valued features selected according to the various experimental cases described below. BoosTexter constructs a simple categorization rule (a **decision tree stub**) for each term (word or ngram) in each text feature in the training data. Each such rule makes a prediction, with a confidence score, about each possible output label of a test dialog based solely on the presence or absence of that term in the dialog's feature set. (Rules for continuous-valued features test for values falling above or below some threshold.) We can say that the rules generate real-valued **weak hypotheses** of the form $h(x,\ell) = c_{0\ell}$ (if term is not in input x) or $c_{1\ell}$ (if term is in input x) for each example x and label ℓ.

The training phase is based on the idea of maintaining a distribution D_t of weights over the examples and labels in the training data to force the training process to concentrate on the examples and labels that have proved hardest to classify on the previous round. Initially, this distribution is uniform. At each round of training t, the weak hypothesis h_t, which minimizes the error on the (weighted) training set, is chosen. The parameter α_t, which will be used to weight the contribution of h_t at classification time, is computed based on its error on the training set. Then, the distribution is updated so that D_{t+1} gives more weight to the examples and labels that are most seriously misclassified with D_t ("boosts" them).

When used during classification, the score for a given label ℓ for dialog x is $f(x,\ell) = \sum_{t=1}^{T} \alpha_t h_t(x,\ell)$, where h_t is the weak hypothesis chosen on round t of training, and α_t is the weight assigned to it. The label with the best score is taken to be the prediction for x.

Experiments: We tested combinations of the input feature sets C, N, and T described above. Training for each of these experiments used 300 rounds of boosting on 5217 dialogs labeled with the **tlb** output feature $\ell \in \{Y, N\}$ from a VoiceTone customer care application. Testing used 1000 held-out dialogs from the same dataset.

Results are given for classifiers trained on the various individual feature sets plus a combination, compared with the **tlb** dialog success value, which is taken as "truth." Confusion matrices are shown in Table I. Table II shows the error rates (the percentage disagreement between the prediction and the **tlb** "truth") and κ values measuring the degree to which the agreement deviates from the expected. The combination $\mathbf{C+T+N}$ classifier using dialog contextual features, turn features, and time features performs best of those tested. Error rates for this case are around 16%. Only this classifier has $\kappa \geq 0.4$, which is sufficient to claim "moderate" agreement on κ according to [15]. It is interesting that the time features $\mathbf{T}$ are not useful alone but do contribute to the best-performing combination classifier. This may be because the typical length of a successful call depends on the call type and/or routing destination.

It should be borne in mind, however, that the predicted success rates (around 89%) are all significantly higher than the "true" success rate of 80%.

TABLE I
CONFUSION MATRICES FOR PREDICTIONS OF DIALOG SUCCESS. C: DIALOG CONTEXT FEATURES. T: TIME FEATURES. N: TURN FEATURES

		C		T		N		C+T+N	
		N	Y	N	Y	N	Y	N	Y
truth	N	72	128	0	200	48	152	79	121
	Y	36	764	7	793	28	772	36	764

TABLE II
ERROR RATES AND κ VALUES FOR PREDICTIONS OF DIALOG SUCCESS. C: DIALOG CONTEXT FEATURES. T: TIME FEATURES. N: TURN FEATURES

Classifier	Error (%)	κ
C	16.4%	0.38
T	20.7%	-0.01
N	18.0%	0.27
C+T+N	15.7%	0.42

V. ANALYSIS AND ALERTING

A. Simple Feature Alerts

Alerts are designed to identify abnormal occurrences in data streams. All features can be tracked by looking at the prevalence of the values of a given feature over a specified time period. We use standard statistical process control techniques [16] that generate an alert if the most recent prevalence of a value/feature pair is outside the expected interval, based on data from an appropriate historic period.

More specifically, at time t, let y_{it} represent the percentage of calls that have value i at time period t. For a given window of n time periods, calculate the mean percentage of calls with value i over that time period $m = \sum_{j=(t-n)}^{t-1} y_{it}/n$. Similarly calculate the variance over that time period s_i^2. Then, the expected interval for y_t, which is the percentage at time t, is $m_i \pm 3s_i$ or within three standard deviations from the mean. Anything outside of this range is considered out-of-process and generates an alert. Features with percentages outside a one standard deviation range are considered minor alerts.

Features of many types can be tracked via this mechanism. One of the most important for us is the **final actionable call type** since this alerts us to changes in customer behavior (e.g., a sharp increase in canceled orders or customer complaints). Integer and real-valued features such as number of turns and call durations are binned so that we may alert on them in a similar fashion.

B. Alerts on Cross-Classified Data Streams

Apart from monitoring univariate events, we are also interested in tracking events cross-classified by two or more factor variables. An example of this might be looking at deviations in caller intent by their geographic location. In situations like these, we are interested in tracking changes in the interactions between two factor variables and do not want to alert on multiple events that can all be attributed to changes in a few marginal events. For instance, we would not alert on a Repair problem at several locations simply because there was a spike in the percentage of Repair calls received by the system. We address this problem by adjusting for changes in row and column marginals. More specifically, let δ_{ijt} denote the change (computed as a difference between observed and expected values) at time t in the ith and jth levels of the row and column variables, respectively. Monitoring the residuals e_{ijt} obtained by fitting a linear model additive in the row and column effects gives an estimate of the interactions after adjusting for marginal changes.

Another difficulty that crops up is the well-known multiple testing problem—conducting a statistical test a large number of times would trigger alerts even when nothing is amiss. Therefore, if we conduct a test 2000 times with a fixed false positive rate of 5%, we will see 100 false anomalies. The challenge

is to find a statistical procedure that would ameliorate this and yet be capable of detecting a large percentage of true changes. When tracking cross-classified data streams with I rows and J columns, one has to conduct IJ tests, which can get large even for moderate values of I and J. The threshold-based testing technique employed in the case of univariate streams did not work well in this situation. We use a Bayesian technique called **hbmix**, which has outperformed the naive threshold-based technique. We give a brief description of the algorithm here. A complete description of all technical details may be obtained from the second author upon request.

Let $E(e_{ijt}) = \Delta_{ijt}$ (unknown) and $\text{Var}(e_{ijt}) = s_{ijt}^2$ (known from historic data). The threshold-based method estimates Δ_{ijt} with e_{ijt} and declares an alert if $|e_{ijt}/s_{ijt}| > M$ (usually 3). The idea of the Bayesian procedure is to penalize for conducting multiple tests by smoothing the estimates of Δ_{ijt}. This is done by assuming that the Δ_{ijt}'s are drawn from a statistical distribution (known as the prior distribution) and getting revised estimates based on the distribution of Δ_{ijt}'s conditional on e_{ijt}'s (called the posterior distribution). The prior distribution controls the amount of smoothing and, hence, the performance of the method. In the simplest case, if the prior distribution is Gaussian centered at zero with variance τ_t^2 (estimated from data), we declare an event to be an anomaly if $|e_{ijt}/s_{ijt}| > M\sqrt{(1 + s_{ijt}^2/\tau_t^2)}$. (The M here could be different than threshold; usually, it is smaller.) Thus, the penalty imposed by the Bayesian procedure for conducting multiple tests is determined by the scale parameter τ_t^2. Events with high statistical variance have higher threshold, whereas for events with small variance relative to τ^2, the procedure converges to the threshold-based rule. The **hbmix** procedure enriches the one-parameter prior to a two-parameter prior, which assumes that a proportion P_t of Δ_{ijt}'s being monitored are exactly zero, and the remainder $(1 - P_t)$ are drawn from a Gaussian with mean zero and variance τ_t^2. This is a reasonable assumption when monitoring a large number of events where it is often the case that a sizable proportion of events do not change significantly at a given time point. Note that the one-parameter prior is nested within the two-parameter prior ($P_t = 0$ gives back the one-parameter model), and hence, if there are significant changes in a large number of events giving an estimate of P_t close to zero, hbmix reduces to the one-parameter model.

We compare the performance of hbmix relative to a threshold-based algorithm that declares a change if $|\delta_{it}/s_{it}| > M$ using a simulation. Our experiment compares the two methods based on performance at a single time point (the last one in the simulated streams). We simulate K streams at $n + 1$ time points introducing "shocks" only at the last time point and compare the false positive and false negative rates. Since the difference in the false positives and negatives is not symmetric, we tweak the value of M so that the false negative rate matches the one obtained by using hbmix.

The procedure is as follows.

1) Simulate a vector of means $(\mu_1, \cdots, \mu_K)$ for the K data streams such that μ_i's are independent and identically distributed (iid) $N(0, \tau^2)$. We set $\tau^2 = 1$.
2) Simulate a vector of variances $(s_1^2, \cdots, s_K^2)$ for the K data streams such that s_i^2's are iid inv-gamma (scale $= a$, shape $= b$). We set $a = 0.5$ and $b = 2$. This distribution is quite dispersed with 2.5%, 25%, 50%, 75%, 97.5% being 0.09, 0.18, 0.30, 0.52, and 2.06, respectively.
3) For the ith stream, simulate $n + 1$ observations as iid $N(\mu_i, s_i^2)$, $i = 1, 2, \cdots, K$.
4) Add "shocks" at time $n + 1$ to 100 randomly selected streams from uniform $(-A, A)$ distribution (we use $A = 15$).
5) Change detection is done at $n+1$ using the first n observations with $n = 10$ in this case.
6) The threshold M is tweaked to match the false negative rate of hbmix.
7) Steps 1–6 are repeated, in this case 100 times.

Results are reported in Table III.

TABLE III
COMPARING **THRESHOLD AND hbmix**

K	Total shocks	false neg rate (%)	M	false pos (%) (hbmix)	false pos (%) (threshold)
500	100	11.4	2.8	7.9	12.9
1000	100	12.0	3.1	9.4	15.9
2000	100	14.0	3.3	13.4	21.0
5000	100	14.5	3.6	19.4	30.2

The simulation shows the false positive rates for hbmix to be significantly smaller than the threshold-based technique. Notice how the performance of the Bayesian procedure gets better with an increase in the number of streams being monitored, which is a characteristic that is key to the success of the procedure in the context of cross-classified data streams.

VI. INFORMATION PRESENTATION

The frequency and alert information for all tracked features plus a large number of preconstructed plots accompanied by XML descriptions are processed fully automatically into an interactive, hierarchically organized web site. Alerts and carefully selected summary plots and tables provide high-level information and serve as entry points for exploration of a volume of data that is intractably large without such prioritization. Users can investigate alerts by querying the database, conducting their own analyses, and drilling down to specific dialogs and audio segments.

A. Web Site Top Level

A typical front page of the web site is shown in Fig. 3.

Daily Tables: On the right, some tables give the top few values for features that are nearly always of interest to the end user, with arrows (after the manner of a stock tracking page) giving a coarse-grained indication of the magnitude of movement for each feature-value pair. Feature-value pairs that exceed the alerting threshold described in Section V (e.g., **RouteToCollections** in Fig. 3) are accompanied by double arrows; those that exceed the minor alerting threshold have a single arrow [e.g., **Request(Make_Payment)**].

Each line in these tables is a link to a page showing a plot of the feature-value pair's frequency and proportional frequency movements over the period of the moving window

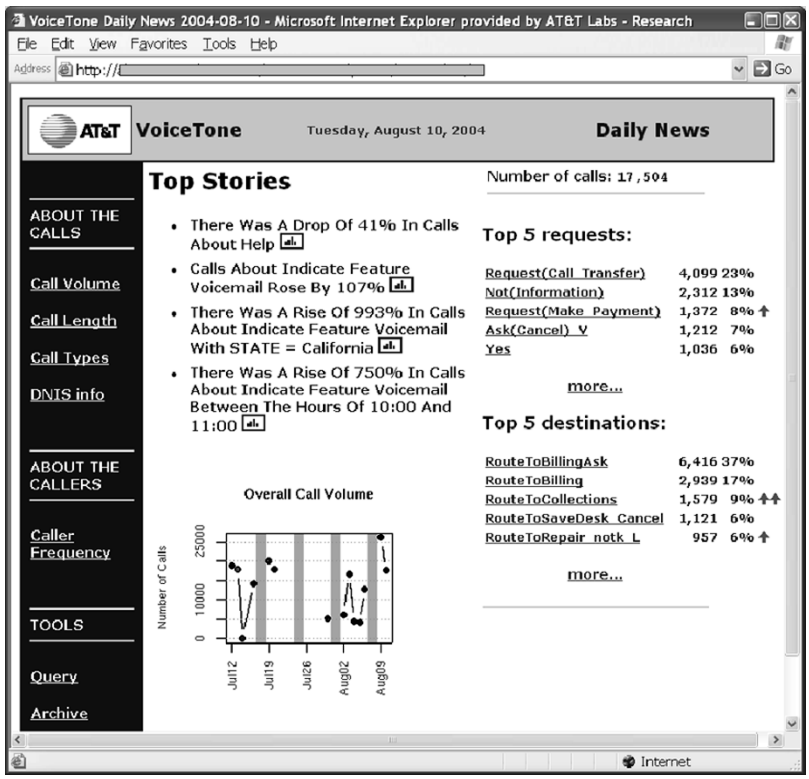


Fig. 3. Front page: A simulated example.

of trend tracking. These Time Series plots [see Fig. 4(a)] plainly show overall trends over the past few weeks, weekly periodic trends, and outliers. This plot type is also used prominently on the front page to display the number of calls into the system over the most recent time period. The plot is visually broken into weekly segments by light gray vertical bars on the weekends, which allow the weekly trends to be seen more readily.

Detailed Statistics: On the left, there are links to pages of plots and tables detailing information about the calls and callers over different periods of time.

Under the "Call Volume" link, we show a map that color-codes states based on the volume of calls originating there. A map [see Fig. 4(d)] is a convenient way of seeing if there are spatial effects in the data. We plan to extend this to plot spatial units that are more granular than the state level. Our maps divide the volume levels into four bins and use a monochromatic color sequence (beige to red) to designate the volumes in different areas.

On the same page, we show a histogram of the number of calls by time of day. These plots allow the user to see modes, skewness, and outliers in a convenient form. The example in Fig. 4(c) clearly shows a dip in volume around lunch time.

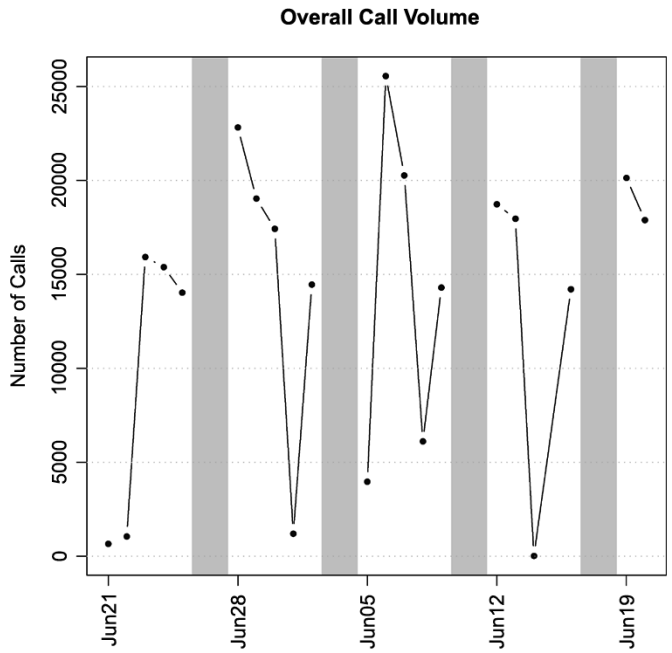
Under the "Call Length" link on the left-hand panel is a boxplot of call length in seconds for the top few final actionable call type values [Fig. 4(b)]. Boxplots show interactions between two variables in a concise manner. Usually, one is a categorical independent variable and the other a real-valued response variable. The box in the middle represents the interquartile range, with a line drawn at the median, and whiskers extending out to the minimum and maximum of the data.

News Headlines: The middle section contains the day's news headlines. Features generating an alert here are brought to the user's attention. Links on each headline allow drill-down to further investigate the dialog subset identified in the alert.

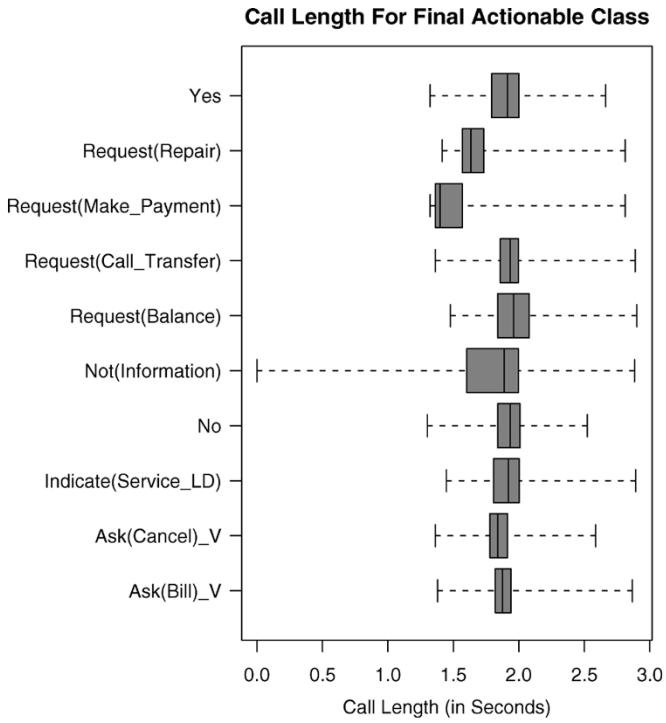
Newsworthy features generate front-page headlines in simple natural language using XSLT templates. Input from the alerting module takes the form of `trend` elements. Heavy use has been made of the **mode** feature of XSLT to allow a given piece of input data to be realized differently, depending on the linguistic context (following [17] and [18]). We also use a simple randomizing mechanism to produce lexical and structural variation. For example, there is a template that matches `trend` nodes with positive `change` attributes in mode `MovementV`, which outputs the movement verbs *rose*, *went up*, or *increased*, depending on the value of a randomly generated number. If the `change` attribute is instead being realized in a noun phrase context, the `trend` node will instead be matched in mode `MovementNP` to generate a noun phrase like *a rise*, *an increase*. The choice between the two sentence-level structures is made using the same randomization idiom.

A single trend instance may generate "*calls about Indicate Feature Voicemail rose by 750% between the hours of 10:00 and 11:00*," "*the number of calls about Indicate Feature Voicemail increased by 750% between 10 pm and 11 am*," etc., as well as "*there was a rise of 750% in calls about Indicate Feature Voicemail between the hours of 10:00 and 11:00*," which appears in Fig. 3.

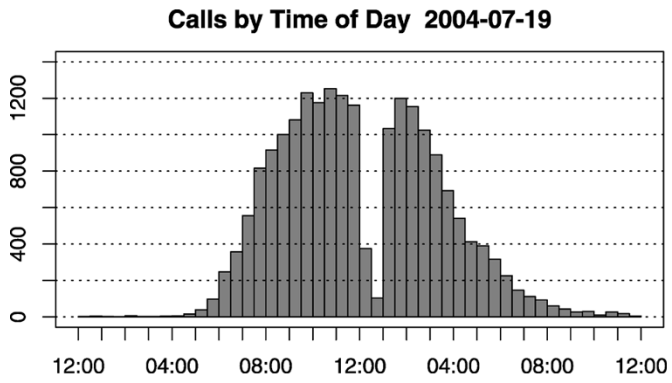
The use of XSLT templates allows modular expansion of the coverage as different features or trend variables are introduced. For instance, the default `FeatureDescriptionPP` template merely outputs *feature=value*. When a new feature is added (for instance, the originating state of the call), appropriate behavior can be produced by simply adding more specific `FeatureDescriptionNP` templates that match the new `feature` nodes better than the default templates do, e.g., to output



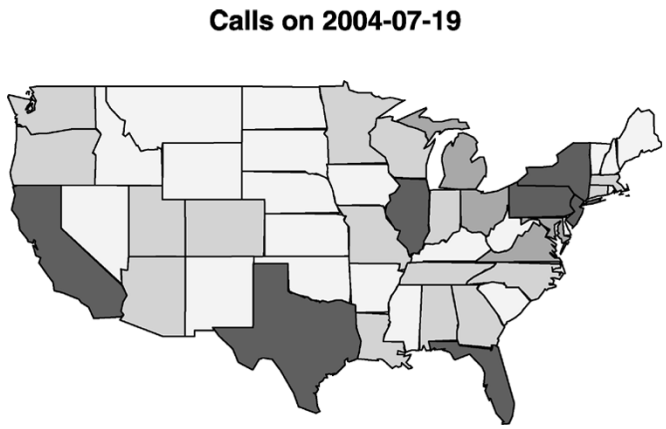
(a)



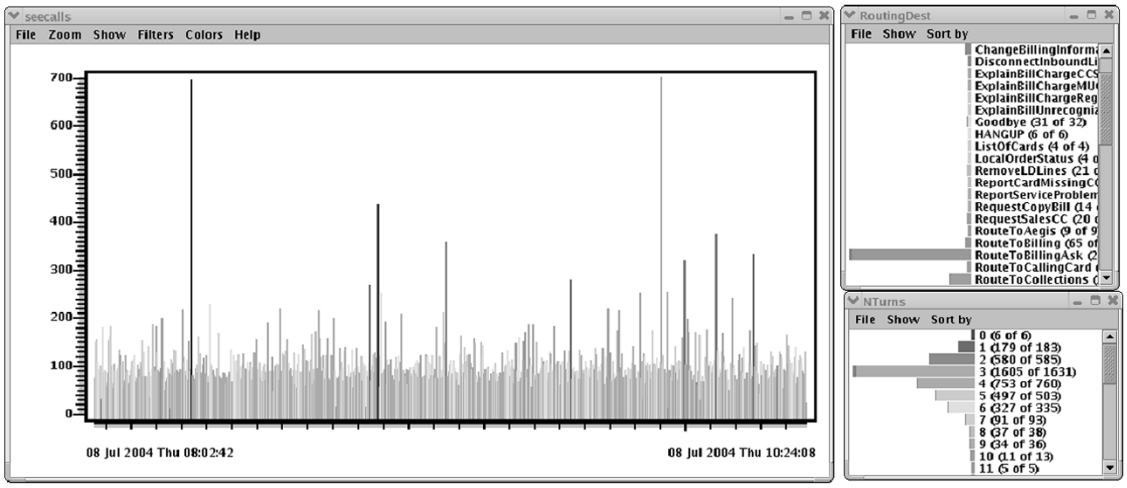
(b)



(c)



(d)



(e)

Fig. 4. Five main plot categories. (a) Time series plot of the number of calls per day during a month. The gray vertical bars mark the weekends and make it easier to see weekly trends. (b) Boxplots displaying the distribution of call length conditioning on Final Actionable Call Type. (c) Histogram of the number of calls by time of day. Note the dip in volume around lunch time. (d) Map of the number of calls by state. (e) SeeCalls display of dialog duration in a 2-hr span. The two barplots of dialog features at the right is linked to the main plot; many other interactions are possible.

"*from New Jersey*" when a template with pattern `feature[@name="State"]` is matched and `feature/@value` is `New Jersey`.

B. Drill-Down

A link from a headline or a daily table entry allows the user to query the database to retrieve more detailed information about the dialogs behind the headlines. Following these links brings up a form to query the database that is prepopulated to select dialogs from the current day and the feature-value pair that is currently being investigated. All query features can be varied in the form, for instance, to add other values for the same feature in order to make comparisons.

There are currently four display options for the selected dialogs from which the user can further drill down.

Turn-Level Display and Audio Access: The default display is a table that shows a line for each dialog, corresponding to the dialog's record in the database. Drilling down further, a dialog may be selected to show a page with the sequence of turns, with prompt, recognition output, and call type for each turn. There is also a link to the comprehensive system log of that dialog, which shows prompts, responses, recognition, understanding, system state, etc. Specific calls can be examined for clues as to the cause of an alert. This log itself has links to the actual audio files of the recorded utterances and allows listening to the complete dialog, e.g. to probe speech recognition issues.

Comparison Plots: The user can compare the distribution of the attributes from the dialogs returned by the query with a baseline set of dialogs. These plots display histograms of each attribute (e.g. number of turns) side by side with plots of the same attributes for a different set of dialogs. Using these plots, one can do a simple investigation by comparing the distributions for dialogs associated with an alert to distributions for a baseline set of dialogs. Currently, we are using a single baseline set for all queries; in future work, we aim to tailor the baseline dialogs to be the most sensible comparison set for a given query.

SeeCalls: The selected dialogs may be sent to SeeCalls, which is a highly interactive visualization and data exploration tool designed for transaction data. For VoiceTone data, we display a set of dialogs as a series of vertical spikes positioned along a time axis, with height proportional to dialog duration [see Fig. 4(e)]. This display is linked to a set of barplots corresponding to the attributes in the dialog database, e.g., RoutingDestination or number of turns, so the spikes are usually colored according to one of those attributes. In addition, the main plot may be manipulated and queried by zooming in on a region of the data to view the plot in more detail or to extract database records.

Dialog Trajectories: We use technology described in [19] to show a directed graph representation of the call flow of the set of dialogs selected by the query.

VII. Discussion

Applying state-of-the-art statistical analysis and visualization techniques to the dynamic behavior of spoken dialog systems offers great benefits both for system evaluation and diagnosis, as well as for domain-oriented data mining. In this paper, we have described the VoiceTone Daily News system for generating newspaper-style reports fully automatically on a daily basis from spoken dialog system logs. The Daily News uses a combination of natural language alerting, tables, and carefully chosen graphics to provide high-level access points to data of interest. A comprehensive drill-down capacity allows the user to select and view parts of the data in various ways, either as aggregates through graphical visualization or down to the granularity of individual dialogs and turns. We have used state-of-the-art alerting and visualization techniques to extend the kind of reporting and alerting offered by commercial Business Intelligence systems providers (such as Business Objects [20] or Cognos [21]) to spoken language dialog systems.

We plan to extend this work to consider other features, including information from sources other than the dialog system logs, such as customer databases and features derived from the speech recognition output and recorded audio, where available. We also intend to allow customization of headline generation and other aspects of the presentation to better support the different needs of business intelligence and system diagnosis. In the area of derived features, we hope to improve the success measure we have been using by evaluating it against human judgments at the dialog level.

References

[1] AT&T. [Online] http://www.att.com

[2] Nuance Communications, Inc.., Menlo Park, CA. [Online] http://www.nuance.com

[3] Scansoft, Inc.., Burlington, MA. [Online] http://www.scansoft.com

[4] S. Douglas, D. Agarwal, T. Alonso, R. Bell, M. Rahim, D. F. Swayne, and C. Volinsky, "Mining customer care dialogs for "Daily News"," in *Proc. INTERSPEECH*, Jeju, Korea, 2004.

[5] AT&T. VoiceTone. [Online] http://www.business.att.com/service_overview.jsp?repoid=Product&repoitem=voicetone&serv=voicetone&serv_port=voice_svcs&serv_fam=contact_ctr

[6] A. Ljolje, D. Hindle, M. Riley, and R. Sproat, "The AT&T LVCSR-2000 system," in *Proc. Speech Transcription Workshop*. College Park, MD, May 2000.

[7] R. E. Schapire and Y. Singer, "BoosTexter: a boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[8] W. W. W. Consortium. XSL Transformations (XSLT), W3C Recommendation. [Online] http://www.w3.org/TR/xslt

[9] R. Greer, "Daytona and the fourth generation language cymbal," in *Proc. Database Conf.*. Philadelphia, PA, Jun. 1999. [Online} Available: http://www.research.att.com/projects/daytona.

[10] T. Paek, "Empirical methods for evaluating dialog systems," in *Proc. 39th Annu. Meet. Assoc. Computational Linguistics 10th Conf. Eur. Chapter*, Toulouse, France, 2001.

[11] K. S. Hone and R. Graham, "Subjective assessment of speech-system interface usability," in *Proc. EUROSPEECH*, Aalborg, Denmark, 2001, pp. 2083–2086.

[12] M. Walker, D. Litman, C. Kamm, and A. Abella, "Evaluating spoken dialogue agents with PARADISE: two case studies," *Comput. Speech Language*, vol. 12, no. 3, 1998.

[13] M. Walker, R. Passonneau, and J. Boland, "Quantitative and qualitative evaluation of DaRPA communicator spoken dialogue systems," in *Proc. 39th Annu. Meet. Assoc. Computational Linguistics 10th Conf. Eur. Chapter*, Toulouse, France, 2001.

[14] H. W. Hastie, R. Prasad, and M. A. Walker, "What's the trouble: Automatically identifying problematic dialogs in DARPA communicator dialog systems," in *Proc. 40th Annu. Meet. Assoc. Computational Linguistics*, Philadelphia, PA, 2002.

[15] J. Landis and G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 45, pp. 255–268, 1977.

[16] G. E. Box, *Time series analysis: forecasting and control*. San Francisco, CA: Holden-Day, 1970.

[17] A. Cawsey, "Presenting tailored resource descriptions: will XSLT do the job?," in *Proc. 9th Int. Conf. World Wide Web*, May 2000.
[18] G. Wilcock, "Pipelines, templates, and transformations: XML for natural language generation," in *Proc. First NLP and XML Workshop*, N. Nomura and C. Nakabasami, Eds., Tokyo, Japan, Nov. 2001.
[19] A. Abella, J. Wright, and A. Gorin, "Dialog trajectory analysis," in *Proc. IEEE ICASSP*, Montreal, QC, Canada, 2004.
[20] Business Objects. Business Intelligence From Business Objects, San Jose, CA. [Online] http://www.businessobjects.com
[21] Cognos. Business Intelligence and Performance Management Solutions, Burlington, MA. [Online] http://www.cognos.com

Shona Douglas has been a member of the Spoken Language Understanding and Dialog Division at AT&T Labs Research, Florham Park, NJ, since 1998. Her research at AT&T, and previously at the University of Edinburgh, Edinburgh, U.K., has been in the area of language technology, including grammar and style checking, information extraction from semi-structured data, statistical machine translation, spoken language understanding, and dialog systems. Most recently, she has been working on data mining from spoken dialog system data.

Deepak Agarwal received the Masters degree from the Indian Statistical Institute in 1997, and the Ph.D. from University of Connecticut, Storrs, in 2001, all in statistics.

He has been a researcher in statistics at AT&T Labs, Florham Park, NJ, since 2001. He has published about 15 papers in areas of Spatial Statistics, Data Mining, Social Networks, and Speech Mining.

Dr. Agarwal received the best applications paper award at the SIAM Data Mining Conference, 2004, and the Best Student Paper Award in environmental application at the Joint Statistical Meetings, 2001. He has served on program committees for KDD and SIAM, has served on two NSF panels, and has been a reviewer for several journals and conferences.

Tirso Alonso (M'93) received the M.S. degree in electrical engineering from Lehigh University, Bethlehem, PA, in 1992.

In 1992, he joined Bell Labs, Murray Hill, NJ, where he worked on various aspects of digital video compression and noise reduction algorithms for digital telephony applications. He is now a senior technical staff member of the Speech Services Research Department, AT&T Labs-Research, Florham Park, NJ, where he is working on distributed architectures for speech-enabled applications, speech recognition, natural language understanding technologies, and speech data mining.

Robert M. Bell is a member of the Statistics Research Department at AT&T Labs-Research, Florham Park, NJ. His research interests include survey research methods, analysis of data from complex samples, and record linkage methods.

Dr. Bell is a fellow of the American Statistical Association and is currently a member of the Committee on National Statistics organized by the National Academies and a member of the board of the National Institute of Statistical Sciences.

Mazin Gilbert (formerly Rahim) (SM'97) received the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K.

He holds 11 U.S. patents and is the Director of the Spoken Language Understanding and Dialog Division at AT&T Labs Research, Florham Park, NJ. His research lies in the areas of speech and language processing, machine learning, speech data mining, and spoken and multimodal dialog systems. He has been instrumental in driving the speech and language research and software tools that led to the AT&TVoiceTone offer in 2003. He has over 70 technical publications and is the author of the book *Artificial Neural Networks for Speech Analysis/Synthesis* (London, U.K.: Chapman and Hall, 1994). He is a Teaching Professor at Princeton University, Princeton, NJ.

Dr. Gilbert is currently the Chair of the CAIP Industry Board at Rutgers University and Chair of the IEEE Speech Technical Committee. He is a recipient of several national and international awards.

Deborah F. Swayne received the B.A. degree in african linguistics from the University of Wisconsin, Madison, in 1973 and the M.S. degree in statistics from Rutgers University, New Brunswick, NJ, in 1988.

She has worked in the Bell System "diaspora" since 1979, first at Bell Laboratories, then Bellcore (now Telcordia), and now AT&T Labs, Florham Park, NJ. Most of her research has been in interactive multidimensional data visualization, as exemplified by the XGobi and GGobi software tools.

Chris Volinsky received the Ph.D. degree from the University of Washington, Seattle, in 1997 under advisor A. Raftery, studying Bayesian Model Averaging and its applications to survival analysis.

He is the Director of the Statistics Research Department at AT&T Research, Florham Park, NJ. He joined AT&T Labs-Research in 1997 and became Director of the Statistics Research Department in 2004. His research at AT&T focuses on the analysis of massive graphs, including models for graph matching, statistical computation and visualization, and fraud detection in telecommunications.