Hierarchical Conversation Structure Prediction in Multi-Party Chat

Elijah Mayfield, David Adamson, and Carolyn Penstein Rosé
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{emayfiel, dadamson, cprose}@cs.cmu.edu

Abstract

Conversational practices do not occur at a single unit of analysis. To understand the interplay between social positioning, information sharing, and rhetorical strategy in language, various granularities are necessary. In this work we present a machine learning model for multi-party chat which predicts conversation structure across differing units of analysis. First, we mark sentence-level behavior using an information sharing annotation scheme. By taking advantage of Integer Linear Programming and a sociolinguistic framework, we enforce structural relationships between sentence-level annotations and sequences of interaction. Then, we show that clustering these sequences can effectively disentangle the threads of conversation. This model is highly accurate, performing near human accuracy, and performs analysis on-line, opening the door to real-time analysis of the discourse of conversation.

1 Introduction

When defining a unit of analysis for studying language, one size does not fit all. Part-of-speech tagging is performed on individual words in sequences, while parse trees represent language at the sentence level. Individual tasks can be performed at the lexical, sentence, or document level, or even to arbitrary length spans of text (Wiebe et al., 2005), while rhetorical patterns are annotated in a tree-like structure across sentences or paragraphs.

In dialogue, the most common unit of analysis is the utterance, usually through dialogue acts. Here, too, the issue of granularity and specificity of tags has been a persistent issue, along with the integration of larger discourse structure. Both theory-driven and empirical work has argued for a collapsing of annotations into fewer categories, based on either marking the dominant function of a given turn (Popescu-Belis, 2008) or identifying a single construct of interest and annotating only as necessary to distinguish that construct. We take the latter approach in this work, predicting conversation structure particularly as it relates to information sharing and authority in dialogue. We use systemic functional linguistics' Negotiation annotation scheme (Mayfield and Rosé, 2011) to identify utterances as either giving or receiving information. This annotation scheme is of particular interest because in addition to sentence-level annotation, well-defined sequences of interaction are incorporated into the annotation process. This sequential structure has been shown to be useful in secondary analysis of annotated data (Mayfield et al., 2012a), as well as providing structure which improves the accuracy of automated annotations.

This research introduces a model to predict information sharing tags and Negotiation sequence structure jointly with thread disentanglement. We show that performance can be improved using integer linear programming to enforce constraints on sequence structure. Structuring and annotation of conversation is available quickly and with comparatively little effort compared to manual annotation. Moreover, all of our results in this paper were obtained using data a real-world, chat-based internet community, with a mix of long-time expert and first-time

Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pages 60–69,
Seoul, South Korea, 5-6 July 2012.

novice users, showing that the model is robust to the challenges of messy data in natural environments.

The remainder of this paper is structured as follows. First, we review relevant work in annotation at the levels of utterance, sequence, and thread, and applications of each. We then introduce the domain of our data and the framework we use for annotation of conversation structure. In Section 4 we define a supervised, on-line machine learning model which performs this annotation and structuring across granularities. In Section 5, we evaluate this model and show that it approaches or matches human reliability on all tasks. We conclude with discussion of the utility of this conversation structuring algorithm for new analyses of conversation.

2 Related Work

Research on multi-party conversation structure is widely varied, due to the multifunctional nature of language. These structures have been used in diverse fields such as computer-supported collaborative work (O'Neill and Martin, 2003), dialogue systems (Bohus and Horvitz, 2011), and research on meetings (Renals et al., 2012). Much work in annotation has been inspired by speech act theory and dialogue acts (Traum, 1994; Shriberg et al., 2004), which operate primarily on the granularity of individual utterances. A challenge of tagging is the issue of specificity of tags, as previous work has shown that most utterances have multiple functions (Bunt, 2011). General tagsets have attempted to capture multi-functionality through independent dimensions which produce potentially millions of possible annotations, though in practice the number of variations remains in the hundreds (Jurafsky et al., 1998). Situated work has jointly modelled speech act and domain-specific topics (Laws et al., 2012).

Additional structure inspired by linguistics, such as adjacency pairs (Schegloff, 2007) or dialogue games (Carlson, 1983), has been used to build discourse relations between turns. This additional structure has been shown to improve performance of automated analysis (Poesio and Mikheev, 1998). Identification of this fine-grained structure of an interaction has been studied in prior work, with applications in agreement detection (Galley et al., 2004), addressee detection (op den Akker and Traum, 2009), and real-world applications, such as customer service conversations (Kim et al., 2010). Higher-order structure has also been explored in dialogue, from complex graph-like relations (Wolf and Gibson, 2005) to simpler segmentation-based approaches (Malioutov and Barzilay, 2006). Utterance level-tagging can take into account nearby structure, e.g. forward-looking and backward-looking functions in DAMSL (Core and Allen, 1997), while dialogue management systems in intelligent agents often have a plan unfolding over a whole dialogue (Ferguson and Allen, 1998).

In recent years, threading and maintaining of multiple "floors" has grown in popularity (Elsner and Charniak, 2010), especially in text-based media. This level of analysis is designed with the goal of separating out sub-conversations which are independently coherent. There is a common ground emerging in the thread detection literature on best practices for automated prediction. Early work viewed the problem as a time series analysis task (Bingham et al., 2003). Treating thread detection as a clustering problem, with lines representing instances, was given great attention in Shen et al. (2006). Subsequent researchers have treated the thread detection task as based in *discourse coherence*, and have pursued topic modelling (Adams, 2008) or entity reference grids (Elsner and Charniak, 2011) to define that concept of coherence.

Other work integrates local discourse structure with the topic-based threads of discourse. Ai et al. (2007) utilizes information state, a dialogue management component which loosely parallels thread structure, to improve dialogue act tagging. In the context of Twitter conversations, Ritter et al. (2010) suggests using dialogue act tags as a middle layer towards conversation reconstruction. Low-level structure between utterances has also been used as a foundation for modelling larger-level sociological phenomena between speakers in a dialogue, for instance, identifying leadership (Strzalkowski et al., 2011) and rapport between providers and patients in support groups (Ogura et al., 2008). These works have all pointed to the utility of incorporating sentence-level annotations, low-level interaction structure, and overarching themes into a unified system. To our knowledge, however, this work is the first to present a single system for simultaneous an-

Negotiation/Threads			Seq	User	Text
K2			1	C	[M], fast question, did your son have a biopsy?
K2			1	C	or does that happen when he comes home
	K1		2	V	i have 3 dogs.
	K1		2	V	man's best friend
	f		2	S	:-D
	o		2	C	and women
	K2		3	J	what kind of dogs????
		K1	4	C	[D], I keep seeing that you are typing and then it stops
		K2	5	C	how are you doing this week
	K1		3	V	the puppies are a maltese/yorkie mix and the full grown is a pomaranian/yorkie.
K1			1	M	No, he did not have a biopsy.
K1			1	M	The surgeon examined him and said that by feel, he did not think the lump was cancerous, and he should just wait until he got home.
f			1	C	that has to be very hard
		o	7	M	A question, however– [J], you would probably know.
		K2	7	M	He was told that they could not just do a needle biopsy, that he would have to remove the whole lump in order to tell if it was malignant.
	o		8	D	Yes.
	K1		8	D	I was waiting for [M] to answer.
		K1	7	J	That sounds odd to me

Table 1: An example excerpt with Negotiation labels, sequences, and threads structure (columns) annotated.

notation and structuring at all three levels.

3 Data and Annotation

Our data comes from the Cancer Support Community, which provides chatrooms, forums, and other resources for support groups for cancer patients. Each conversation took place in the context of a weekly meeting, with several patient participants as well as a professional therapist facilitating the discussion. In total, our annotated corpus consists of 45 conversations. This data was sampled from three group sizes - 15 conversations from small groups (2 patients, in addition to the trained facilitator), 15 from medium-sized groups (3-4 patients), and 15 from large groups (5 or more patients).

3.1 Annotation

Our data is annotated at the three levels of granularity described previously in this paper: *sentences*, *sequences*, and *threads*. In this section we define those annotations in greater detail. Sentence-level and sequence-level annotations were performed using the Negotiation framework from systemic functional linguistics (Martin and Rose, 2003). Once sequences were identified, those sequences were grouped together into threads based on shared topic.

We annotate our data using an adaptation of the Negotiation framework. This framework has been proven reliable and reproducible in previous work (Mayfield and Rosé, 2011). By assigning aggregate scores over a conversation, the framework also gives us a notion of *Authoritativeness*. This metric, defined later in Section 5, allows us to test whether automated codes faithfully reproduce human judgments of information sharing behavior at a per-user level. This metric has proven to be a statistically significant indicator of outcome variables in direction giving (Mayfield et al., 2011) and collaborative learning domains (Howley et al., 2011).

In particular, Negotiation labels define whether each speaker is a *source* or *recipient* of information. Our annotation scheme has four turn-level codes and a rigidly defined information sharing structure, rooted in sociolinguistic observation. We describe

each in detail below.

Sentences containing new information are marked as **K1**, as the speaker is the "primary knower," the source of information. These sentences can be general facts and world knowledge, but can also contain opinions, retelling of narrative, or other contextualized information, so long as the writer acts as the source of that information. Sentences requesting information, on the other hand, are marked **K2**, or "secondary knower," when the writer is signalling that they want information from other participants in the chat. This can be direct question asking, but can also include requests for elaboration or indirect illocutionary acts (e.g. *"I'd like to hear more."*). In addition to these primary moves, we also use a social feedback code, **f**, for sentences consisting of affective feedback or sentiment, but which do not contain new information. These moves can include emoticons, fixed expressions such as "good luck," or purely social banter. All other moves, such as typo correction or floor grabbing, are labelled **o**.

This annotation scheme is highly flexible and adaptive to new domains, and is not specific to medical topics or chatroom-based media. It also gives us a well-defined structure of an interaction: each sequence consists of exactly one primary knower (**K1**) move, which can consist of any number of primary knower sentences from a single speaker. If a **K2** move occurs in the sequence, it occurs before any **K1** moves. Feedback moves (**f**) may come at any time so long as the speaker is responding to another speaker in the same sequence. Sentences labeled **o** are idiosyncratic and may appear anywhere in a sequence. In section 4.3, we represent these constraints formally.

In addition to grouping sentences together into sequences structurally, we also group those sequences into threads. These threads are based on annotator judgement, but generally map to the idea that a single thread should be on a single theme, e.g. "handling visiting relatives at holidays." These threads are both intrinsically interesting for identifying the topics of a conversation, as well as being a useful preprocessing step for any additional, topic-based annotation that may be desired for later analysis.

We iteratively developed a coding manual for these layers of annotation; to test reliability at each iteration of instructions, two annotators each independently annotated one full conversation. Inter-annotator reliability is high for sentence-level annotation ($\kappa = 0.75$). Following Elsner and Charniak (2010), we use micro-averaged f-score to evaluate inter-rater agreement on higher-level structure. We find that inter-annotator agreement is high for both sequence-level structure ($f = 0.82$) and thread-level structure ($f = 0.80$). A detailed description of the annotation process is available in Mayfield et al. (2012b). After establishing reliability, our entire corpus was annotated by one human coder.

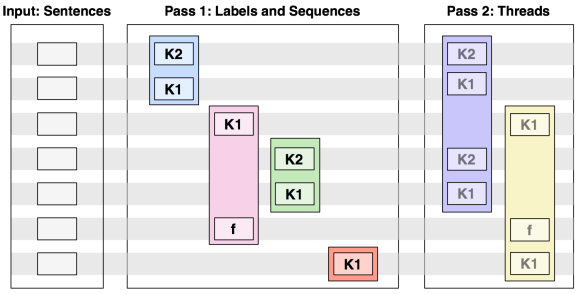

Figure 1: Structured output at each phase of the two-pass machine learning model. In pass one, utterances are grouped into sequences with organizational structure; the second pass groups sequences based on shared themes.

4 Conversation Structure Prediction

In previous work, the Negotiation framework has been automatically coded with high accuracy (Mayfield and Rosé, 2011). However, that work restricted the domain to a task-based, two-person dialogue, and structure was viewed as a segmentation, rather than threading, formulation. At each turn, a sequence could continue or a new sequence could begin.

Here, we extend this automated coding to larger groups speaking in unstructured, social chat, and we extend the structured element of this coding scheme to structure by sequence and thread. To our knowledge, this is also the first attempt to utilize functional sequences of interaction as a preprocessing step for thread disentanglement in chat. We now present a comprehensive machine learning model which annotates a conversation by utterance, groups utterances topics by local structure into sequences, and assigns sequences to threads.

4.1 On-Line Instance Creation

This is a two-pass algorithm. The first pass labels sentences and detects sequences, and the second pass groups these sequences into threads. We follow Shen et al. (2006) in treating the sequence detection problem as a single-pass clustering algorithm. Their model is equivalent to the **Previous Cluster** model described below, albeit with more complex features. In that work a threshold was defined in order for a new message to be added to an existing cluster. If that threshold is not passed, a new cluster is formed. Modelling the probability that a new cluster should be formed is similar to a context-sensitive threshold, and because we do not impose a hard threshold, we can pass the set of probabilities for cluster assignments to a structured prediction system.

4.2 Model Definitions

At its core, our model relies on three probabilistic classifiers. One of these models is a classification model, and the other two treat sequence and thread structure as clusters. All models use the LightSIDE (Mayfield and Rosé, 2010) with the LibLinear algorithm (Fan et al., 2008) for machine learning..

Negotiation Classifier (Neg)

The Negotiation model takes a single sentence as input. The output of this model is a distribution over the four possible sentence-level labels described in section 3.1. The set of features for this model consists of unigrams, bigrams, and part-of-speech bigrams. Part-of-speech tagging was performed using the Stanford tagger (Toutanova et al., 2003) within LightSIDE.

Cluster Classifiers (PC, NC)

We use two models of cluster assignment probability. The Previous Cluster (PC) classifier takes as input a previous set of sentences $C = \{c_1, c_2, \ldots, c_n\}$ and set of new sentences $N = \{N_1, N_2, \ldots, N_m\}$. To evaluate whether c^* should be added to this cluster, we train a binary probabilistic classifier that predicts the probability that the sentences in N belong to the same cluster as the sentences already in C. In the first pass, each input N to the PC classifier is a set containing a single sentence, and each C is the set of sentences in a previously-identified sequence. In the second pass, each N is a sequence as predicted by the first pass.

The PC model uses two features. The first is a time-based feature, measuring the amount of time that has elapsed between the last sentence in C and the first sentence in N. The time feature is represented differently between sequence prediction and thread prediction. Elsner and Charniak (2010) recommends using bucketed nominal values based on the log time, to group together very recent and very distant posts. We follow this for sequence prediction. Due to the more complex structure of the sequence grouping task in the second pass, we use a raw numeric time feature. The second feature is a coherence metric, the cosine similarity between the centroid of C and the centroid of N. We define the centroid based on TF-IDF weighted unigram vectors.

We impose a threshold after which previous clusters are no longer considered as options for the PC classifier. Because sequences are shorter than threads, we set these thresholds separately, at 90 seconds for sequences and 120 seconds for threads. Approximately 1% of correct assignments are impossible due to these thresholds.

The New Cluster (NC) classifier takes as input a set of sentences $n = \{n_1, n_2, \ldots, n_m\}$, and predicts the probability that a given sentence is initiating a new sequence (or, in the second pass, whether a given sequence is initiating a new thread). This model contains only unigram features.

At each sentence s we consider the set of possible previous cluster assignments $C = \{c_1, c_2, \ldots, c_n\}$, and define $p_{sc}(s, c)$ to be the probability that s will be assigned to cluster c. We define $p_{nc}(s) = \lambda_s NC(s)$. The addition of a weight parameter to the output of the NC classifier allows us to tune the likelihood of transitioning to a new cluster. This prediction structure is illustrated in Figure 2. In the first pass, these cluster probabilities are used in conjunction with the output of the Negotiation classifier to form a structured output; in the second pass, the maximum cluster probability is chosen.

4.3 Constraining Sequence Structure with ILP

In past work the Negotiation framework has benefited from enforced constraints of linguistically supported rules on sequence structure (Mayfield and

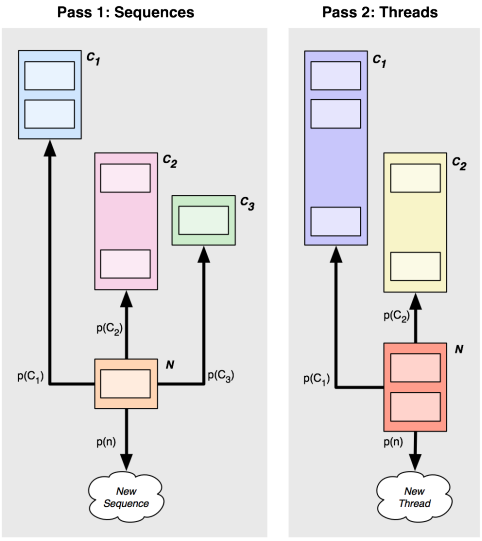

Figure 2: The output of the cluster classifier in either pass is a set of probabilities corresponding to possible cluster assignments, including that of creating a new cluster. In the second pass, the input is a set of sentences (a sequence) rather than a single sentence, and output assignments are to threads rather than sequences.

Rosé, 2011). Constraints on the structure of annotations are easily defined using Integer Linear Programming. Recent work has used boolean logic (Chang et al., 2008) to allow intuitive rules about a domain to be enforced at classification time. ILP inference was performed using Learning-Based Java (Rizzolo and Roth, 2010).

First, we define the classification task. Optimization is performed given the set of probabilities $\mathcal{N}(s)$ as the distribution output of the Neg classifier given sentence s as input, and the set of probabilities $\mathcal{C}(s) = p_{nc}(s) \cup p_{sc}(s, c), \forall c \in C$. Instance classification requires maximizing the objective function:

$$\arg\max_{n \in \mathcal{N}(s), c \in \mathcal{C}(s)} \quad n + c$$

We impose constraints on sequence prediction. If the most likely output from this function assigns a label that is incompatible with the assigned sequence, either the label is changed or a new sequence is assigned so that constraints are met. For each constraint, we give the intuition from section 3.1, followed by our formulation of that constraint. u_s is shorthand for the user who wrote sentence s; n_s is shorthand for a proposed Negotiation label of sentence s; while c_s is a proposed sequence assignment for s, c' is shorthand for assignment to a new sequence, and $S_c = \{(n_{c,1}, u_{c,1}), (n_{c,2}, u_{c,2}), \ldots, (n_{c,k}, u_{c,k})\}$ is the set of Negotiation labels n and users u associated with sentences $(s_{c,1} \ldots s_{c,k})$ already in sequence c.

1. **K2** moves, if any, occur before **K1** moves.
   $((c_s = c) \wedge (n_s = \textbf{K2}))$
   $\rightarrow (\nexists i \in S_c \; s.t. \; n_{c,i} = \textbf{K1})$

2. **f** moves may occur at any time but must be responding to a different speaker in the same sequence.
   $((c_s = c) \wedge (n_s = \textbf{f}))$
   $\rightarrow (\exists i \in S_c \; s.t. \; u_{c,i} \neq u_s)$

3. Functionally, therefore, **f** moves may not initiate a sequence).
   $(c_s = c') \rightarrow (n_s \neq \textbf{f})$

4. Speakers do not respond to their own requests for information (the speakers of **K2** and **K1** moves in the same sequence must be different).
   $((c_s = c) \wedge (n_s = \textbf{K1}))$
   $\rightarrow (\forall i \in S_c, ((n_{c,i} = \textbf{K2}) \rightarrow (u_{c,i} \neq u_s)))$

5. Each sequence consists of at most one continuous series of **K1** moves from the same speaker.
   $(c_s = c) \rightarrow ((\exists i \in S_c \; s.t. \; (n_{c,i} = \textbf{K1}))$
   $\rightarrow (\; (u_{c,i} = u_s) \wedge (\forall j > i,$
   $(u_{c,j} = u_s) \wedge (n_{c,i} = \textbf{K1}))\;)$

Human annotators treated these rules as hard constraints, as the classifier does. In circumstances where these rules would be broken (for instance, due to barge-in or trailing off), a new sequence begins.

5 Evaluation

5.1 Methods

To evaluate the performance of this model, we wish to know how it replicates human annotation at each granularity. For Negotiation labels, agreement is measured by terms of absolute accuracy and kappa agreement above chance. We also include a measure of aggregate information sharing behavior per user. This score, which we term *Information Authoritativeness (Auth)*, is defined per user as the percentage

of their contentful sentences (K1 or K2) which were giving information (K1). To measure performance on this measure, we measure the r^2 coefficient between user authoritativeness scores calculated from the predicted labels compared to actual labels. This is equivalent to measuring the variance explained by our model, where each data point represents a single user's predicted and actual authoritativeness scores over the course of a whole conversation ($n = 215$).

Sequence and thread agreement is evaluated by micro-averaged f-score (MAF), defined in prior work for a gold sequence i with size n_i, and a proposed sequence j with size n_j, based on precision and recall metrics:

$$P = \frac{n_{ij}}{n_j} \qquad R = \frac{n_{ij}}{n_i} \qquad F(i,j) = \frac{2 \times P \times R}{P+R}$$

MAF across an entire conversation is then a weighted sum of f-scores across all sequences[1]:

$$MAF = \sum_i \frac{n_i}{n} \max_j F(i,j)$$

We implemented multiple baselines to test whether our methods improve upon simpler approaches. For sequence and thread prediction, we implement the following baselines. **Speaker Shift** predicts a new thread every time a new writer adds a line to the chat. **Turn Windows** predicts a new sequence or thread after every n turns. **Pause Length** predicts a new sequence or thread every time that a gap of n seconds has occurred between lines of chat. For both of the previous two baselines, we vary the parameter n to optimize performance and provide a challenging baseline. None of these models use any features or constraints, and are based on heuristics. To compare to our model, we present both an **Unconstrained** model, which uses machine learning and does not impose sequence constraints from Section 4.3, as well as our full **Constrained** model.

Evaluation is performed using 15-fold cross-validation. In each fold, one small, one medium, and one large conversation are held out as a test set, and classifiers are trained on the remaining 42 conversations. Significance is evaluated using a paired student's t-test per conversation ($n = 45$).

Sentence-Level (Human $\kappa = 0.75$)			
Model	Accuracy	κ	Auth r^2
Unconstrained	.7736	.5870	**.7498**
Constrained	**.7777**	**.5961**	.7355
Sequence-Level (Human MAF $= 0.82$)			
Model	Precision	Recall	MAF
Speaker Shift	.7178	.5140	.5991
Turn Windows	.7207	.6233	.6685
Pause Length	.8479	.6582	.7411
Unconstrained	.7909	.7068	.7465
Constrained	**.8557**	**.7116**	**.7770**
Thread-Level (Human MAF $= 0.80$)			
Model	Precision	Recall	MAF
Turn Windows	.5994	**.7173**	**.6531**
Pause Length	.6145	.6316	.6229
Unconstrained	**.7132**	.5781	.6386
Constrained	.6805	.6024	.6391

Table 2: Tuned optimal annotation performances of baseline heuristics compared to our machine learning model.

5.2 Results

Results of experimentation show that all models are highly accurate in their respective tasks. With sentence-level annotation approaching 0.6 κ, the output of the model is reliable enough to allow automatically annotated data to be included reliably alongside human annotations. Performance for sequence-based modelling is even stronger, with no statistically significant difference in f-score between the machine learning model and human agreement.

Table 2 reports our best results after tuning to maximize performance of baseline models, our original machine learning model, and the model with ILP constraints enforced between Negotiation labels and sequence. In all three cases, we see machine performance approaching, but not matching, human agreement. Incorporating ILP constraints improves per-sentence Negotiation label classification by a small but significant amount ($p < .001$).

Clustering performance is highly robust, as demonstrated in Figure 3, which shows the effect of changing window sizes and pause lengths and values of λ_s for machine learned models. Our thread disentanglement performance matches our baselines, and

[1]This metric extends identically to a gold thread i and proposed thread j.

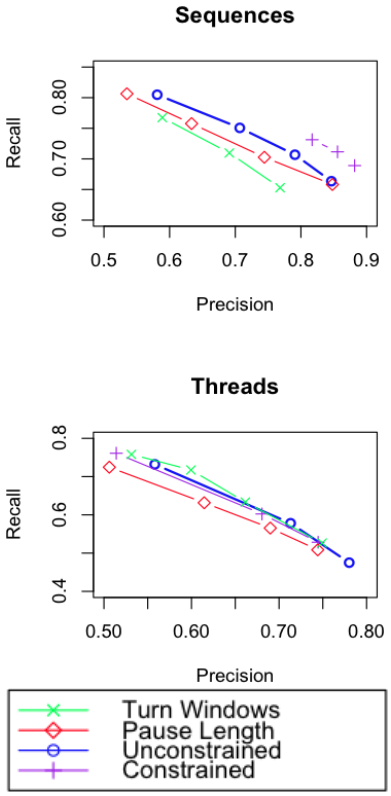

Figure 3: Parameter sensitivity on sequence-level (top) and thread-level (bottom) annotation models.

is in line with heuristic-based assignments from Elsner and Charniak (2010). In sequence clustering, we observe improvement across all metrics. The Constrained model achieves a higher f-score than all other models ($p < 0.0001$). We determine through a two-tailed confidence interval that sequence clustering performance is statistically indistinguishable from human annotation ($p < 0.05$).

Error analysis suggests that the constraints are too punishing on the most constrained labels, **K2** and **f**. The differences in performance between constrained and unconstrained models is largely due to higher recall for both **K1** and **o** move prediction, while recall for **K2** and **f** moves lowered slightly. One possibility for future work may include compensating for this by artificially inflating the likelihood of highly-constrained Negotiation labels. Additionally, we see that the most common mistakes involve distinguishing between **K1** and **f** moves. While many **f** moves are obviously non-content-bearing (*"Wow, what fun!"*), others, especially those based in humor, may look grammatical and contentful (*"We've got to stop meeting this way."*). Better detection of humor and a more well-defined definition of what information is being shared will improve this aspect of the model. Overall, these errors do not limit the efficacy of the model for enabling future analysis.

6 Conclusion and Future Work

This work has presented a unified machine learning model for annotating information sharing acts on a sentence-by-sentence granularity; grouping sequences of sentences based on functional structure; and then grouping those sequences into topic-based threads. The model performs at a high accuracy, approaching human agreement at the sentence and thread level. Thread-level accuracy matched but did not exceed simpler baselines, suggesting that this model could benefit from a more elaborate representation of coherence and topic. At the level of sequences, the model performs statistically the same as human annotation.

The automatic annotation and structuring of dialogue that this model performs is a vital preprocessing task to organize and structure conversational data in numerous domains. Our model allows researchers to abstract away from vocabulary-based approaches, instead working with interaction-level units of analysis. This is especially important in the context of interdisciplinary research, where other representations may be overly specialized towards one task, and vocabulary may differ for spurious reasons across populations and cultures.

Our evaluation was performed on a noisy, real-world chatroom corpus, and still performed very accurately. Coherent interfacing between granularities of analysis is always a challenge. Segmentation, tokenization, and overlapping or inconsistent structured output are nontrivial problems. By incorporating sentence-level annotation, discourse-level sequence structure, and topical thread disentanglement into a single model, we have shown one way to reduce or eliminate this interfacing burden and allow greater structural awareness in real-world systems. Future work will improve this model's accuracy further, test its generality in new domains such as spoken multi-party interactions, and evaluate its usefulness in imposing structure for secondary analysis.

Acknowledgments

The research reported here was supported by National Science Foundation grant IIS-0968485, Office of Naval Research grant N000141110221, and in part by the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation grant SBE-0836012.

References

Paige H. Adams. 2008. *Conversation Thread Extraction and Topic Detection in Text-based Chat*. Ph.D. thesis.

Hua Ai, Antonio Roque, Anton Leuski, and David Traum. 2007. Using information state to improve dialogue move identification in a spoken dialogue system. In *Proceedings of Interspeech*.

Ella Bingham, Ata Kaban, and Mark Girolami. 2003. Topic identification in dynamical text by complexity pursuit. In *Neural Processing Letters*.

Dan Bohus and Eric Horvitz. 2011. Multiparty turn taking in situated dialog. In *Procedings of SIGDIAL*.

Harry Bunt. 2011. Multifunctionality in dialogue. In *Computer Speech and Language*.

Lauri Carlson. 1983. *Dialogue Games: An Approach to Discourse Analysis*. Massachussetts Institute of Technology.

Ming-Wei Chang, Lev Ratinov, Nicholas Rizzolo, and Dan Roth. 2008. Learning and inference with constraints. In *Proceedings of the Association for the Advancement of Artificial Intelligence*.

Mark G Core and James F Allen. 1997. Coding dialogs with the damsl annotation scheme. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*.

Micha Elsner and Eugene Charniak. 2010. Disentangling chat. *Computational Linguistics*.

Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of the Association for Computational Linguistics*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification.

George Ferguson and James Allen. 1998. Trips: An integrated intelligent problem-solving assistant. In *Proceedings of AAAI*.

Michael Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of ACL*.

Iris Howley, Elijah Mayfield, and Carolyn Penstein Rosé. 2011. Missing something? authority in collaborative learning. In *Proceedings of Computer Supported Collaborative Learning*.

Daniel Jurafsky, Rebecca Bates, Noah Coccaro, Rachel Martin, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Andreas Stolcke, Paul Taylor, and Carol Van Ess-Dykema. 1998. Switchboard discourse language modelling final report. Technical report.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

M Barton Laws, Mary Catherine Beach, Yoojin Lee, William H. Rogers, Somnath Saha, P Todd Korthuis, Victoria Sharp, and Ira B Wilson. 2012. Provider-patient adherence dialogue in hiv care: Results of a multisite study. *AIDS Behavior*.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of ACL/COLING*.

J.R. Martin and David Rose. 2003. *Working with Discourse: Meaning Beyond the Clause*. Continuum.

Elijah Mayfield and Carolyn Penstein Rosé. 2010. An interactive tool for supporting error analysis for text mining. In *NAACL Demonstration Session*.

Elijah Mayfield and Carolyn Penstein Rosé. 2011. Recognizing authority in dialogue with an integer linear programming constrained model. In *Proceedings of Association for Computational Linguistics*.

Elijah Mayfield, Michael Garbus, David Adamson, and Carolyn Penstein Rosé. 2011. Data-driven interaction patterns: Authority and information sharing in dialogue. In *Proceedings of AAAI Fall Symposium on Building Common Ground with Intelligent Agents*.

Elijah Mayfield, David Adamson, Alexander I Rudnicky, and Carolyn Penstein Rosé. 2012a. Computational representations of discourse practices across populations in task-based dialogue. In *Proceedings of the International Conference on Intercultural Collaboration*.

Elijah Mayfield, Miaomiao Wen, Mitch Golant, and Carolyn Penstein Rosé. 2012b. Discovering habits of effective online support group chatrooms. In *ACM Conference on Supporting Group Work*.

Kanayo Ogura, Takashi Kusumi, and Asako Miura. 2008. Analysis of community development using chat logs: A virtual support group of cancer patients. In *Proceedings of the IEEE Symposium on Universal Communication*.

Jacki O'Neill and David Martin. 2003. Text chat in action. In *Proceedings of the International Conference on Supporting Group Work*.

Rieks op den Akker and David Traum. 2009. A comparison of addressee detection methods for multiparty conversations. In *Workshop on the Semantics and Pragmatics of Dialogue*.

Massimo Poesio and Andrei Mikheev. 1998. The predictive power of game structure in dialogue act recognition: Experimental results using maximum entropy estimation. In *Proceedings of the International Conference on Spoken Language Processing*.

Andrei Popescu-Belis. 2008. Dimensionality of dialogue act tagsets: An empirical analysis of large corpora. In *Language Resources and Evaluation*.

Steve Renals, Hervé Bourlard, Jean Carletta, and Andrei Popescu-Belis. 2012. *Multimodal Signal Processing: Human Interactions in Meetings*.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Proceedings of NAACL*.

Nicholas Rizzolo and Dan Roth. 2010. Learning based java for rapid development of nlp systems. In *Proceedings of the International Conference on Language Resources and Evaluation*.

E. Schegloff. 2007. *Sequence organization in interaction: A primer in conversation analysis*. Cambridge University Press.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of SIGIR*.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The icsi meeting recorder dialog act (mrda) corpus. In *Proceedings of SIGDIAL*.

Tomek Strzalkowski, George Aaron Broadwell, Jennifer Stromer-Galley, Samira Shaikh, Ting Liu, and Sarah Taylor. 2011. Modeling socio-cultural phenomena in online multi-party discourse. In *AAAI Workshop on Analyzing Microtext*.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL*.

David Traum. 1994. *A computational theory of grounding in natural language conversation*. Ph.D. thesis.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*.

Florian Wolf and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational Linguistics*.