DISCOURSE STRUCTURE FOR MULTI-SPEAKER SPONTANEOUS SPOKEN DIALOGS: INCORPORATING HEURISTICS INTO STOCHASTIC RTNS

Sheryl R. Young[***]
School of Computer Science, Carnegie Mellon University, Pgh, PA 15213

ABSTRACT

Abstract

In real spoken language applications, where speakers interact spontaneously, there is much seeming unpredictability that makes recognition difficult. Multi-speaker spontaneous dialog where two speakers interact verbally to cooperatively solve a mutual, shared problem is more varied than human-computer interactions. Spontaneous speech is not well structured, exhibiting mid-utterance corrections and restarts in utterances. Discourse contains digressions, clarifications, corrections and topic changes. But, multi-speaker discourse is even more varied, with initiative effects, speakers interacting, planning and responding. This makes it extremely difficult to develop grammars and language models with adequate coverage and reliable stochastic parameters. Perplexity increases and recognition degrades considerably vis-a-vis human-database dialog. In spite of all this, multi-speaker dialogs are structured and predictable when the discourse is appropriately modelled. We have developed heuristics to model spontaneous speech and multi-speaker dialogs [1, 2]. The underlying heuristics have been evaluated and shown to adequately and accurately predict discourse phenomena, as evaluated on a 10,000+ utterance corpus. Generally, the heuristics for computing discourse structure and deriving constraints from it are rule based. We have taken the rules and used them to develop a set of stochastic RTNs that capture both the rules and corpus probabilities. The resulting language model can be used predictively to dynamically generate stochastic utterance predictions or can be incorporated into any recognition/understanding system where a single prior state is maintained.

1. DISCOURSE STRUCTURE: INTRODUCTION

Discourse has been studied from the point of view of plan recognition [3, 4, 5, 1], speech acts, and domain independent properites of discourse structure [6, 7, 2]. Inferring a speakers' underlying plans, goals or intentions assists in interpretting what is stated and implied by an utterance. The advent of large corpora of real, naturally elicited dialogs from multiple application domains has facilitated research on spontaneous spoken discourse. They have enabled researchers to identify, characterize and model domain-independent discourse properties of goal directed dialogs. Spoken language systems [SLS], particularly those that process unconstrained, spontaneous dialogs, have further enabled discourse models to be empirically evaluated for thoroughness, coverage and explanatory utility. Because SLS' evaluate both a system's ability to recognize and understand spontaneous speech, they enable evaluation of both the predictive components of discourse models that specify how prior interaction may constrain future utterances and their contribution to understanding spoken input.

Discourse structure attempts to model a dialog by modelling the relations among utterances and the constraints implied by prior interaction upon the contents of a current utterance. Usually, this is distinct from models of the structure of an utterance, for example a model of mid-utterance corrections and restarts. Today, discourse structure is composed from the interaction of models of the application domain and the set of algorithms or heuristics that model the types of discourse level control actions that a speaker can initiate or perform at a specific point in time [3]. Models of the application domain and domain-independent heuristics for processing and applying these hierarchically organized domain plan models are an important feature of any discourse model. Initially, a domain model alone was used to help interpret spoken input and to constrain what a speaker may say [8, 1]. By tracking progress through verbal problem solving, systems dynamically applied constraints and used them to modify the SLS' language model and restrict recognition of meaningless and inappropriate word strings. However, speakers often diverge from strict verbal problem solving. To account for these behaviors, it is necessary to consider discourse actions. Discourse actions or discourse plans include continue-plan, or continue solving the application problem, but they are much broader and account for divergent speaker actions such as clarify what was said. The set of discourse actions observed in human-computer dialogs is somewhat distinct from the set observed in multi-speaker interactions. Specifically, multi-speaker interactions introduce variability associated with discussions of the plans speakers will execute to solve their assigned task. The second, most prevalent characteristic of multi-speaker interactions is the propose-counterpropose cycle observed when initiative for solving a problem is shared. This paper overviews domain and discourse plans, describes their implementation and our heuristics for inferring and exploiting multi-speaker discourse structure.

2. DOMAIN PLANS

Domain trees are hierarchical specifications of all the plans a speaker may pursue to solve any type of problem in a specific application domain. The application specific, generic domain tree encodes all the plans in a domain and the relations among them, (e.g. three top-level plans may be related by an "exclusive or" relation that indicates that only one of them may be pursued.) The domain plan tree is represented as a hierarchical "AND OR" tree. Sets of plans for solving specific types of problems are grouped together with an indication of whether they are mutually exclusive, alternative methods for solving (some part of) a problem or whether they are complementary. Plans are represented hierarchically, and the decomposition of any specific plan or plan step is specified in the domain tree, along with any required or preferred orderings among plan steps and an indication of whether a plan step is required, optional or conditionally required. The domain plan tree is designed to record dependencies among plan steps, to record pre-requisites and to propagate constraints among problem solving stages. Two types of constraints are automatically propagated: Constraints on potential plan steps are propagated by the "AND/OR" structure of the tree and by the "REQUIRED/OPTIONAL" field of each plan step. The second way constraints are automatically propagated among plan steps involves restrictions on how a plan step may be executed, or what objects can be involved in the action resulting from executing the plan step. These constraints are derived from computing contextual constraints upon what is available or reference given a specific set of active plan steps. The information available for reference is tied to the set of active domain plans and prevailing discourse plans. Presuming a "continue-plan" discourse plan, the objects available for reference vary when one or more active plans are completed or popped from the focus stack of active plans.

2.1. Representation during Processing

The system uses three data structures: a structured knowledge base of semantic information about the application domain, a domain plan tree and a current focus stack. The domain knowledge base and plan tree must be generated for each application. However, the algorithms responsible for plan inference and tracking, constraint propagation, general inferencing and for processing subdialogs, plan failures and other discourse plans are constant across applications. The basic idea underlying

---

*An earlier draft of this manuscript appeared in *Proceedings of the 1994 International Conference on Spoken Language Processing, ICSLP-94*, Kyoto, Japan, 1994.
**This research was sponsored by the National Science Foundation, under Grant No. IRI-9314992 and by the German Bundesminiterium fuer Forschung und Technologie

0-7803-2431-5/95 $4.00 © 1995 IEEE

the system is that by tracking all information communicated it is possible to infer speaker goals and plans and track problem solving progress. Tracking progress enables the system to predict the discourse actions that can be taken at each point in the dialog and constrain their content. These "predictions" can be used to better infer utterance meaning, to detect misrecognitions and to dynamically generate or constrain grammars for recognition or reprocessing misrecognized input [9].

The domain knowledge base represents all objects, attributes, values, plans, goals and the environment in which the actions and plans are executed. The knowledge base uses standard frame-based representations with tangled-inheritance networks and multiple relations among frames and frame slots to represent domain information. Different types of objects [actions & events vs. plans vs. goals vs objects, attributes and values] are represented distinctly and relations among these "types" are structured. For example, actions and events are indexed to the objects, attributes and values that may be involved in the action. Plans are related to goals via a "contribute-to" relation. Hence, an action can activate a plan which can in turn activate a goal in the knowledge base. These representations are used for inferencing and are designed to limit spurious inferences.

2.2. Traversal Heuristics

There are four basic rules used for traversing domain plan trees. Each of these is associated with significant decreases in the recognition language model's perplexity when used predictively [1, 8].

1. Once a plan step is complete, do not re-do it unless there is a planning failure.
2. Traverse the tree depth-wise. Whenever a decomposible node is encountered, all of its child plan steps or actions will be executed prior to any sibling nodes.
3. Propagate constraints as you progress, pruning paths through the domain plan tree when applicable.
4. Propage constraints as you progress by constraining how a domain plan step may be executed (e.g. by constraining applicable objects).

The idea is to trace through the domain plan tree hierarchically, pursuing each subtask in any requisite order until complete. Inapplicable subtrees (due to exclusive OR's) are pruned as they become obsolete. Constraints on how a plan step may be executed are also propagated as they are inferred or entailed by the discourse. The active and yet-to-be-completed tree nodes are used to predict what can come next.

Basically, during depth-wise traversal, nodes are first evaluated by whether they are decomposible or "leaf" nodes. When a decomposible node is processed, its preconditions and required and preferential orderings among its child steps are evaluated. Second, the required or optional status of the children nodes are evaluated or computed if necessary. Third, the node itself is marked "Active" and placed on the current focus stack. Then any mutually exclusive relations between nodes are considered in order to prune alternate paths through the domain tree and when necessary (e.g. the node is activated singly, without a child node or during prediction) constraints on which child plan steps are likely to be accessed next by the speaker are generated.

When a non-decomposible plan step is accessed, the following occur: First, the node is verified to ensure that it is being accessed or executed. Second, the node is marked "Active" if it is not already so marked and added to the focus stack, if its not in it. Next, the domain knowedge base is updated by activating any applicable domain goals. Finally, the system uses the information communicated to compute information that is available for reference and then propagates constraints throughout the domain plan tree by restricting the objects available for the plans to operate upon and by pruning alternate paths. In the absense of significant discourse plan interactions, as described below, when a node is complete, it is popped off the focus stack, marked "Complete" and control is passed back to the parent node.

3. DISCOURSE STRUCTURE

Domain plans, together with discourse plans can help interpret, predict or constrain the content of a speakers next utterance. The actual discourse structure that results from a dialog is a function of the interaction between discourse and domain plans. The heuristics in this section have been evaluated on two domains, on over 10,000 utterances, and found to significantly (50%+) reduce recognition errors and correctly predict content compared with normal recognition using statistical language models with perplexity 52-63.

3.1. Discourse Plans

In real applications, speakers frequently diverge from the task at hand by initiating various types of subdialogs. These behaviors range from clarifying information or requesting an explanation, to referring to the external environment, e.g. *"scroll the screen down"* or *"where can I buy a newspaper?"*. Based upon our corpora, we have developed a taxonomy of discourse plans. Our taxonomy is based upon the function plans serve in the larger interaction. There have been a few other attempts to categorize discourse plans [3, 7]. Ours are implemented in an operational SLS and affect processing primarily by modifying the domain plan tree and current focus stack. We have identified the following discourse plans: ***"continue"***, -continue working through the domain plan tree; ***"clarify"***, -ask for further information about objects, attributes or properties included in the last response or your last statement; ***"confirm"***, -confirm a set of facts communicated; and ***"repair/correct"***, -repair an unfeasible or failed domain problem solving plan or correct or modify information reviewed in a confirmation subdialog. In addition, we idenitified three types of discourse plans to change topic. The most common pursues an additional domain problem. A second discusses attributes of the interaction or of the plan step begin executed. For example a speaker asks the system to modify its display, as in *redisplay that last information.* The third type follows up on preceding day's interactions. In the multi-speaker corpora processed, extra-domain topic changes were rare (< .01%). Hence, we refer the reader to [2].

3.2. Processing and Representation

Only specific types of discourse plans can be initiated at any point in a dialog. We have developed and evaluated a set of rules that specify both when specific types of discourse plans can be initiated and what content they may address. When taken together with the algorithms for traversing domain plan trees, these two sets of rules circumscribe the set of behaviors observed in most goal directed spoken dialogs [2].

3.2.1. Clarifications

Clarificational subdialogs address the last interaction, clarifying either the speaker's last input or the response obtained. When a response is clarified, a speaker (or machine) can ask about the range of values acceptable, the meaning of an item contained in the response, an attribute of something included in the range of acceptable responses, or an attribute of one of the items named in the response. Often, clarifications are used to obtain information required for performing the task. Clarificational subdialogs can be nested and initiated by either conversational participant. They were prominent in our data. Their content does not affect the domain plan tree except in those cases where a speaker clarifies their input by asking an essentially different question (or giving a different response) that opens a different node in the domain plan tree. Normally, the clarification serves to provide additional information. We represent the content of clarifications and nested clarifications in a focus stack. The only information available for reference in a clarificational subdialog are the objects and attributes contained in the immediately preceding turn. Hence, when a nested clarification is initiated, the only information available for reference is from the parent clarification. No information from a clarification is propagated into the "main discourse" and the domain tree is not modified.

3.2.2. Confirmations

Confirmation subdialogs can only occur at the end of a subtask, or when a domain subtree is complete. For example, when two speakers schedule a meeting, the specifics may be confirmed following the negotiation. A confirmation can be initiated by either conversational participant to verify that they have understood correctly. Each item in the applicable completed unit can be verified and is available for reference. Confirmation are often followed by correction subdialogs that occur when the information to be confirmed is incorrect, incomplete or has been misunderstood.

We process confirmation subdialogs using the domain tree alone. All completed nodes in the applicable subtree are temporarily re-activated by a clarification until they have been discussed or the confirmation phase is complete. The following example illustrates a confirmation subdialog responded to with a correction subdialog (starred).

OK its set, 4 o'clock on the 10th
in room 4210, After we'll go eat.
No I forgot I have a 5pm meeting.
** Well, we could just skip dinner.
OK, that will work.

3.2.3. Corrections

Correction subdialogs are initiated under two conditions, when a confirmation fails or when there is a plan failure. These two are grouped together because both serve to re-activate a completed portion of the domain tree. Plan failures are easily detected, normally the user will encounter a null database response or be explicitly informed of a plan failure. For example, there will be insufficient resources in a resource limited problem solving domain. Plan failures occur when it is not possible to fulfill all requirements simultaneously. They are followed by a re-planning phase where speakers must prioritize goals and abandon one or more.

In our system, when a plan fails, all the specifications up to and including the point of the failure become re-activated in the domain tree. On the other hand, when a correction is initiated in response to a confirmation failure, the relevant nodes are already activated and only the node where the failure occurred and nodes that are causally related to it are available for reference and reexamination during the correction phase. Corrections only follow confirmations or plan failures.

4. MULTI-SPEAKER DISCOURSE

The multi-speaker discourse structure attributes discussed were derived from the analysis and processing of two spoken language applications: lunch-ordering dialogs, and a speech-to-speech translation system where two people attempt to schedule meetings verbally. Speakers are completely unconstrained in what they say and how they say it. In the meeting application, each speaker is randomly given a partially filled calendar sheet and both are assigned the goal of scheduling a specific type of meeting with a specified duration. Speakers communicate via a "fake" telephone that records all utterances. The data show that subdialogs and spontaneous phenomena occur with greater frequency in multiple-speaker goal directed dialogs than in human-computer interactions. This makes it far more difficult to develop grammars and language models with adequate coverage, increases the likelihood of encountering out-of-vocabulary words.

We investigated discourse structure attributes of multi-speaker dialogs. Our results indicated two types of structural phenomena are found in multi-speaker discourse: initiative-based effects and speaker meta-planning. First, we characterized the phenomena heuristically. Second, we incorporated the new heuristics into the existing rules for traversing domain plan trees and computing discourse structure. Third, we took the aggregate rules and transformed them into sets of recursive transition networks. Finally, we trained the transition probabilities in the RTNs using our training corpus. Our goals were to generate the most restrictive, yet comprehensive set of content predictions that could be used to detect and re-recognize misrecognized word strings, dynamically modify the recognizer's language model and constrain interpretation of an utterance.

4.1. Initiative and Discourse Structure

Much of the interchange among speakers can be accounted for by modelling ***initiative*** in problem solving. We observed that the joint plan developed by two interacting speakers was only furthered when a speaker took initiative. Initiative means the speaker proposed a solution (or partial solution) to part of the domain "problem". There are two general classes of non-initiative utterances: those utterances associated with a subdialog and utterances that were responses or reactions to the last "proposal" offered. The first class of non-initiative utterances, subdialogs, were discussed earlier. The heuristics governing when specific types of subdialogs can be introduced apply equally to multi-speaker and human-database dialogs. The second class of non-initiative utterances contain an indication, either implicit or explicit, that the proposal is unacceptable, acceptable or needs modification and, optionally, why this is so. Further, we observed that initiative could vary considerably among discourse participants. In some dialogs, both speakers took initiative. In others, one speaker was dominant, and responsible for the majority of solution plans and options. However, unlike human-database interactions where initiative rests almost exclusively with one conversational participant, rarely if ever was one speaker entirely reactive and passive while the other responsible for all problem solving options and solutions.

Because progress in solving an application problem only occurred when a discourse participant assumed initiative, we refined the most common discourse plan, ***"continue"***, subdividing it to represent initiative taking options. Patterns of responses to a proposed solution, or utterance where a speaker has taken initiative, interact with the algorithms for traversing the domain goal tree, reducing the set of possible next behaviors. For example, when a speaker initiates part of a solution in their utterance, the system activates the algorithms associated with the "continue" discourse-plan and evaluates whether a next plan step should be activated or if the speaker is proposing a new solution to an already active plan step. If the proposal follows a negative response to a prior proposal, the system knows that an already active domain plan tree node is being addressed. On the other hand, if the proposal follows a positive response to a prior proposal, the system knows that a new portion of the problem, or another variable is under discussion and a new domain plan tree node must be activated.

By modelling initiative and the possible responses available to a speaker after another speaker has taken initiative and generated or modified a proposal or potential solution, we are able to significantly streamline content predictions and account for a far greater portion of the variability evidenced in multi-speaker discourse. Specifically, by adding heuristics to account for speaker initiative and combining them with the other discourse and domain plan algorithms outlined above we can predict most all of the discourse phenomena observed in our multi-speaker application domains. For example, if some set of domain plans are active {A} and represented in the focus stack and speaker "B" rejects a proposed solution, speaker "A"'s options include: request further clarification from speaker "B", request help from speaker "B", or incorporate any additional constraints communicated and propose an alternate solution. In each potential case, the all heuristics result in a finite set of potential actions. Each of these sets of action alternatives can be represented as an RTN. The actual form an action (e.g. clarify what speaker "B" means) may take is dynamically scoped (e.g. request further information about when they are not available on Tuesday afternoon). However, the set of actions is finite and is broadly a function of the following list of questions:

- Did the last discourse action introduce or continue a subdialog?
- Did the last discourse action modify the current focus stack?
  - was a domain plan step completed?
- Did the last utterance evidence "initiative"?

Further, we can train probabilities for each of the transitions in the RTNs. transition nets that represent both the types of transitions that can be made following each utterance and the apriori probabilities associated with various discourse plans (e.g. clarification, confirmation, etc.), as illustrated in Figure .

4.2. Illustrative Example

To illustrate, consider the following scheduling exchange:

(1) A: How about next Monday at 10 am?
(2) B: Ohh hmm I have to meet with a student then
(3) A: What about later in the day -- say 5?
(4) B: No, uuum that's no good. Can you make 2 o'clock Wednesday?
(5) A: I have a class until 2:30 but after is alright
(6) B: That's o.k.

In the example speakers A and B are scheduling a meeting, pursing the subtask of finding a candidate meeting time (a date and time combination) when they are both free. There are no subdialogs in the above interaction. However, there is a clear pattern of propose, counterpropose, accept, reject. All proposals

contain an implicit request for agreement or approval, although occassionally, a speaker may explicitly ask for a reaction or a proposal. When a speaker introduces a proposal, the hearer either asks for a clarification or must (implicitly or explicitly) agree, disagree or modify the proposal. Disagreement may or may not be followed by an accompanying reason. In other words, upon hearing a proposal, unless the hearer introduces a subdialog, the hearer must respond and has the option of further proposing.

In the example, utterance (1) contains a proposal from speaker A to speaker B. In utterance (2), speaker B indirectly rejects the proposal but does not take the initiative and counterpropose. Utterances (3) and (4) illustrate A's modified proposal followed by B's indirect rejection and counterproposed solution. Utterance (5) illustrates a modification. Speaker A agrees that around 2 p.m. Wednesday is acceptable if 2:30 can be substituted for 2:00.

The discourse structure derived for multiple, interacting speakers is less structured than human - computer dialogs. Computer can occasionaly initiate solutions to problems or request clarifications on potentially misrecognized or misinterpretted input. In contrast, multi-speaker goal directed dialogs involve joint problem solving, where each participant is free to modify the underlying plans, take initiative for solving a portion of the problem or ask for assistance from the other person. Initiative makes it easier to determine when a plan step is complete and adds additional predictive structure within the most frequently executed discourse plan, **"continue-plan"**. Further, modeling initiative with a finite state transition network can even further assist basic recognition.

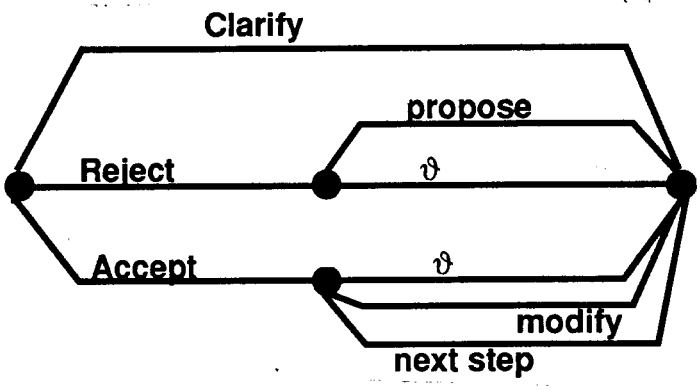

Figure 1: Transition Network Within a Plan Step

4.3. Meta-Planning Interactions

The second phenomena observed in multi-speaker interactions are meta-level discussions of how to solve a problem, on how to order plan steps and general problem solving constraints. For example:

Shouldn't we find a time before plan lunch?
Lets see when we can get the conference room before deciding when to schedule the group.

Meta-level discussions occur when transiting to a domain tree node that is decomposable, usually at major problem solving junctures. In many domains, meta-planning is rare. A low apriori probability can be reflected in the RTNs that capture the discourse and domain traversal rules. When the transitions are trained on corpora.

4.4. Summary of Multi-Speaker Algorithms

Domain trees are traversed depth-first. Two types of constraints are automatically computed and propagated: constraints that eliminate portions of the tree; and constraints that restrict the objects and attributes that an action may operate upon. The conjoined discourse and domain rules for traversing a domain tree are:

- Trace through domain plan tree depth-first, permitting meta-planing discussions at the root and major decomposible nodes.
- When a non-decomposible node is activated,
  - Eliminate "sibling" problem solving paths if an exclusive "OR" is present.
- Within a non-decomposible plan step, track initiative and permit clarifications of the last interaction.
  - If a clarification is introduced, temporarily suspend processing until its resolved, then proceed.
  - Following an initiative-taking utterance, expect a reply possibly followed by a modification, counter proposal or request.
  - When a suggestion is replied-to positively, evaluate if the plan step is complete.
    - If so, pop the focus stack, mark the node "Complete" and resume processing the parent node.
- When a decomposible node is complete, or when all required children are complete, look for confirmation subdialogs. If none, move up tree to parent node.
- When transiting to a node associated with different, modeled, environmental attributes than the prior plan steps, look for a topic change.
- When the enitre problem has been completed, look for a possible confirmation subdialog.

Discourse structure heuristics restrict what can be said when in a dialog. Multi-speaker dialogs exhibit significant initiative-based effects. By modelling these, we can explain much of the behavior observed in multi-person spontaneous interactions.

REFERENCES

1. Young, S.R., Hauptmann, A.G., Ward, W.H., Smith, E.T., Werner, P., "High Level Knowledge Sources in Usable Speech Recognition Systems", *Communications of the ACM*, Vol. 32, No. 2, 1989, pp. 183-194.
2. Young, S. R., "Dialog Structure and Plan Recognition in Spontaneous Spoken Interaction", *Proceedings of the European Conference on Speech Communication and Technology*, ESCA: Paris, London, 1993.
3. Litman, D. J. and Allen, J. F., "A Plan Recognition Model for Subdialogs in Conversation", *Cognitive Science*, Vol. 11, 1987, pp. 163-200.
4. Pollack, M., "Plans as Complex Mental Attitudes", in *Intentions in Communication*, Cohen, P.R., Morgan, J. and Pollack. M. E., eds., MIT Press, 1990.
5. Ferguson, G. and Allen, J. F., "Generic Plan Recognition for Dialogue Systems", *Proceedings of the DARPA Human Lanague Technology Conference*, 1993.
6. Grosz, B. J. and Sidner, C. L., "Attention, Intentions and the Structure of Discourse", *Computational Linguistics*, Vol. 12, 1986, pp. 175-204.
7. Allen, J. A., "Discourse Structure in the TRAINS Project", *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1991, pp. 325-330.
8. Matrouf, K., Gauvin, J.L., Neel, F., Mariani, J., "Adapting Probability-Transitions in DP Matching Process for an Oral Task-Oriented Dialogue", *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1990.
9. Young, S. R. and Ward, W. H., "Semantic and Pragmatically Based Re-Recognition of Spontaneous Speech", *Proceedings of the European Conference on Speech Communication and Technology*, ESCA: Paris. London. 1993.