"I Object!" Modeling Latent Pragmatic Effects in Courtroom Dialogues

Dan Goldwasser
University of Maryland
Institute for Advanced Computer Studies
College Park, MD , USA
goldwas1@umiacs.edu

Hal Daumé III
Department of Computer Science
University of Maryland
College Park, MD , USA
hal@cs.umd.edu

Abstract

Understanding the actionable outcomes of a dialogue requires effectively modeling situational roles of dialogue participants, the structure of the dialogue and the relevance of each utterance to an eventual action. We develop a latent-variable model that can capture these notions and apply it in the context of courtroom dialogues, in which the *objection* speech act is used as binary supervision to drive the learning process. We demonstrate quantitatively and qualitatively that our model is able to uncover natural discourse structure from this distant supervision.

1 Introduction

Many dialogues lead to decisions and actions. The participants in such dialogues each come with their own goals and agendas, their own perspectives on dialogue topics, and their own ways of interacting with others. Understanding the actionable results of a dialogue requires accurately modeling both the content of dialogue utterances, as well as the relevant features of its participants.

In this work, we devise a discriminative latent variable model that is able to capture the overall structure of a dialogue as relevant to specific acts that occur as a result of that dialogue. We aim to model both the *relevance* of preceding dialogue to particular action, as well as a binary structured relationship among utterances, while taking into account the pragmatic effect introduced by the different speakers' perspectives.

We focus on a particular domain of dialogue: courtroom transcripts. This domain has the advantage that while its range of topics can be broad, the roles of participants are relatively well-defined. Courtroom dialogues also contain a specialized speech act: **the objection**.

In real court settings (as opposed to fictionalized courts), an objection is a decision made by the party opposing the side holding the floor, to *interrupt* the flow of the courtroom discussion. While motivation behind taking this decision can stem from different reasons, it is typically an indication that a particular pragmatic *rule* has been broken. The key insight is that objections are sustained when a nuanced rule of court is being violated: for instance, the *argumentative* objection is "raised in response to a question which prompts a witness to *draw inferences* from facts of the case"[1], as opposed to the witness stating concrete facts.

The objectionable aspects of the preceding dialogue can be identified by a well-trained person; however these aspects are quite subtle to a computational model. In this work we take a first step toward addressing this problem computationally, and focus on identifying the key properties of dialogue interactions relevant for learning to identify and classify courtroom objections.

Our technical goal is to drive latent learning of dialogue structure based on a combination of raw input and pragmatic binary supervision. The binary supervision we use is derived from *objection* speech acts appearing in the dialogue (described in Section 2.1).

We are primarily interested in constructing a representation suitable for learning the challenging task of identifying objections in courtroom proceedings (Figure 1 provides an example).

In order to make classifications reliably, a deeper representation of the dialogue is required.

[1] Source: Wikipedia, July 2011, http://en.wikipedia.org/wiki/Argumentative.

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 655–663,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

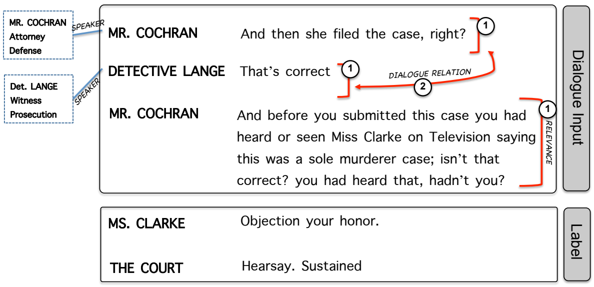


Figure 1: **Moving from raw text to a meaningful representation.** The raw textual representation hides complex interactions, relevant for understanding the dialogue flow and making decisions over it. We break the text into dialogue turns, each associated with a *speaker*, explicitly annotated with their role and side in the court case. Judgements of the relevance of each dialogue component for the classification task, produce a more accurate representation of the dialogue which is easier to learn. These judgments can be over individual sentences (①) or over pairs of sentences across different turns (②), which represent relevant information flow. The parameters required for making these judgements are obtained via interaction with the learning process. We explain these consideration and the construction stages in Section 2.

Notation	Explanation
$\mathbf{x}$	Input dialogue
$\mathbf{x}_{Sit}$	Situated dialogue
$\mathbf{h}$	Latent structure variables
t	Dialogue turn
$t.speaker.\{name,role,side\}$	Speaker information
$t.text$	Text in a dialogue turn
$t.s_i.\{text,type,subj,entities\}$	Sentence level information

Table 1: Notation Summary

Our model makes use of three conceptually different components capturing linguistic and pragmatic considerations and their relevance in the context of the dialogue structure.

Our linguistic model focuses on enriching a lexical representation of the dialogue utterances using linguistic resources capturing biased language use, such as subjective speech, expressions of sentiment, intensifiers and hedges. For example, the phrase *"So he was driving negligently?"* is an *argumentative* expression, as it requires the witness to draw inferences, rather than describe facts. Identifying the use of biased language in this phrase can help capture this objectionable aspect. In addition, we use a named entity recognizer, as we observe that relevant entity mentions provide a good indication of the dialogue focus. We refer the reader to Section 2.2 for further explanations.

The surface representation of dialogue turns hides the complex interactions between its participants. These interactions are driven by their agendas and roles in the trial. Understanding the lexical cues in this context requires *situating* the dialogue in the context of the court case. We condition the lexical representation of a turn on its speaker, the speaker's role and side in the trial, thus allowing the model to capture the relevant pragmatic influences introduced by the different speakers.

Next, a discriminative latent variable model learns a structured representation of the dialogue that is useful in making high-level semantic/pragmatic predictions (section 2.3). The latent variable model consists of two types of variables.

The first type of latent variable aims to identify content relevant for the objection identification decision. To this end, it determines the relevance of individual sentences to the classification decision, based on properties such as the lexical items appearing in the sentence, the sentence type, and expressions of subjectivity. *The second latent variable type focuses on the information flow* between speakers. It identifies relevant dialogue relations between turns. This decision is made by constructing a joint representation of two sentences, across different dialogue turns, capturing responses to questions and joining lexical items appearing in factual sentences across different turns.

Both dialogue aspects are formalized as latent variables, trained jointly with the final classification task using automatically extracted supervision. In Sec. 3 we describe the learning process.

We evaluate our approach over short dialogue snippets extracted from the O.J. Simpson murder trial. Our experiments evaluate the contribution of the different aspects of our system, showing that the dialogue representation determined by our latent model results in considerable improvements.

Our evaluation process considers several different views of the extracted data. Interestingly, despite the formal definitions of objections, the majority of objections are raised without justification (and are subsequently overruled), typically for the purpose of interrupting the opposing side when controversial topics are touched upon. Our experiments analyze the differences between sustained and overruled objections and show that sustained objections are easier to detect. We describe our experiments in section 4.

2 Dialogue Structure Modeling

Making predictions in such a complex domain requires a rich representation, capturing the interactions between different participants, the tone of conversation, understanding of controversial issues presented during the trial, and their different interpretations by either side in the trial. Obtaining this information manually is a labor intensive task, furthermore, its subjective nature allows for many different interpretations of the interactions leading to the objection.

Our approach, therefore, tries to avoid this difficulty by using a data-driven approach to learn the correct representation for the input, jointly with learning to classify correctly. Our representation transforms the raw input, dialogue snippets extracted automatically from court proceedings, into meaningful interactions between dialogue participants using a set of variables to determine the relevant parts of the dialogue and the relations between them. We inform these decisions using generic resources providing *linguistic* knowledge and *pragmatic* information, situating the dialogue in the context of the trial.

In this section we explain this process, starting from the automatic process of extracting examples (Section 2.1), the linguistic knowledge resources and pragmatic information used (Section 2.2), we summarize the notation used to describe the dialogue and its properties in Table 1. We formulate the inference process, identifying the meaningful interactions for prediction as an Integer Linear Programming (ILP) optimization problem (Section 2.3). The objective function used when solving this optimization problem is learned from data, by treating these decisions as latent variables during learning. We explain the learning process and its interaction with inference in Section 3.

2.1 Mining Courtroom Proceedings

The first step in forming our dataset consists of collecting a large set of relevant courtroom dialogue snippets. First, we look for textual occurrences of *objections* in the trial transcript by looking for *sustain* or *overrule* word lemma patterns, attributed to the judge. We treat the judge ruling turn and the one preceding it as sources of supervision, from which an indication of an objection, its type and sustained/overruled ruling, can be extracted. [2]

We treat the preceding dialogue as the cause for the objection, which could appear in any of the previous turns (or sequence of several turns intervening).We consider the previous *n=6* turns as the context potentially relevant for the decision and let the latent variable model *learn* which aspects of the context are actually relevant.

2.2 Linguistic and Pragmatic Information

Objection decisions often rely on semantic and pragmatic patterns which are not explicitly encoded. Rather than annotating these manually, we use generic resources to enrich our representation.

We make a conceptual distinction between two types of resources. The first, an array of linguistic resources, which provides us an indication of structure, topics of controversy, and the sentiment and tone of language used in the dialogue.

The second captures pragmatic considerations by situating the dialogue utterances in the context of the courtroom. Each utterance is attributed to a speaker, thus capturing meaningful patterns specific to individual speakers.

Linguistic Resources (1) **Named Entities** provide strong indications of the topics discussed in the dialogue and help uncover relevant utterances, such as ones making claims associating individuals with locations. We use the Named Entity Recognizer (NER) described in (Finkel et al., 2005) to identify this information.
(2) **Subjective and Biased Language** Equally important to understanding the topics of conversation is the way they are discussed. Expressions of subjectivity and sentiment are useful linguistic tools for changing the tone of the dialogue and are likely to attract opposition. We use several resources to capture this information. We use a lexicon of subjective and positive/negative sentiment expressions (Riloff and Wiebe, 2003). This resource can help identify subjective statements attempting to bias the discussion (e.g., *"So he was driving* ***negligently?****"*)

We use a list of hedges and boosters (Hyland, 2005). This resource can potentially allow the model to identify evasive (*"I* ***might*** *have seen him"*) and (overly) confident responses (*"I am* ***absolutely sure*** *that I have seen him"*).

We use a lexicon of biased language provided by (Recasens et al., 2013), this lexicon extracted from Wikipedia edits consists of words indicative of bias, for example in an attempt to *frame* the facts raised in the discussion according to one of the viewpoints (*"The* ***death*** *of Nicolle Simposon"* vs. *"The* ***murder*** *of Nicolle Simposon"*).

Finally we use a Patient Polarity Verbs lexicon (Goyal et al., 2010). This lexicon consists

[2] In 4 we provide details about the extracted dataset and its distribution according to types.

of verbs in which the agent performs an action with a positive (*"He **donated** money to the foundation"*) or negative (*"He **stole** money from the foundation"*) consequence to the patient.

(3) **Sentence Segmentation** Many turns discuss multiple topics, some more relevant than others. In order to accommodate a finer-grained analysis, we segment each turn into its sentences. Each sentence is associated with a label, taken from a small set of generic labels. Labels include FORMALITY (e.g., a witness being sworn in), QUESTION, RESPONSE (which could be either POSITIVE or NEGATIVE) and a general STATEMENT[3].

Capturing Pragmatic Effects We observe that in the context of a courtroom discussion, utterance interpretation (and subsequent dialogue actions) is conditioned to a large extent on the speaker's motivation and goals rather than in isolation. We capture this information by explicitly associating relevant characteristics of the speakers involved in the dialogue with their utterances. We use the list of *actors* which appear in the trial transcripts, and associate each turn with a speaker, their role in the trial and the side they represent. We augment the lexical turn representation with this information (see Sec. 2.3.4).

2.3 Identifying Relevant Interactions using Constrained Optimization

In this section we take the next step towards a meaningful representation by trying to identify dialogue content and information flow relevant for objection identification. Since this information is not pre-annotated, we allow it to be learned as latent variables. These latent variables act as boolean indicator variables, which determine how each dialogue input example will be represented.

This process consists of two conceptual stages, corresponding to two types of boolean variables: (1) relevant utterances are identified; (2) meaningful connections between them, across dialogue turns, are identified. This information is exemplified as ① and ② in Figure 1. These decisions are taken jointly by formalizing this process as an optimization problem over the space of possible binary relations between dialogue turns and sentences.

2.3.1 Relevance Decisions

Our raw representation allows as many as six previous turns to be relevant to the classification decision, however not all turns are indeed relevant, and even relevant turns may consist only of a handful of relevant sentences. Given a dialogue consisting of $(t_1, .., t_n)$ turns, each consisting of $(t_i.s_1, .., t_i.s_k)$ sentences, we associate with each sentence.

- **Relevance** variables, denoted by $h^r_{i,j}$, indicating the relevance of the j-th sentence in the i-th turn, for the classification decision.

- **Irrelevance** variables, denoted by $h^i_{i,j}$, indicating that the j-th sentence in the i-th turn is not relevant for the classification decision.

- **Variable pair activation constraints** Given a sentence the activation of these variables should be mutually exclusive. We encode this fact by constraining the decision with a linear constraint.

$$\forall i, j, \quad h^r_{i,j} + h^i_{i,j} = 1 \qquad (1)$$

2.3.2 Dialogue Structure Decisions

In many cases the information required to make the classification is not contained in a single dialogue turn, but rather is the product of the information flow between dialogue participants. Given a dialogue consisting of $(t_1, .., t_n)$ turns, each consisting of $(t_i.s_1, .., t_i.s_k)$ sentences, we associate with every two sentences, $s_j \in t_i$, $s_k \in t_l$, such that $(i \neq l)$:

- **Sentences-Connected** variables, denoted by $h^c_{(i,j),(k,l)}$, indicating that the combination of the two sentences is relevant for the classification decision.

- **Sentences-not-Connected** variables, denoted by $h^n_{(i,j),(k,l)}$, indicating that the combination of the two sentences is not relevant for the classification decision.

- **Variable pair activation constraints** Given a sentence pair the activation of these variables should be mutually exclusive. We encode this fact by constraining the decision with a linear constraint.

$$\forall i, j, k, l \quad h^c_{(i,j),(k,l)} + h^n_{(i,j),(k,l)} = 1 \quad (2)$$

[3] Determined by lexical information (question marks, dis/agreement indications and sentence length)

- **Decision Consistency constraints** Given a sentence pair, the activation of the variable indicating the relevance of the sentence pair entails the activation of the variables indicating the relevance of the individual sentences.

$$\forall i,j,k,l, (h^c_{(i,j),(k,l)}) \implies (h^r_{i,j} \ \wedge \ h^r_{k,l}) \tag{3}$$

2.3.3 Overall Optimization Function

The boolean variables described in the previous section define a space of competing dialogue representations, each representation considers different parts of the dialogue as relevant for the objection classification decision. When making this decision a single representation is selected, by quantifying the decisions and looking for the optimal set of decisions maximizing the overall sum of decision scores. We construct this objective function by associating each decision with a feature vector, obtained using a feature function ϕ (described in Section 2.3.4), mapping the relevant part of the input to a feature set.

More formally, given an input $\mathbf{x}$, we denote the space of all possible dialogue entities (i.e., sentences and sentence pairs) as $\Gamma(\mathbf{x})$. Assuming that $\Gamma(\mathbf{x})$ is of size N, we denote latent representation decisions as $\mathbf{h} \in \{0,1\}^N$, a set of indicator variables, that selects a subset of the possible dialogue entities that constitute the dialogue representation. For a given dialogue input $\mathbf{x}$ and a dialog entity $s \in \Gamma(\mathbf{x})$, we denote $\phi_s(\mathbf{x})$ as the feature vector of s. Given a fixed weight vector $\mathbf{w}$ that scores intermediate representations for the final classification task, our decision function (for predicting "objectionable or not") becomes:

$$f_{\mathbf{w}}(\mathbf{x}) = \max_{\mathbf{h}} \quad \sum_s h_s \mathbf{w}^T \phi_s(\mathbf{x})$$
$$\text{subject to} \qquad \text{(1)-(3);} \quad \forall s; h_s \in \{0,1\} \tag{4}$$

In our experiments, we formalize Eq. (4) as an ILP instance, which we solve using the highly optimized Gurobi toolkit[4].

2.3.4 Features

In this section we describe the features used in each of the different decision types.

Relevance (h^r) :

Bag-of-words: $\{(w, t.\textit{speaker}. *^5)|\forall w \in t.s.\textit{text}\}$

Biased-Language:$\{(w, \textit{resourceContains}(w), t.\textit{speaker}.*)$ $|\forall w \in t.s.\textit{text}\}$ [6]

Irrelevance (h^i) :

SentType: $(t.s.\textit{type})$

ContainsNamedEntity $(t.s.\textit{entities} \neq \emptyset)$

Sentences-(not)-Connected (h^c,h^n) :

SentTypes: $(t_i.s_j.\textit{type}, t_k.s_l.\textit{type})$

QA pair: $(t_i.s_j.\textit{type} = \textit{Question}) \wedge (t_k.s_l.\textit{type} = \textit{Response})$

$\times \{qa|\forall w \in t_i.s_j.\textit{text}, qa = (w, t_k.s_l.\textit{type})\}$

FactPair: $(t_i.s_j.\textit{type} = \textit{Statement}) \wedge (t_k.s_l.\textit{type} = \textit{Statement})$

$\times \{qa|\forall w \in t_i.s_j.\textit{text}, qa = (w, t_k.s_l.\textit{type})\}$

SpeakerPair: $(t_i.\textit{speaker}.*, t_k.\textit{speaker}.*)$

3 Learning and Inference

Unlike the traditional classification settings, in which learning is done over a fixed representation of the input, we define the learning process over a set of latent variables. The process of choosing a good representation is formalized as an optimization problem that selects the elements and associated features that best contribute to successful classification. In the rest of this section we explain the learning process for the parameters of the model needed both for the representation decision and the final classification decision.

3.1 Learning

Similar to the traditional formalization of support vector machines (Boser et al., 1992), learning is formulated as the following margin-based optimization problem, where λ is a regularization parameter, and ℓ is the squared-hinge loss function:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \ell\left(-y_i f_{\mathbf{w}}(\mathbf{x}_i)\right) \tag{5}$$

Unlike standard support vector machines, our decision function $f_{\mathbf{w}}(\mathbf{x}_i)$ is defined over a set of latent variables. We substitute Eq. (4) into Eq.(5), and obtain the following formulation for a latent structure classifier:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \ell\left(-y_i \max_{\mathbf{h}\in\mathcal{C}} \mathbf{w}^T \sum_{s\in\Gamma(\mathbf{x})} h_s \phi_s(\mathbf{x}_i)\right) \tag{6}$$

[4] http://www.gurobi.com/

[5] "$*$" denotes all properties

[6] refers to all linguistic resources used. We also included a +/-1 word window around words appearing in these resources

This formulation is not a convex optimization problem and care must be taken to find a good optimum. In our experiments, we use the algorithm presented in (Chang et al., 2010) to solve this problem. The algorithm solves this non-convex optimization function iteratively, decreasing the value of the objective in each iteration until convergence. In each iteration, the algorithm determines the values of the latent variables of positive examples, and optimizes the modified objective function using a cutting plane algorithm. This algorithmic approach is conceptually (and algorithmically) related to the algorithm suggested by (Yu and Joachims, 2009).

As standard, we classify $\mathbf{x}$ as positive iff $f_{\mathbf{w}}(\mathbf{x}) \geq 0$. In Eq. (4), $\mathbf{w}^T\phi_s(\mathbf{x})$ is the score associated with the substructure s, and $f_{\mathbf{w}}(\mathbf{x})$ is the score for the entire intermediate representation. Therefore, our decision function $f_{\mathbf{w}}(\mathbf{x}) \geq 0$ makes use of the intermediate representation and its score to classify the input.

4 Empirical Study

Our experiments were designed with two objectives in mind. Since this work is the first to tackle the challenging task of objection prediction, we are interested in understanding the scope and feasibility of finding learning-based solutions.

Our second goal is to examine the individual aspects of our model and how they impact the overall decision and the latent structure it imposes. In particular, we are interested in understanding the effect that modeling the situated context (pragmatics) of the dialogue has on objection prediction.

4.1 Experimental Setup

Evaluated Systems In order to understand the different components of our system, we construct several variations, which differ according to the resources used during learning (see Section 2.2 for details), and the latent variable formulation used (see Section 2.3). We compare our latent model with and without using pragmatic information (denoted DIAL($\mathbf{x}_{Sit}$) and DIAL($\mathbf{x}$), respectively). We also compare two baseline systems, which do not use the latent variable formulation, these systems are trained, using linear SVM, directly over all the features activated by the h^r decisions for all the turns in the dialogue. Again, we consider two variations, with and without pragmatic information (denoted ALL($\mathbf{x}_{Sit}$) and ALL($\mathbf{x}$), respectively).

4.2 Datasets

Our dataset consists of dialogue snippets collected from the transcripts of the famous O.J. Simpson murder trial[7], collected between January of 1995 to September of that year. We also extracted from the same resource a list of all trial participants, their roles in the murder case. Section 2.1 describes the technical details concerned with mining these examples. The collected dataset consists of 4981 dialogue snippets resulting in an objection being raised, out of which 2153 were *sustained.* In addition, we also mined the trial transcript for negative examples, collecting 6269 of those examples. Negative examples are dialogue snippets which do not result in an objection. To ensure fair evaluation, we mined negative examples from each hearing, proportionally to the number of positive examples identified in the same hearing. These examples were mined randomly, by selecting dialogue snippets that were not followed by an objection in any of the three subsequent turns.

We constructed several datasets, each capturing different characteristics of courtroom interaction.

All Objections Our first dataset consists of all the objections (both sustained and overruled). The objection might not be justified, but the corresponding dialogue either has the characteristics of a justified objection, or it touches upon points of controversy. In order to simulate this scenario, we use all the examples, treating all examples resulting in an objection as positive examples. We randomly select 20% as test data. We refer to this dataset as ALLOBJ. In addition, to examine the different properties of sustained and overruled objections we create two additional dataset, consisting only of sustained/overruled objections and negative examples. We denote the dataset consisting only of sustained/overruled objections as SUSTAINEDOBJ and OVERRULEDOBJ, respectively.

Objections by Type Our final dataset breaks the objections down by type. Unfortunately, most objections are not raised with an explanation of their type. We therefore can only use subsets of the larger ALLOBJ dataset. We use the occurrences of each objection type as the test dataset and match it with negative examples, proportional to the size of the typed dataset. For training, we use all the positive examples marked with an UNKNOWN type. The size of each typed dataset appears in Table 3.

[7] http://en.wikipedia.org/wiki/O._J._Simpson_murder_case

Objection Type	#Pos/#Neg	Dial($\mathbf{x}_{Sit}$)	Dial($\mathbf{x}$)	All($\mathbf{x}_{Sit}$)	All($\mathbf{x}$)
Calls for Speculation	304 / 364	59.4	58.6	58	58
Irrelevant	275 / 330	58.5	58.6	55.2	56.6
Lack of Foundation	238 / 285	60.6	55	57	52.1
Hearsay	164 / 196	60.3	57.2	60	55
Argumentative	153 / 183	68.8	65.8	64.8	64.8
Facts not in Evidence	120 / 144	64.7	65.5	59.8	59.4
Leading Question	116 / 139	56.7	58.4	56.8	58

Table 3: **Accuracy results by objection type**. Note that the dataset size varies according to the objection type.

System	AllObj	OverruledObj	SustainedObj
All($\mathbf{x}$)	64.9	63.7	66.9
All($\mathbf{x}_{Sit}$)	65.1	63.7	67.9
Dial($\mathbf{x}$)	65.4	65.1	66.7
Dial($\mathbf{x}_{Sit}$)	**69.1**	**66.3**	**70.2**

Table 2: **Overall Accuracy results**. Results show considerable improvement when using our latent learning framework with pragmatic information.

4.3 Empirical Analysis

Overall results We begin our discussion with the experiments conducted over the three larger datasets (AllObj, SustainedObj, OverruledObj). Table 2 summarizes the results obtained by the different variations of our systems over these datasets.

The most striking observation emerging from these results is the combined contribution of capturing relevant dialogue content and interaction (using latent variables), combined with pragmatic information. For example in the AllObj, when used in conjunction, their joint contribution pushed performance to 69.1 accuracy, a considerable improvement over using each one in isolation - 65.1 for the deterministic system using pragmatic information, and 65.4 of the latent-variable formulation which does not use this information. These results are consistent in all of our experiments.

We also observe that sustained objections are easier to predict than overruled objections. This is not surprising since objections raised for unjustified reasons are harder to detect.

Pragmatic Considerations Pragmatic information in our system is modeled by using the $\mathbf{x}_{Sit}$ representation, which conditions all decisions on the speaker identity and role. The results in Table 2 show that this information typically results in better quality predictions.

An interesting side effect of using pragmatic information is its impact on the dialogue structure predictions learned as latent variables during learning. We can quantify the effect by looking at the number of latent variables activated for each model. When pragmatic information is used, 5.6 relevance variables are used on average (per dialogue snipped). In contrast, when pragmatic information is not used, this number rises to 6.3[8]. In addition, the average number of sentence-connection variables active when pragmatic information is used is 3.44. This number drops to 2.53 when it is not. These scores suggest that information about the dialogue pragmatics allows the model to take advantage of the dialogue structure at the level of the latent information, focusing the learner of higher level information, such as the relation between turns, and less on low level, lexical information. The effect of using the pragmatic information can be observed qualitatively as exemplified in Figure 2, where the latent decisions, when pragmatic information is available, construct a more topically centered representation of the dialogue for the classification decision.

Typed Objections The results over the different objection types are summarized in Table 3. These results provide some intuition on which of the objection types are harder to predict, and the contribution of each aspect of our system for that objection type.[9] We can see that across the objection types, using latent variables modeling typically results in a considerable improvement in performance. The most striking example of the importance of using pragmatic information is the Lack of Foundation objection type. This objection definition as "the evidence lacks testimony as to its authenticity or source."[10] can explain this fact, as information about the side in the trial introducing specific evidence in testimony is very likely to impact the objection decision.

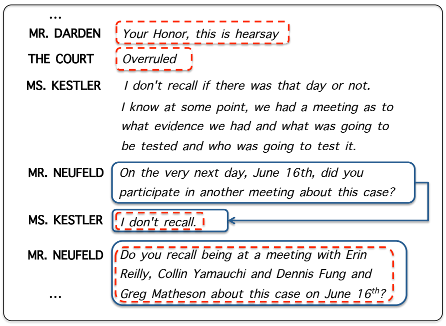
5 Related Work and Discussion

Our work applies latent variable learning to the problem of uncovering pragmatic effects in court-

[8] The average number of sentences per dialogue is 8.6

[9] Since these datasets vary in size, their results are neither directly comparable to each other nor to the results in Table 2.

[10] http://en.wikipedia.org/wiki/Foundation_(evidence)



...

MR. DARDEN *Your Honor, this is hearsay*

THE COURT *Overruled*

MS. KESTLER *I don't recall if there was that day or not. I know at some point, we had a meeting as to what evidence we had and what was going to be tested and who was going to test it.*

MR. NEUFELD *On the very next day, June 16th, did you participate in another meeting about this case?*

MS. KESTLER *I don't recall.*

MR. NEUFELD *Do you recall being at a meeting with Erin Reilly, Collin Yamauchi and Dennis Fung and Greg Matheson about this case on June 16th?*

...

Figure 2: **Example of the pragmatic effect on latent dialogue structure.** Constructing the latent dialogue structure over situated text marks unrelated sentences as irrelevant, while marking topically related sentences and identifying the connection between the question-answer pair (decisions marked in solid blue lines). When trained *without* situated information, the latent output structure marks topically unrelated sentences as relevant for objection classification. Note that in this case all the edge variables are turned off (marked with dashed red lines).

room dialogues. We adopted the structured latent variable model defined in (Chang et al., 2010), and use ILP to solve the structure prediction inference problem (Roth and Yih, 2007).

Our prediction task, identifying the actionable result of a dialogue, requires capturing the dialogue and discourse relations. While we view these relations as latent variables in the context of action prediction, studying these relations independently has been the focus of significant research efforts, such as discourse relations (Prasad et al., 2008), rhetorical structure (Marcu, 1997) and dialogue act modeling (Stolcke et al., 2000). Fully supervised approaches for learning to predict dialogue and discourse relations (such as (Baldridge and Lascarides, 2005)) typically requires heavy supervision and has been applied only to limited domains.

Moving away from full supervision, the work of (Golland et al., 2010) uses a game-theoretic model to explicitly model the roles of dialogue participants. In the context of dialogue and situated language understanding, the work of (Artzi and Zettlemoyer, 2011) shows how to derive supervision for dialogue processing from its structure.

Discriminative latent variables models have seen a surge of interest in recent years, both in the machine learning community (Yu and Joachims, 2009; Quattoni et al., 2007) as well as various application domains such as NLP (Täckström and McDonald, 2011) and computer vision (Felzenszwalb et al., 2010). In NLP, one of the most well-known applications of discriminative latent structured classification is to the Textual Entailment (TE) task (Chang et al., 2010; Wang and Manning, 2010). The TE task bears some resemblances ours, as both tasks require making a binary decision on the basis of a complex input object (i.e., the history of dialogue, pairs of paragraphs), creating the need for a learning framework that is flexible enough to model the complex latent structure that exists in the input. Another popular application domain is sentiment analysis (Yessenalina et al., 2010; Täckström and McDonald, 2011; Trivedi and Eisenstein, 2013). The latent variable model allows the learner to identify finer grained sentiment expression than annotated in the data.

A related area of work with different motivations and different technical approaches has focused on attempting to understand narrative structure. For instance, Chambers and Jurafsky (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009) model narrative flow in the style of Schankian scripts (Schank and Abelson, 1977). Their focus is on common sequences of actions, not specifically related to dialogue. Somewhat more related is recent work (Goyal et al., 2010) that aimed to build a computational model of Lehnert's Plot Units (Lehnert, 1981) model. That work focused primarily on actions and not on dialogue: in fact, their results showed that the lack of dialogue understanding was a significant detriment to their ability to model plot structure.

Instead of focusing on actions, like the above work, we focus on dialogue content and relationships between utterances. Furthermore, unlike most of the relevant work in NLP, our approach requires only very lightweight annotation coming for "free" in the form of courtroom objections, and use a latent variable model to provide judgements of relevant linguistic and dialogue relations, rather than annotating it manually. We enhance this model using pragmatic information, capturing speakers' identity and role in the dialogue, and show empirically the relevance of this information when making predictions.

It is important to recognize that courtroom objections are not the only actionable result of dialogues. Many discussions that occur on online forums, in social media, and by email result in measurable *real-world* outcomes. We have shown that one particular type of outcome, realized as a speech-act, can drive dialogue interpretation; the field is wide open to investigate others.

References

Adam Vogel and Christopher Potts and Dan Jurafsky. 2011. Implicatures and Nested Beliefs in Approximate Decentralized-POMDPs. In *EMNLP*.

Yoav Artzi and Luke S. Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *EMNLP*.

Jason Baldridge and Alex Lascarides. 2005. Probabilistic head-driven parsing for discourse structure. In *CoNLL*.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 144–152.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, June.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL/IJCNLP*, pages 602–610.

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *NAACL*.

Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *EMNLP*.

Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

K. Hyland. 2005. Metadiscourse: Exploring interaction in writing. In *Continuum, London and New York*.

W. G. Lehnert. 1981. Plot units and narrative summarization. In *Cognitive Science*.

Daniel Marcu. 1997. The rhetorical parsing of natural language texts. In *ACL*.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L Robaldo, A. Joshi, and B. Webber. 2008. The penn discourse treebank 2.0. In *LREC*.

Ariadna Quattoni, Sybor Wang, L-P Morency, Michael Collins, and Trevor Darrell. 2007. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of ACL*.

E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *NAACL*.

D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Roger C. Schank and Robert P. Abelson. 1977. Scripts, plans, goals and understanding. In *ACL/IJCNLP*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *COMPUTATIONAL LINGUISTICS*, 26:339–373.

Oscar Täckström and Ryan T. McDonald. 2011. Discovering fine-grained sentiment with latent variable structured prediction models. In *ECIR*.

Rakshit Trivedi and Jacob Eisenstein. 2013. Discourse connectors for latent subjectivity in sentiment analysis. classification. In *NAACL*.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*.

Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *EMNLP*.

C. Yu and T. Joachims. 2009. Learning structural svms with latent variables. In *Proc. of the International Conference on Machine Learning (ICML)*.