# TOWARDS LEARNING TO CONVERSE: STRUCTURING TASK-ORIENTED HUMAN-HUMAN DIALOGS

*Srinivas Bangalore, Giuseppe Di Fabbrizio*

AT&T Labs-Research
180 Park Ave
Florham Park, NJ 07932

*Amanda Stent*

Dept of Computer Science
Stony Brook University
Stony Brook, NY

## ABSTRACT

Data-driven techniques have influenced many aspects of speech and language processing. Models derived from data are generally more robust than hand-crafted systems since they better reflect the distributions of the phenomena being modeled. With the availability of large spoken dialog corpora, dialog management can now reap the benefit of data-driven techniques. In this paper, we present our view of structuring human-human dialogs in order to learn models for human-machine dialogs. We present the problems of dialog segmentation and dialog act labeling, develop a model for predicting and labeling topic segments and dialog acts and evaluate the model on customer-agent dialogs from a catalog service domain.

## 1. INTRODUCTION

As large amounts of language data have become more widely available, approaches to sentence-level processing tasks such as parsing, language modeling, named-entity detection and machine translation have become increasingly data-driven and empirical. Models for these tasks can be trained to capture the distributions of phenomena in the data resulting in improved robustness and adaptability. However, this trend has yet to significantly impact approaches to dialog management in dialog systems. Dialog managers (both plan-based and call-flow based) have traditionally been hand-crafted and consequently somewhat brittle and rigid. With the ability to record, store and process large numbers of human-human dialogs (e.g. from call centers), we anticipate that data-driven methods will increasingly influence approaches to dialog management.

A successful dialog system relies on the synergistic working of several components: speech recognition (ASR), spoken language understanding (SLU), dialog management (DM), language generation (LG) and text-to-speech synthesis (TTS). While data-driven approaches to ASR and SLU are prevalent, such approaches to DM, LG and TTS are much less well-developed. In on-going work, we are investigating data-driven approaches for building all components of spoken dialog systems.

In this paper, we present one aspect of this research program – *inferring models that predict the structure of task-oriented dialogs*. In Section 2, we review current approaches to building dialog systems. In Section 3, we review related work in data-driven dialog modeling. In Section 4, we present our view of analyzing the structure of task-oriented human-human dialogs. In Section 5, we discuss the problems of segmenting and labeling dialog structure and building models for predicting these labels. In Section 6, we report experimental results on a large corpus from a catalog ordering service.

## 2. CURRENT METHODOLOGY FOR BUILDING DIALOG SYSTEMS

Current approaches to building dialog systems involve several manual steps and careful crafting of different modules for a particular domain or application. The process starts with a small scale "Wizard-of-Oz" data collection where subjects talk to a machine driven by a human 'behind the curtains'. A user experience (UE) engineer analyzes the collected dialogs, subject matter expert interviews, user testimonials and other evidence (e.g. customer care history records). The UE engineer uses this information to design some system functionalities, mainly: the system's semantic scope (e.g. call-types in the case of call routing systems), the LG model, and the DM strategy. A larger automated data collection follows [1] and the collected data is transcribed and labeled by expert labelers following the UE engineer recommendations. Finally, the transcribed and labeled data is used to train both the ASR and the SLU.

This approach has proven itself in many deployed dialog systems. However, the initial UE requirements phase is an expensive and error prone process because it involves non-trivial design decisions that can only be evaluated after system deployment. Moreover, scalability is compromised by the time, cost and high level of UE know-how needed to reach a consistent design.

In the AT&T VoiceTone® [2] product, the process of building speech-enabled automated contact center services has been formalized and cast into a scalable commercial environment in which dialog components developed for different applications are reused and adapted. However, we still believe that exploiting dialog data to train/adapt or complement hand-crafted components will be vital for robust and adaptable spoken dialog systems.

## 3. RELATED WORK

Automatic creation of part or all of a dialog system from data is a research area of increasing interest (e.g. [3, 4, 5, 6, 7]). However, as described in the previous section, the data are used only indirectly; to help humans write dialog scripts for dialog management or templates for response generation. For example, [7] limit their automatic acquisition of dialog behavior to acquiring domain information for discussion in dialog. [4] focuses on automatic acquisition of sentence planning and surface realization rules from labeled/annotated corpora. In the work most similar to ours, [6] used a corpus of transcribed and annotated telephone conversations from the banking domain. They trained separate task and dialog act classifiers on this corpus. For task identification they report an accuracy of 85% (true task is one of the top 2 results returned by the classifier); for dialog act tagging they report 86% accuracy.

There has been considerable research on automatic dialog act tagging (e.g. [8, 9, 10, 11, 12]) and the building of dialog models from data annotated with dialog act tags (e.g. [13, 9, 14, 15]). Sev-

eral disambiguation methods (hidden Markov models, maximum entropy models, decision trees, SVMs) that include a variety of features (cue phrases, word n-grams, prosodic features, syntactic features, dialog history) have been used for dialog act tagging. However, some of this research used text or read speech rather than spoken dialog, and the tagging schemes used were not specific enough to be used for generation. More recent work has looked at human-human dialog in meetings ([16, 17]), and at human-computer dialog [18], but has focused on prosodic features.

## 4. STRUCTURAL ANALYSIS OF A DIALOG

In order to infer models of task-oriented dialog, we annotate human-human dialogs according to the structure shown in Figure 1 and then train models to predict this structure. We consider a dialog to be composed of a set of high-level tasks (e.g. *ordering, canceling, order_checking*). A subset of tasks might appear in any order in a dialog. Each task is composed of a sequence of subtasks/topics, and each subtask is composed of a sequence of interactions between the user and the agent. Each interaction is represented as a tuple consisting of a dialog act and a set of predicate-argument relations. Each clause in an utterance realizes one predicate-argument relation. To build our dialog structure, we apply several processes to input utterances: utterance segmentation (Section 4.1), syntactic annotation (Section 4.2), dialog act tagging (Section 4.3) and subtask labeling (Section 5).

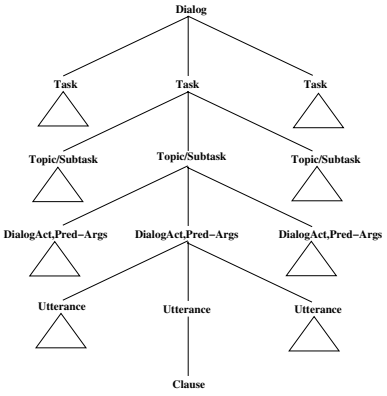


**Fig. 1**. Structural analysis of a dialog

### 4.1. Utterance Segmentation

The task of "cleaning up" spoken language utterances by detecting and removing speech repairs and dysfluencies and identifying sentence boundaries has been a focus of spoken language parsing research for several years [19, 20, 21, 22, 23, 24, 25, 26, 27]. In related work [27], we have developed a system that takes as input the ASR output text for a user's utterance, and that outputs clauses. The system annotates an utterance for sentence boundaries, restarts and repairs, and identifies coordinating conjunctions, filled pauses and discourse markers. These annotations are done using a cascade of classifiers, details of which are described in [27].

### 4.2. Syntactic Annotation

In order to extract predicate-argument relations for each clause, we use a dependency parser [28] based on supertags [29]. Supertags encapsulate predicate-argument information in a local structure. They are composed with each other using the substitution and adjunction operations of Tree-Adjoining Grammars [30] to derive a dependency analysis of an utterance.

We also detect and extract named entities. By contrast to approaches to named entity extraction discussed in the literature, we cannot rely on annotated data. In on-going work, we are exploring techniques that use meta-data (e.g. application-specific databases), as well as linguistic cues in utterances, to identify named entities from unannotated data.

### 4.3. Dialog Act Tagging

We use a domain-specific dialog act tagging scheme based on an adapted version of DAMSL [13]. The DAMSL scheme is quite comprehensive, but the multi-dimensionality of the scheme makes the building of models from DAMSL-tagged data complex [9]. Furthermore, the generality of the DAMSL tags reduces their utility for natural language generation. We were particularly concerned with obtaining sufficient discriminatory power between different types of statement (for generation), and to include an out-of-domain tag (for interpretation). Other tagging schemes, such as the Maptask scheme [31], are also too general for our purposes. We provide a sample list of our dialog act tags in Table 2. Our experiments in automatic dialog act tagging are described in Section 6.3.

## 5. MODELING SUBTASK SEGMENTATION

As discussed previously, in our application domain (customer service) most dialogs contain discussion of one task, composed of several subtasks. For example, an *order placement* task is typically composed of the sequence *opening, contact-information, order-item, related-offers, summary*. The goal of subtask segmentation is to predict if the current utterance in the dialog is part of the current subtask or starts a new subtask. We model this prediction problem as a classification task as follows: given a sequence of utterances $u_i$ in a dialog $U = u_1, u_2, \ldots, u_p$ and a subtask label vocabulary ($t_i \in \mathcal{T}$), we need to predict the best subtask label sequence $T^* = t_1, t_2, \ldots, t_q$ as shown in Equation 1.

$$T^* = \underset{T=t_1,..,t_q}{argmax}\, P(T|U) \quad (1)$$

We refine this model by viewing each subtask as having a *begin*, *middle* and an *end* utterance. The refined vocabulary of subtask labels is denoted as $\mathcal{T}_r = \{t_i^b, t_i^m, t_i^e \mid t_i \in \mathcal{T}\}$. Furthermore, the search is limited to the label sequences that respect precedence among the refined labels (*begin* < *middle* < *end*). This *well-formedness constraint* is captured in a grammar *G* encoded as a regular expression ($L(G) = \bigcup_i (t_i^b\ (t_i^m)^*\ t_i^e)^*$). Thus the search for the refined label sequence is shown in Equation 2. We assume markov independence between labels and rewrite Equation 2 as Equation 3. We use a classifier for assigning a refined subtask label to each utterance conditioned on a vector of local contextual features ($\mathbf{F}_i$). We use the speaker identity (agent versus customer) and $n$-grams computed from the current utterance, previous utterance, next utterance and previous turn as local contextual features. We also investigate the effect of including the previous subtask predictions contextual features for subtask label prediction. In order to cope with the prediction errors of the classifier, we approximate $L(G)$ with an $n$-gram language model on sequences of the refined tag labels.

$$T_r^* = \underset{T_r=t_1,\ldots,t_q;T_r\in L(G)}{argmax} P(T_r|U) \quad (2)$$

$$\approx \underset{T_r=t_1,\ldots,t_q;T_r\in L(G)}{argmax} \prod_{i=1}^{q} P(t_i|\mathbf{F}_i) \quad (3)$$

We use a discriminative classification model, Boostexter, based on the boosting family of algorithms first proposed in [32] in order to estimate $P(t_i|\mathbf{F}_i)$. In the boosting framework, each feature is considered as a weak classifier and a set of features are iteratively selected to be combined to obtain an accurate classifier. The set of selected base classifiers constitutes the model ($f$). As described in [33], Boostexter uses *confidence rated* classifiers $h$ that, rather than providing a binary decision of -1 or +1, output a real number $h(x)$ whose sign (- or +) is interpreted as a prediction, and whose magnitude $|h(x)|$ is a measure of "confidence".

The output of the model on a new instance ($x$) is computed as $f(x) = \sum_{t=1}^{T} h_t(x)$, i.e. the sum of confidence of all the classifiers $h_t$ selected during the training process. The real-valued predictions of the final classifier $f$ can be converted into probabilities by a logistic function transform; that is

$$P(y = 1|x) = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}} \quad (4)$$

## 6. EXPERIMENTS AND RESULTS

In this section, we present the results of the experiments for predicting subtask and dialog act labeling.

### 6.1. Data

We used 915 telephone-based customer-agent dialogs related to the task of ordering products from a catalog. Each dialog was transcribed by hand; all numbers (telephone, credit card, etc.) were removed for privacy reasons. The average dialog lasted for 3.71 minutes and included 61.45 changes of speaker. A single customer-service representative might participate in several dialogs, but customers are represented by only one dialog each. Although the majority of the dialogs were on-topic, some were idiosyncratic, including: requests for order corrections, transfers to customer service, incorrectly dialed numbers, and long friendly out-of-domain asides. Annotations applied to these dialogs include: utterance segmentation (Section 4.1), syntactic annotation (Section 4.2), dialog act tagging and subtask segmentation. The former two were done in a domain-independent fashion while the latter two are domain-specific.

### 6.2. Features

In order to train the dialog act (DA) and subtask segmentation classifiers, we used *static* and *dynamic* features (Table 1). Static features are word $n$-gram features derived from the local context of the utterance being tagged (e.g. Speaker ID (agent versus customer), $n$-grams from current, previous utterance) while dynamic features are computed based on previous predictions (e.g. two previous subtask labels).

### 6.3. Dialog Act Labeling

As mentioned in Section 4, we used a domain-specific tag set designed to be useful for language generation. We have a total of 67 dialog act tags (DAMSL has 375, the Maptask scheme has 13). In Table 2, we illustrate some of the tags we used for annotation. We annotated 1864 clauses from 20 dialogs selected at random from our corpus. In our annotation, a single utterance may have multiple dialog act labels.

| Label Type | Features |
|---|---|
| Dialog Acts | Speaker, word bigrams from current/previous utterance |
| Subtask | Speaker, word trigrams from current utterance, previous utterance/turn, next utterance, unigram, trigram (two previous) subtask labels |

**Table 1**. Features used for the classifiers.

| **Type** | **Subtype** |
|---|---|
| Ask | Info |
| Explain | Catalog, CC_Related, Discount, Order_Info |
| | Order_Problem, Payment_Rel, Product_Info |
| | Promotions, Related_Offer, Shipping |
| Conversational | Ack, Goodbye, Hello, Help, Hold, YoureWelcome, Thanks, Yes, No, Ack, |
| | Repeat, Not(Information) |
| Request | Code, Order_Problem, Address, Catalog, |
| | CC_Related, Change_Order, Conf, Credit, |
| | Customer_Info, Info, Make_Order, Name, |
| | Order_Info, Order_Status, Payment_Rel, |
| | Phone_Number, Product_Info, Promotions, |
| | Shipping, Store_Info |
| YNQ | Address, Email, Info, Order_Info, |
| | Order_Status,Promotions, Related_Offer |

**Table 2**. Sample set of dialog labels used in our domain

Table 3 shows the error rates for dialog act labeling using word bigram features from the current and previous utterance. We compare error rates for our tag set against those of Switchboard-DAMSL and Maptask using the same features and the same classifier learner. The error rates are an average of ten-fold cross-validation execution. It is interesting to note that the error rate for our tag set is close to the error rate for DAMSL. Also, we suspect that the lack of improvement in the error rate for our tag set when including the previous utterance might be due to the small size of our annotated corpus (about 2K utterances for our domain as against about 20K utterances for Maptask and 200K utterances for DAMSL).

| Tagset | current utterance | + previous utterance |
|---|---|---|
| Catalog Domain | 38.8% | 38.8% |
| DAMSL | 39.3% | 37.2% |
| Maptask | 31.5% | 29.6% |

**Table 3**. Error rates in dialog act tagging

### 6.4. Subtask Segmentation and Labeling

For subtask labeling, we used a random partition of 864 dialogs as the training set and 51 dialogs as the test set. All the dialogs were annotated with subtask labels by hand. We used a set of 18 labels grouped as shown in Table 4.

Table 5 shows error rates on the test set for predicting refined subtask labels using word n-grams computed on different dialog contexts as features for training classifiers. It is clear from the table

| Category | Subtask Labels |
|---|---|
| 1 | opening, closing |
| 2 | contact-information, delivery-information, payment-information, shipping-address,summary |
| 3 | order-item, related-offer, order-problem discount, order-change, check-availability |
| 4 | call-forward, out-of-domain, misc-other, sub-call |

**Table 4**. Subtask label set

that the well-formedness constraint on the refined subtask labels vastly improves prediction accuracy.

| Tag Context | Utterance Context | | | | |
|---|---|---|---|---|---|
| | Current utt | +prev utt | +prev +next utt | +prev turn | +prev turn +next utt |
| Unigram | 32.6% (47.7%) | 28.6% (41.3%) | 27.1% (35.7%) | 27.7% (40.8%) | 26.8% (35.5%) |
| Trigram | 1.3% (16.2%) | 1.4% (14.7%) | 1.6% (8%) | 1.4% (16.0%) | 1.6% (8%) |

**Table 5**. Error rate for predicting the refined subtask labels. The error rates without the well-formedness constraint is shown in parenthesis.

The experiments reported in this section have been performed on transcribed speech. The audio for these dialogs, collected at a call center, was stored in a compressed format, so the speech recognition error rate is high. In future work, we will assess the performance of dialog structure prediction on recognized speech.

The research presented in this paper is but one step, albeit a crucial one, towards achieving the goal of inducing human-machine dialog systems using human-human dialogs. The dialog structure is necessary for language generation (predicting the agents' response) and dialog state specific text-to-speech synthesis. However, there are several challenging problems that remain to be addressed.

## 7. CONCLUSIONS

In this paper, we have presented an approach to structuring human-human dialogs with a long term goal of learning models for human-machine dialogs. We presented the problem of decomposing a dialog into subtask segments and develop a model for predicting and labeling these segments. We have evaluated the model on customer-agent dialogs from a catalog service domain and show the effectiveness of the well-formedness constraint in this task.

We view a dialog between two participants as an interleaved trajectory of utterances mediated by intermediate structures such as subtask and dialog act structure. In this paper, we have presented models to predict the intermediate structure. In on-going work, we are exploiting this structure in order to generate the next utterance.

## 8. ACKNOWLEDGMENTS

We thank Barbara Hollister and her team for annotating the dialogs for subtask structure. We also thank Alistair Conkie, Mazin Gilbert, Narendra Gupta, and Benjamin Stern for discussions during the course of this work.

### 9. REFERENCES

[1] G. Di Fabbrizio, G.Tur, and D.Hakkani-Tür, "Automated Wizard-of-Oz for spoken dialogue systems," in *Proceedings of Interspeech 2005*, Lisboa, Portugal, 2005.

[2] M. Gilbert, J.G. Wilpon, B.Stern, and G. Di Fabbrizio, "Intelligent virtual agents for contact center automation," in *IEEE Signal Processing Magazine*, Volume 22, Number 5, September 2005, pp.32-45.

[3] P. Carpenter et al., "Is this conversation on track?," in *Proceedings of Eurospeech*, 2001.

[4] J. Chen, S. Bangalore, O. Rambow, and M. Walker, "Towards automatic generation of natural language generation systems," in *Proceedings of COLING 2002*, 2002.

[5] S. Young, "Talking to machines (statistically speaking)," in *ICSLP*, Denver, Colorado, 2002.

[6] H. Hardy et al., "Data-driven strategies for an automated dialogue system," in *Proceedings of ACL 2004*, 2004.

[7] J. Polifroni, G. Chung, and S. Seneff, "Towards the automatic generation of mixed-initiative dialogue systems from web content," in *Proceedings of Eurospeech*, 2003.

[8] J. Chu-Carroll, "A statistical model for discourse act recognition in dialogue interactions," in *Proceedings of the AAAI spring symposium on Applying machine learning to discourse processing*, 1998.

[9] D. Jurafsky et al., "Switchboard discourse language modeling project report," Tech. Rep. Research Note 30, Center for Speech and Language Processing, Johns Hopkins University, Baltimore, MD, 1998.

[10] M. Mast et al., "Automatic classification of speech acts with semantic classification trees and polygrams," in *Proceedings of the IJCAI workshop on New approaches to learning for natural language processing*, 1995.

[11] K. Samuel, S. Carberry, and K. Vijay-Shanker, "Computing dialogue acts from features with transformation-based learning," in *Proceedings of the AAAI spring symposium on Applying machine learning to discourse processing*, 1998.

[12] A. Venkataraman, A. Stolcke, and E. Shriberg, "Automatic dialog act tagging with minimal supervision," in *Proceedings of the 9th Australian International Conference on Speech Science and Technology*, 2002.

[13] M. Core, "Analyzing and predicting patterns of DAMSL utterance tags," in *Proceedings of the AAAI spring symposium on Applying machine learning to discourse processing*, 1998.

[14] A. Stolcke et al., "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, no. 3, 2000.

[15] D.Surendran and G. Levow, "Combining text and prosodic features with support vector machines," in *Proceedings of ASRU 2005*, 2005.

[16] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proceedings of ACL 2003*, 2003.

[17] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proceedings of ICASSP*, 2005.

[18] G. A. Levow, "Prosodic cues to discourse segment boundaries in human-computer dialogue," in *Proceedings of SIGdial 2004*, 2004.

[19] J. Bear, J. Dowding, and E. Shriberg, "Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog," in *Proceedings of ACL 1992*, 1992.

[20] S. Seneff, "A relaxation method for understanding spontaneous speech utterances," in *Proceedings, Speech and Natural Language Workshop*, San Mateo, CA, 1992.

[21] P. Heeman, *Speech Repairs, Intonation Boundaries and Discourse Markers: Modeling Speakers' Utterances*, Ph.D. thesis, University of Rochester, 1997.

[22] T. Ruland, C.J. Rupp, J. Spilker, H. Weber, and K.L. Worm, "Making the most of multiplicity: A multi-parser multi-strategy architecture for the robust processing of spoken language," Tech. Rep., DFKI, Verbmobil report 230, 1998.

[23] M.G. Core and L.K. Schubert, "A syntactic framework for speech repairs and other disruptions," in *Proceedings of ACL 1999*, 1999, pp. 413–420.

[24] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, September 2000.

[25] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Proceedings of ICASSP*, Atlanta, GA, 1996.

[26] E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in *Proceedings of NAACL 2001*, Pittsburgh, PA, 2001.

[27] S. Bangalore and N. Gupta, "Extracting clauses in dialogue corpora : Application to spoken language understanding," *Journal Traitement Automatique des Langues (TAL)*, vol. 45, no. 2, 2004.

[28] S. Bangalore, "A lightweight dependency analyzer for partial parsing," *JNLE*, vol. 6, no. 2, pp. 113–138, 2000.

[29] S. Bangalore and A. K. Joshi, "Supertagging: An approach to almost parsing," *Computational Linguistics*, vol. 25, no. 2, 1999.

[30] A. K. Joshi, "An introduction to tree adjoining grammars," in *Mathematics of Language*, A. Manaster-Ramer, Ed. John Benjamins, Amsterdam, 1987.

[31] J. Carletta et al., "The reliability of a dialog structure coding scheme," *Computational Linguistics*, 1997.

[32] Y. Freund and R. E. Schapire, "Experiments with a new boosting alogrithm," in *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996, pp. 148–156.

[33] R.E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, December 1999.