Joint Inference for Mode Identification in Tutorial Dialogues

Deepak Venugopal
Department of Computer Science
University of Memphis
Memphis, TN 38152
dvngopal@memphis.edu

Vasile Rus
Department of Computer Science
Institute for Intelligent Systems (IIS)
University of Memphis
Memphis, TN 38152
vrus@memphis.edu

Abstract

Identifying *dialogue acts* and *dialogue modes* during tutorial interactions is an extremely crucial sub-step in understanding patterns of effective tutor-tutee interactions. In this work, we develop a novel joint inference method that labels each utterance in a tutoring dialogue session with a dialogue act and a specific mode from a set of pre-defined dialogue acts and modes, respectively. Specifically, we develop our joint model using Markov Logic Networks (MLNs), a framework that combines first-order logic with probabilities, and is thus capable of representing complex, uncertain knowledge. We define first-order formulas in our MLN that encode the inter-dependencies between dialogue modes and more fine-grained dialogue actions. We then use a joint inference to jointly label the modes as well as the dialogue acts in an utterance. We compare our system against a pipeline system based on SVMs on a real-world dataset with tutoring sessions of over 500 students. Our results show that the joint inference system is far more effective than the pipeline system in mode detection, and improves over the performance of the pipeline system by about 6 points in F1 score. The joint inference system also performs much better than the pipeline system in the context of labeling modes that highlight important pedagogical steps in tutoring.

1 Introduction

One-on-one instruction, i.e. tutoring, is one of the most effective forms of instruction. Intelligent Tutoring Systems (ITS) (Rus et al., 2013) have the potential to make effective and affordable "instruction-for-all" a reality since they do not suffer from traditional constraints such as lack of trained and expensive human tutors, physical teaching facilities, etc. However, in order to build effective automated tutoring systems, i.e. tutoring systems the induce student learning gains, we first need to understand what effective human tutors do. Specifically, we would like to identify specific pedagogical steps that promote effective tutoring. For instance, a good tutor may start by building a rapport with the students, followed by helping the student identify the domain of the problem, and so on. The sequence of steps taken by expert human tutors can in turn be used to improve the performance of ITS by re-enacting such effective tutorial strategies that are likely to promote better learning.

Understanding what good tutors do to help students learn has been the subject of much theoretical and empirical research (Chi et al., 2001; Eugenio et al., 2006; Cade et al., 2008; Jeong et al., 2008; Boyer et al., 2010; Lehman et al., 2012). A standard approach to understanding effective tutoring is to characterize tutor-tutee interactions based on the actions tutors and tutees take and then identify patterns of such actions that are associated with effective tutoring. For instance Cade et al. (Cade et al., 2008) used dialogue acts, which are constructs used to describe the intentions behind speakers' utterances, to model tutor-learner dialogue-based interactions. Boyer et al. (Boyer et al., 2010) modeled interactions as a combination of both task actions, which specify fine-grained steps taken by a user such as opening a file, and dialogue acts. However, dialogue acts only identify individual, isolated acts, e.g. asking a question, associated with a particular utterance lacking to characterize the meaning of a sequence of coherent actions, e.g. by the tutor, that might reveal high level constructs such as pedagogical strategies,

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2000–2011, Osaka, Japan, December 11-17 2016.

Speaker	Utterance	Act	Subact	Mode
Tutor	Welcome	Expressive	Greeting	Opening
Student	Hi	Expressive	Greeting	Opening
Tutor	How can i help you today?	Prompt	Question	Rapport Building
Tutor	you can just write the problem on board	Assertion	Process	Process Negotiation
Student	okay	Expressive	Neutral	Process Negotiation
Tutor	so we need to find the slope-intercept form	Assertion	Identification	Problem Identification
Student	are we asked for the graph or just the equation	Question	Neutral	Problem Identification
Tutor	We have slope given as m =0	Assertion	Calculation	Scaffolding
Student	I graphed the intercept (0, -27) correctly	Assertion	Calculation	Scaffolding
Tutor	and the y-intercept is (0, -27)	Assertion	Calculation	Scaffolding
Student	the slope is 0	Assertion	Calculation	Scaffolding

Table 1: Example for modes, acts and subacts in a dialogue.

e.g. scaffolding. In this work, we present a novel approach to identify higher-level tutorial constructs called *modes* in tutor-learner dialogue-based interactions. Specifically, dialogue modes are sequences of dialogue acts that map to pedagogical goals such as scaffolding (and sometimes to general dialogue goals such as opening a conversation). An example of the hierarchy of modes, dialogue acts and subacts in utterances is shown in Table 1.

As is often the case in several NLP tasks, a *pipeline* architecture can be naturally adopted to identify dialogue modes. Specifically, we first label acts in each utterance of the dialogue, then, using the labeled acts, we label subacts, and using both the labeled acts and subacts, we finally label the higher level modes. However, as is the case in general with pipeline based architectures, such a system is bound to have a fair amount of *error propagation*, where errors in labeling the acts or subacts affect the performance of mode labeling. Therefore, we propose a novel *joint inference* method for this task where we label modes jointly with dialogue acts and subacts, thereby taking advantage of the inter-dependencies between them. Prior approaches in the ITS research community have largely focused on dialogue act classification (Marineau et al., 2000; Serafin and Di Eugenio, 2004; Moldovan et al., 2011) or on mode labeling given labeled dialogue acts (Cade et al., 2008; Boyer et al., 2010; Rus et al., 2015). To the best of our knowledge, our work is the first joint inference method for this task.

We develop our joint inference system using a modeling language called Markov Logic Networks (MLNs) (Domingos and Lowd, 2009). MLNs are a powerful representation, where uncertain domain-knowledge is encoded as first-order formulas with weights attached to each formula. The weights in an MLN model indicate the uncertainty associated with the formulas. The larger the weight, the more confidence we have in the formula being true. Over the last few years, MLNs have been routinely used for several joint inference tasks in entity resolution (Poon and Domingos, 2008), event extraction (Poon and Vanderwende, 2010; Venugopal et al., 2014) and question answering (Khot et al., 2015). The main advantage of MLNs is that it can represent a large, complex probabilistic model through a highly compact, lifted representation specified through first-order formulas. However, at the same time, the compact representation makes scaling up probabilistic inference and learning a huge challenge in MLNs (Domingos and Lowd, 2009; Poon and Domingos, 2007). More specifically, in our task, the Markov network underlying the MLN turns out to be extremely large with millions of nodes and edges. By systematically exploiting the structure of our MLN model, we scale up MLN learning and inference methods for our task.

We evaluate our joint inference model on a dataset of human annotated dialogue transcripts of 500 students with around 32,000 dialogue utterances. To compare against our approach, we build a baseline, pipeline system using Support Vector Machines where we treat utterances as independent instances and sequentially label dialogue acts, subacts and modes in this dataset. We then compare our joint inference model with the baseline and obtain nearly a 6 point increase in F1-score for mode labeling with both higher recall and higher precision, clearly showing the promise of our joint inference approach.

The rest of this paper is organized as follows. We first present related work and give a brief overview of MLNs. We then present our joint inference model using MLNs and finally conclude with our evaluation.

2 Related Work

Speech-act theory that was developed in in the 1960's (Austin, 1962; Searle, 1969) has been typically used to model speakers' intentions. According to speech-act theory, when we say something, we do something. There are three levels of speech: the locutionary level which is the actual utterance, the illocutionary level which is the intention behind the utterance and perlocutionary level which is the effect of the utterance. Speech acts model the illocutionary level and denote speech acts such as greeting (*Hello*), questioning (*how is the weather?*), etc.

A speech act could be described as the sum of the illocutionary forces carried by an utterance (Moldovan et al., 2011). It is worth mentioning that within one utterance, speech acts can be hierarchical, hence the existence of a division between direct and indirect speech acts, the latter being those by which one says more than what is literally said, in other words, the deeper level of intentional meaning. In the phrase, *Would you mind passing me the salt?*, the direct speech act is the request best described by *Are you willing to do that for me?* while the indirect speech act is the request *I need you to give me the salt*. In a similar way, in the phrase, *Bill and Wendy lost a lot of weight with a diet and daily exercise*. the direct speech act is the actual statement of what happened, i.e., *They achieved "this" by doing "that"*, while the indirect speech act could be the encouraging, *If you do the same, you could lose a lot of weight too*. The present study assumes there is one direct speech act per utterance.

The task of classifying direct speech acts has been well-studied in the general context (Reithinger, 1995; Stolcke et al., 2000; Reithinger and Maier, 1995; Ries, 1999; Moldovan et al., 2011) as well as in the specific context of ITS (Marineau et al., 2000; Serafin and Di Eugenio, 2004; Samei et al., 2014). A related problem of generating the next speech act in a dialogue has also been investigated to some extent (Reithinger, 1995; Bangalore and Stent, 2009). Also, there is work on automatically discovering dialogue acts using data-driven approaches (Moldovan et al., 2011) but it is beyond the scope of this paper to automatically discover the dialogue acts in our tutoring sessions. In the automated speech act classification literature, typically researchers have considered rich feature sets extracted from the utterances such as the actual words (possibly lemmatized or stemmed) and ngrams (sequences of consecutive words) to characterize the type of speech act.

Dialogue modes in tutorial dialogues are sequences of dialogue acts that correspond to general conversational segments of a dialogue, e.g. an Opening mode corresponds to the first phase of the dialogue when the conversational partners greet each other, or to segments associated with pedagogical goals, e.g. a Scaffolding mode would correspond to the tutorial dialogue segment when the student works on something and the tutor scaffolds the learners activity. Compared to speech act classification, mode identification has been far less studied. Based on a manual analysis, Cade et al. (Cade et al., 2008) defined a set of eight mutually exclusive tutorial modes: introduction, lecture, highlighting, modeling, scaffolding, fading, off-topic, and conclusion. An interesting aspect of their analysis is the granularity at which they defined the pedagogically important modes. In their approach, the modes correspond to either the tutor or the student or both focusing on solving a full problem. In our approach, we used a different definition of modes proposed by Morrison et al. (Morrison et al., 2014). In this approach, a tutor or student could switch between proposed modes while working on a particular problem. That is, a particular mode is not associated with one problem solving task but rather with parts of such a problem solving task. Finally, Boyer et al. (Boyer et al., 2010) used acts in conjunction with more specific task actions, e.g., opening a specific file, etc., to discover hidden modes using a HMM. In contrast, we assume a pre-defined set of modes (see next section) that generalize across tutors and identify modes and acts jointly. This is similar to the Conditional Random Fields (CRF) approach proposed by Rus and colleagues (Rus et al., 2015) who used a expert-defined set of modes. It should be noted that Rus and colleagues report a best dialogue mode labeling performance of accuracy=57.18% when they used gold, i.e. human-labeled, dialogue acts as input. The accuracy dropped to 28.77% when automatically labeled dialogue acts were provided as input to the CRF-based dialogue mode labeling system. It should be noted that our results reported here are not exactly comparable to the ones reported by Rus and colleagues as they used a different, albeit related, human-labeled dataset to train and test their system.

3 Background

In this section, we give a brief overview of MLNs and describe the dataset used in this paper.

3.1 Markov Logic Networks

Markov logic networks (MLNs) unify first-order logic with Markov networks (undirected probabilistic graphical models abbreviated as PGMs). Formally, an MLN consists of a set of weighted first-order formulas, $\{(f_i; w_i)\}_{i=1}^{K}$, where f_i is a first-order formula and w_i is a real-valued weight attached to f_i. The weight w_i quantifies the uncertainty in f_i. Higher the weight of a formula, the more belief we have that the formula is true. If the weight w_i is ∞, then it acts as a hard constraint that f_i should always be true, while a weight $-\infty$ specifies the hard constraint that f_i should always be false. MLNs assume Herbrand semantics, i.e., there is a finite number of objects that can be substituted for the variables in the first-order formulas. This set of real-world objects is referred to as the domain. Throughout this paper, we specify constants with capital letters (e.g., A, B, etc.) and variables in the formulas with small letters (e.g., x, y, etc.)

A *ground atom* in the MLN is a first-order predicate where all variables are grounded with constants from the domain. Similarly, a ground formula is an instantiation of a first-order formula, where all variables have been grounded with constants from the domain. Given a domain of interest, MLNs specify a Markov network where a ground atom (a first-order predicate where all variables are grounded with constants) is a binary variable in the network and each ground formula (a first-order formula grounded with constants) is a function over the variables specific to that formula. For example, assume that the domain for the MLN, $\texttt{Smokes}(x) \Rightarrow \texttt{Cancer}(x)$; w, is equal to $\{A, B\}$. Then, $\texttt{Smokes}(A)$ is a ground atom in the MLN which represents a binary random variable in the Markov network. Similarly, $\texttt{Smokes}(A) \Rightarrow \texttt{Cancer}(A)$ represents a function in the Markov network defined over the binary variables corresponding to $\texttt{Smokes}(A)$ and $\texttt{Cancer}(A)$. An assignment (either 0 or 1) to all possible ground atoms in the MLN, $\texttt{Smokes}(A)$, $\texttt{Smokes}(B)$, $\texttt{Cancer}(A)$, $\texttt{Cancer}(B)$, is called a *world*. The MLN describes a *log-linear* model where the probability distribution is defined over the set of possible worlds. Specifically, the probability distribution represented by the MLN is given by,

$$\Pr(\omega) = \frac{1}{Z} \exp\left(\sum_i w_i N_i(\omega)\right) \tag{1}$$

where $N_i(\omega)$ is the number of groundings of the first order formula f_i that evaluate to True given a world ω.

Since MLNs are simply a compact representation of PGMs, all inference tasks in PGMs are also applicable to MLNs. Specifically, the two main inference tasks for MLNs are, 1) Marginal inference, and 2) MAP inference. In marginal inference, given evidence atoms, i.e., ground atoms whose truth value is known/observed, the task is to compute marginal probabilities over other query atoms. For example, say we are given evidence atoms $\texttt{Smokes}(A)$ and $\texttt{Smokes}(B)$, the task is to compute probabilities such as $P(\texttt{Cancer}(A)|\texttt{Smokes}(A), \texttt{Smokes}(B))$. In MAP inference, given evidence, we compute the assignment to the non-evidence atoms such that the probability of that assignment is maximized. For instance, given evidence $\texttt{Smokes}(A)$ and $\texttt{Smokes}(B)$, we need to compute the assignment to $\texttt{Cancer}(A)$, $\texttt{Cancer}(B)$ for which the probability is maximum in the joint distribution. Both marginal inference and MAP inference are computationally intractable and therefore typically approximate algorithms are used for both these tasks.

3.2 Dataset

Our dataset consists of dialogue transcripts of 500 tutoring sessions collected from 500 students working on elementary algebra and physics problems. In all, there are 32,368 individual utterances in these tutorial sessions, where we define an utterance as a single dialogue turn by either the student or the tutor. We selected this data from a sample of sessions obtained from an online, commercial tutoring service. These

sessions are about problem solving in the context of various Algebra and Physics topics. These are student-initiated sessions, mostly in the context of homework help.

We label each utterance with a predefined set of dialogue acts. The dialogue act taxonomy was developed with the assistance of subject matter experts, all experienced tutors and tutor mentors working for an online commercial tutoring service, resulting in a fine-grained 2-level hierarchical taxonomy that includes 17 main act categories. Each main dialog act category consists, in turn, of different subcategories, which we refer to as *subacts*, resulting in an overall taxonomy of 196 distinct dialog act-subact combinations. The size of the dialogue-act and -subact taxonomy is at least one order of magnitude larger than taxonomies proposed and used by others such as Boyer and colleagues (Boyer et al., 2010). It should be noted that the dialog acts were defined and refined to minimize overlap between categories and maximize the coverage of distinct acts.

There were a set of 17 dialogue modes defined by the experts and each utterance was annotated with the act, subact and the dialogue mode for the utterance by humans. The data was manually annotated by a group of tutoring experts who were trained on both the dialogue act taxonomy and set of dialogue modes. When annotating independently, the inter-annotator agreement was 80.91% and kappa statistic was 0.77 for dialogue acts and 64.90% and kappa of 0.63 for dialogue acts and subacts together. These values correspond to very good agreement among the annotators. For modes, the agreement was lower at 55.03% and kappa of 0.47. The list of modes and the number of times they occur in our labeled data set is shown below.

Opening(667), Problem Identification (3177), Assessment (338), Method Identification (126), Method Roadmap (1056), Rapport Building (1006), Process Negotiation (3281), MetaCognition (533), Sensemaking (2889), Fading (2466), Scaffolding (4574), Modeling (1159), Telling (1806), Session (8), ITSupport (1251), WrapUp/Close (871), and Off-topic (4).

4 MLN Model

Here, we describe our joint inference model for identifying dialogue modes based on MLNs. We first describe the set of first-order formulas of the joint model. We then discuss how we perform joint inference and learning scalably in our model.

4.1 MLN Formulas

The four main predicates in our MLN are: `Act`, `Subact`, `Mode` and `ModeSwitch`. We next describe each of these predicates.

`Act`(s, t, a!) is a predicate that asserts that the dialogue act in the tutorial session corresponding to student s, at time step t, is equal to a. When defining our MLN, we refer to "time" as the utterance number in a dialogue session between the tutor and student. The "!" mark is a special symbol in the MLN language that specifies a hard constraint that for every grounding of s and t, there is exactly one act label. That is, every utterance corresponds to one and only one dialogue act. Similarly, `Subact`(s, t, u!) asserts that the dialogue subact in the tutorial session for student s at time t is equal to u. `Mode`(s, t, m!) asserts that the dialogue mode in the tutorial session for student s at time t is equal to m. Finally, `ModeSwitch`(s, t) asserts that there was a switch in dialogue mode at time t for the tutoring session associated with student s. That is, the mode in the previous time step was different from the mode in the current time step. Since our interest in this task is mode identification, `Mode` is called as a *query* predicate and the inference task is to collectively set a 0/1 truth assignment to all groundings of this predicate. `Act`, `Subact` and `ModeSwitch` are called *hidden* predicates since the truth assignments of their groundings are unknown.

Using the above predicates, we define the following formulas. Unless specified, all variables in the below described formulas are assumed to be universally quantified.

1. The first set of hard formulas specify that each act maps to a specific subset of subacts. This formula encodes the two-level hierarchy that we define in our taxonomy. We specify this by implication formulas of the form,

$$\texttt{Act}(s, t, A) \Leftrightarrow \texttt{Subact}(s, t, U_1) \vee \ldots \texttt{Subact}(s, t, U_k)$$

where $U_1 \ldots U_k$ are possible subacts corresponding to act A.

2. Next, we define hard formulas that encode the rule of mode switching. That is, we specify that mode-switching causes a shift in the dialogue mode using two implications.

$$\texttt{ModeSwitch}(s,t) \wedge \texttt{Mode}(s,t-1,m) \Rightarrow \neg\texttt{Mode}(s,t,m)$$
$$\neg\texttt{ModeSwitch}(s,t) \wedge \texttt{Mode}(s,t-1,m) \Rightarrow \texttt{Mode}(s,t,m)$$

3. The first and last modes of a dialogue are always fixed. We specify this with a conjunctive hard formula,

$$\texttt{Mode}(s,T_0,Opening) \wedge \texttt{Mode}(s,T_k,Closing)$$

where T_0 is the first utterance in the dialogue and T_k is the last utterance in the dialogue.

4. We encode the inter-dependency between modes, acts and subacts with a set of soft formulas. Specifically, we model this interaction by encoding a formula that connects two successive time-steps of a dialogue. The resulting formulation is similar to encoding Hidden Markov Models using MLNs, where we assert that the dialogue mode at time-step t is influenced by the acts, modes and subacts at the previous time-step. Clearly, this is a formula which would not hold true for all possible instantiations. Therefore, we specify a soft formula of the form,

$$\texttt{Mode}(s,t-1,+m_1) \wedge \texttt{Act}(s,t-1,+a) \wedge \texttt{Subact}(s,t-1,+u) \Rightarrow \texttt{Mode}(s,t,+m_2)$$

An important aspect to note about the above soft formula is the "+" sign for variables in the formula. The "+" sign is a special symbol in MLNs that allows us to define multiple weights for a single formula. Recall that in MLNs, generally, all the groundings of a first-order formula share the exact same weight. However, in several practical cases, we need to decrease the bias of the model by introducing more parameters for it. With the use of a "+" sign, we can increase the total number of weights in the MLN and thus induce more complex distributions. Specifically, we can set a different weight for each partially ground formula obtained by grounding all variables in the formula corresponding to the $+$ symbol. For instance, in this case, for each possible grounding of the variables m_1, m_2, u and a in the formula, we will define a distinct weight. This allows us more degrees of freedom to model the data rather than using a single weight for the formula.

5. Next, we define several soft formulas that connect features of the dialogue utterances to `Mode`, `Act`, `Subact` and `ModeSwitch`. Let $\texttt{Feature}_1 \ldots \texttt{Feature}_N$ denote N features extracted from the utterances (we discuss the actual features in the next section), then, we encode these features using soft formulas of the form,

$$\texttt{Feature}_1(s,t,+f_1) \wedge \texttt{Feature}_2(s,t,+f_2) \ldots \texttt{Feature}_N(s,t,+f_N) \Rightarrow \texttt{Mode}(s,t,+m)$$
$$\texttt{Feature}_1(s,t,+f_1) \wedge \texttt{Feature}_2(s,t,+f_2) \ldots \texttt{Feature}_N(s,t,+f_N) \Rightarrow \texttt{ModeSwitch}(s,t)$$
$$\texttt{Feature}_1(s,t,+f_1) \wedge \texttt{Feature}_2(s,t,+f_2) \ldots \texttt{Feature}_N(s,t,+f_N) \Rightarrow \texttt{Act}(s,t,+a)$$
$$\texttt{Feature}_1(s,t,+f_1) \wedge \texttt{Feature}_2(s,t,+f_2) \ldots \texttt{Feature}_N(s,t,+f_N) \Rightarrow \texttt{Subact}(s,t,+u)$$

4.2 Joint Inference

Given the MLN specified in the previous section, the inference task is to jointly compute an assignment to all possible groundings of the query predicate, `Mode`. Specifically, we compute this assignment as a solution to the following optimization problem

$$\max_{\omega'} \sum_{h \in H} P(Q = \omega') \tag{2}$$

where H is the set ground atoms of hidden predicates and Q is the set of ground atoms of query predicates, ω' is an assignment on all atoms in Q. However, Eq. (2) which is an instance of the *marginal-MAP* (MMAP) inference problem involves both summation (summing out the hidden variables) and

maximization, and is well-known to be a very hard problem (Park and Darwiche, 2004). Instead, we approximate the solution to the MMAP problem with a solution to the following *Max a-posteriori* (MAP) inference problem which only involves maximization.

$$\max_{\omega} P(Q \cup H = \omega) \tag{3}$$

where ω is an assignment on all atoms in $Q \cup H$. To obtain an approximate MMAP assignment for only the atoms in Q, we simply project the complete solution obtained from the MAP problem in Eq. (3) on the atoms in Q. Note that even the MAP problem in Eq. (3) is NP-hard. However, several highly efficient off-the-shelf approximate MAP solvers can be used to obtain high-quality approximations. Notable examples include MaxWalkSAT (Kautz et al., 1997), dual-decomposition based solvers (Sontag and Globerson, 2011) and ILP based solvers such as Gurobi (Gurobi., 2013). In our experiments, we use Gurobi, a state-of-the-art ILP solver to compute the MAP solution for the MLN (Sarkhel et al., 2014). However, it turns out that a naive application of approximate MAP solvers to our problem is still infeasible in practice. For instance, suppose we have 500 students' dialogues in our dataset, and each dialogue has on average 100 utterances/time-steps, then, the formula, $\texttt{ModeSwitch}(s, t) \wedge \texttt{Mode}(s, t-1, m) \Rightarrow \neg \texttt{Mode}(s, t, m)$ itself has at least 1 million possible groundings. In other words, grounding the entire MLN and then applying MAP inference on the ground MLN quickly becomes infeasible. However, we notice that our MLN has a decomposable structure, i.e., the ground Markov network obtained by grounding the MLN with a single student's dialogue is independent of the ground Markov network obtained when we ground the MLN with the rest of students' dialogues. This means that we can decompose the MAP problem as,

$$\prod_{k} \max_{\omega_k} P(Q_k \cup H_k = \omega_k) \tag{4}$$

where Q_k and H_k are the query and hidden atoms specific to the dialogues of student the k-th student and ω_k is an assignment to all atoms in $\{Q_k, H_k\}$. Thus, using Eq. (4), we can essentially compute the MAP solution independently for each student dialogue using a standard MAP solver which greatly reduces the computational requirements of the solver and allows us to scale up joint inference over our large dataset of dialogues.

Next, we describe weight-learning for the soft formulas in our MLN. Specifically, we use gradient ascent to compute weights of the soft formulas that maximize the log-likelihood of our dataset. Note that, our model contains hidden variables that are not observed directly, i.e., atoms corresponding to `Act`, `Subact` and `ModeSwitch`. Due to the presence of these hidden variables in our model, the resulting log-likelihood function is no longer convex. Therefore, gradient ascent can get struck in local optima. We reduce the severity of the problem using *random restarts* (Selman et al., 1996). That is, we start gradient ascent from several different initialization points and average all the different weights that gradient ascent converges to when starting from these different initialization points. Note that in each step of gradient ascent, we need to compute the gradient as,

$$\mathbb{E}_w[N_i] - \mathbb{E}_w[N_i'] \tag{5}$$

where $\mathbb{E}_w[N_i]$ is the expected number of groundings of the i-th soft formula that are true given the current set of weights w w.r.t the MLN distribution $P(Q|H)$ and $\mathbb{E}_w[N_i']$ is the expected number of groundings of the i-th soft formula that are true in the dataset w.r.t the MLN distribution $P(Q)$. Both expectations are intractable to compute exactly. Therefore, we approximate these distributions with their respective MAP values. This means that, for each gradient ascent step, we run MAP inference twice and compute the approximate expectations and from the approximate expectations, we compute the approximate gradient direction. We continue updating the weights with the gradient until the weights converge. In order to reduce computation, we compute the weights only for a feasible set of groundings of the "+" variables in the soft formulas. For instance, consider soft formula 4 in the previous section. Here, the number of groundings of the "+" variables is equal to the product of $|Modes| * |Modes| * |Acts| *$

Classifier Type	Model Type	Features	#Classes
Act Classifier	$SVM^{multiclass}$	Unigrams, Bigrams, Number of Tokens, Ending Punctuation, Utterance number	17
Subact Classifier	$SVM^{multiclass}$	All features of Act Classifier, the output acts labeled by Act Classifier	61
Mode Classifier	$SVM^{multiclass}$	All features of Subact Classifier, the output Subacts labeled by Subact Classifier	17
Mode Switch Classifier	SVM	Merged features of Mode classifier for two successive utterances	2

Table 2: SVM Models for Act, Subact, Mode identification and Mode Switch detection.

Metric	Pipeline			Joint Model		
	Precision	Recall	F1	Precision	Recall	F1
Average	0.275	0.28	0.271	0.338	0.341	**0.332**
Weighted-Average	0.32	0.34	0.324	0.375	0.39	**0.378**

Table 3: 5-fold Cross Validation results for Mode labeling.

$|Subacts|$. However, the number of feasible combinations is much lower. That is, several combinations never occur in the training dataset and we assume that for all such cases, the weight is 0 (the likelihood that the formula is true/false is the same). We remove these cases from the set of ground formulas and compute weights only for the remaining set of feasible groundings.

4.3 Learning Feature Based Formulas

Unfortunately, the above weight learning procedure does not work very well to learn weights of the feature-based soft formulas (listed as 5. in the previous section). The number of weights that we need to learn corresponding to the feature-based formulas turns out to be extremely large. Specifically, grounding the "+" variables, we will have at least $O(N * d * |Modes| * |Acts| * |Subacts|)$, where N is the number of features, d is an upper-bound on the number of possible feature-values for a feature. For lexical features of the utterances such as unigrams, bigrams, etc. this number is extremely large. Thus, weight-learning for the MLN that includes the feature-based soft formulas is infeasible in our model. Instead, we utilize the flexibility of MLNs to incorporate the feature-based soft formulas implicitly. Specifically, we remove all the feature-based formulas from the MLN and learn the MLN weights using only the other formulas. We then derive weights for the feature-based formulas through a separate model and add this back into the MLN as described next.

We train an SVM-based pipeline system to label the acts, subacts, modes and mode-switches in sequence. That is, we use $SVM^{multiclass}$ to first label the acts. Using the labeled acts, we label the subacts, and using both the labeled acts and subacts, we label the modes. We detect mode-switches using a binary SVM classifier. The features used in each of these models are shown in Table 2. The SVM-based pipeline system yields confidence values in the form of hyper-plane distances for each dialogue utterance for every mode, act, subact and also whether a mode switch occurred. Specifically, given an utterance t for student s, we will have hyper-plane distances for `Act`(s,t,$+a$), `Subact`(s,t,$+u$), `Mode`(s,t,$+m$) and `ModeSwitch`(s,t) (Note that, these are the atoms in the RHS of the soft formulas specified in 5). We then add to the MLN, a unit clause corresponding to the RHS of each soft formula, with the weight of the unit clause given by the SVM confidence value, which we normalize into the range $[-1, 1]$. Thus, if the SVM classifier is confident that an utterance number t for student s has mode type M, it will output a large confidence value for the type M label, which in turn is encoded into the MLN as the formula `Mode`(s,t,M) with a large weight. This will then make it more likely that the atom `Mode`(s,t,M) will be set as true when computing the MAP solution for the overall joint model.

Type	Pipeline			Joint Model		
	Precision	Recall	F1	Precision	Recall	F1
Act	0.672	0.68	0.65	0.69	0.698	**0.68**
Subact	0.49	0.51	0.48	0.518	0.535	**0.513**

Table 4: 5-fold Cross Validation results for the hidden predicates, Act and Subact (weighted-average F1 scores).

5 Experiments

This section presents the details of our experimental setup and the results obtained. As already mentioned, we compared a pipeline approach with the MLN joint-inference approach.

5.1 Setup

We evaluate the performance of our joint model by comparing it with the SVM based pipeline system which uses the features outlined in Table 2. This system is similar to the one presented in Rus et al. (Rus et al., 2015) who used a related, but not identical dataset, except that Rus et al. use Conditional Random Fields to label the modes, while, here we use SVMs. The performance of Rus et al.'s mode identification system that uses the labels of acts and subacts that were detected using a supervised classifier, is similar to ones we present here.

For our joint model, we use Gurobi, a state-of-the-art ILP solver, to solve the MAP inference problem. That is, we ground the MLN with dialogue data from each student independently and solve the MAP problem for each such partially ground MLN independently using Gurobi. Note that this problem is embarrassingly parallel since each MAP solution can be computed independently of the others. Using this, we could run a single instance of MAP inference over the entire dataset in just a few minutes using a cluster of 5 8-core machines, each with 8GB RAM.

5.2 Results

Table 3 shows a comparison of the F1-scores, precision and recall obtained by running 5-fold cross validation. The scores are reported for simple average of the scores (average over all mode labels) and for the weighted average (average weighted by instances of a particular label). As seen here, the joint method clearly outperforms the pipeline method in every case, in terms of F1-score, precision and recall. The average F1-score we obtained using the joint method was nearly 6 points higher than the average F1-score obtained using the pipeline SVM classifier. Particularly, both precision and recall of mode identification improved over both metrics.

Next, we evaluated statistical significance of our results. Specifically, we ran 5-fold paired t-tests (cf. (Dietterich, 1998)) to determine if our results were significant. Our results showed that our results attained statistical significance at $p \leq 0.05$, i.e., we obtained $t = 3.75$ with $p = 0.009$.

In our next experiment, we evaluated the performance of our model on hidden predicates. Specifically, Table 4 shows a comparison of how well the systems perform in terms of labeling the hidden ground atoms (ground atoms of the `Act` and `Subact` predicates). Since joint inference takes advantage of inter-dependencies between modes, acts and subacts, the accuracy of labeling the hidden variables is also better in the joint model as compared to the pipeline SVM classifier. The improvement in act and subact labeling was slightly smaller than the improvement we got for our main task of mode labeling. However, as shown in Table 4, here again, we observed significant improvements in both precision and recall as compared to the pipeline system.

In our final experiment, we compared results over key pedagogical steps to evaluate the effect of joint inference in these steps. These results are shown in Table 5. The mode names are quite self-explanatory for Rapport Building, Problem Identification and Assessment. Scaffolding is a concept where the tutor scaffolds the learner who is working through the solution by giving hints. Sense Making is the concept of explanations for understanding purposes. Process Negotiation is discussing/confirming the process of how to go about solving the problem. As we see from the results, in most cases, the joint model is significantly

Mode-Label	Pipeline			Joint Model		
	Precision	Recall	F1	Precision	Recall	F1
Rapport Building	0.25	0.5	0.338	0.31	0.53	0.389
Scaffolding	0.2	0.34	0.258	0.22	0.54	0.312
Problem Identification	0.246	0.42	0.31	0.265	0.42	0.325
Assessment	0.06	0.34	0.11	0.28	0.27	0.28
Process Negotiation	0.279	0.47	0.35	0.275	0.56	0.37
Sense Making	0.20	0.21	0.21	0.22	0.32	0.26

Table 5: 5-fold Cross Validation results for modes important in tutoring.

better than the pipeline system. Particularly, in some cases such as Scaffolding, which is an important step that corrects learners when they are going in the wrong direction, there was nearly a 20 percent increase in recall. As such, in almost all modes, we observed improvements in both precision and recall, which clearly illustrates the benefit of our joint model.

6 Conclusion

In this paper, we presented a novel joint inference method to detect modes in human-to-human tutoring. Specifically, modes are high level abstractions of dialogue speech acts, which give us a much deeper understanding of the underlying process by which natural language tutoring occurs. This is an important sub-step in designing Intelligent Tutoring Systems since strategies taken by expert human tutors can be adapted to AI-based tutors. In this work, we exploited inter-dependencies between lower-level dialogue acts and the higher-level modes using joint inference. Specifically, we developed a Markov Logic Network (MLN) to encode the the joint dependencies between dialogue acts, subacts and modes using weighted first-order logic formulas. We then developed a scalable MAP inference strategy for our model by partially grounding the MLN in each inference sub-step instead of pre-grounding the full MLN. We demonstrated the effectiveness of our approach on a real-world dialogue-based tutoring dataset collected from 500 students and annotated by multiple expert tutors. We showed that our MLN-based joint model outperforms a pipeline model that we built using SVMs that detects modes, acts and subacts independently of each other.

Future work includes mode detection without pre-specifying the dialogue acts and modes, i.e., automatically induce the dialogue acts and modes in the dialogue using non-parametric unsupervised machine learning methods. We will also apply joint inference to other complex sub-problems in Intelligent Tutoring Systems such as semantic similarity matching, automatically generate the best subsequent tutoring strategies, and generating hints to a student based on student response. We will also explore utilizing advanced lifted inference methods (Venugopal and Gogate, 2012; Venugopal and Gogate, 2014) in tutoring systems.

This work makes substantial contributions towards discovering effective tutorial strategies using data-driven approaches which in turn will contribute to the development of effective intelligent tutoring systems that could provide affordable, effective, one-on-one instruction to any learner of any age, anytime (24/7), anyhwhere as long as an Internet-connected device is available. The impact of such effective educational technologies will be far-reaching.

Acknowledgements

The authors would like to thank the University of Memphis for partially supporting this work. This work was also partially supported by a contract from the Advanced Distributed Learning Initiative of the United States Department of Defense (Award W911QY-15-C-0070). The authors would also like to thank the anonymous reviewers for their inputs.

References

J.L. Austin. 1962. *How to Do Things with Words*. Oxford Press.

Srinivas Bangalore and Amanda Stent. 2009. Incremental parsing models for dialog task structure. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 94–102.

Kristy Elizabeth Boyer, Eun Young Ha, Robert Phillips, Michael D. Wallis, Mladen A. Vouk, and James C. Lester. 2010. Dialogue act modeling in a complex task-oriented domain. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 297–305.

Whitney L. Cade, Jessica L. Copeland, Natalie K. Person, and Sidney K. D'Mello. 2008. Dialogue modes in expert tutoring. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS)*, pages 470–479.

Michelene T.H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G. Hausmann. 2001. Learning from human tutoring. *Cognitive Science*, 25(4):471–533.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923.

P. Domingos and D. Lowd. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool, San Rafael, CA.

Barbara Di Eugenio, Trina C. Kershaw, Xin Lu, Andrew Corrigan-Halpern, and Stellan Ohlsson. 2006. Toward a computational model of expert tutoring: A first report. In *Proeedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 503–508. AAAI Press.

Gurobi. 2013. *Gurobi Optimizer Reference Manual*. Gurobi Inc.

Hogyeong Jeong, Amit Gupta, Rod Roscoe, John Wagster, Gautam Biswas, and Daniel Schwartz. 2008. Using hidden markov models to characterize student behaviors in learning-by-teaching environments. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS)*, pages 614–625.

H. Kautz, B. Selman, and Y. Jiang. 1997. A General Stochastic Approach to Solving Problems with Hard and Soft Constraints. In D. Gu, J. Du, and P. Pardalos, editors, *The Satisfiability Problem: Theory and Applications*, pages 573–586. American Mathematical Society, New York, NY.

Tushar Khot, Niranjan Balasubramanian, Eric Gribkoff, Ashish Sabharwal, Peter Clark, and Oren Etzioni. 2015. Exploring markov logic networks for question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 685–694.

Blair Lehman, Sidney K. D'Mello, Whitney L. Cade, and Natalie K. Person. 2012. How do they do it? investigating dialogue moves within dialogue modes in expert human tutoring. In *Proceedings of the 11th International Conference on Intelligent Tutoring Systems (ITS)*, pages 557–562.

J. Marineau, Peter Wiemer-Hastings, D. Harter, B. Olde, P. Chipman, A. Karnavat, V. Pomeroy, Arthur Graesser, and the TRG. 2000. Classification of speech acts in tutorial dialog. In *Proceedings of the workshop on modeling human teaching tactics and strategies at the Intelligent Tutoring Systems 2000 conference*, pages 65–71.

Cristian Moldovan, Vasile Rus, and Arthur C. Graesser. 2011. Automated speech act classification for online chat. In *The 22nd Midwest Artificial Intelligence and Cognitive Science Conference*, pages 23–29.

Donald Morrison, Benjamin Nye, Borhan Samei, Vivek Varma Datla, Craig Kelly, and Vasile Rus. 2014. Building an intelligent pal from the tutor.com session database phase 1: Data mining. In *Educational Data Mining 2014*.

James D. Park and Adnan Darwiche. 2004. Complexity results and approximation strategies for map explanations. *J. Artif. Intell. Res. (JAIR)*, 21:101–133.

H. Poon and P. Domingos. 2007. Joint Inference in Information Extraction. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 913–918, Vancouver, Canada. AAAI Press.

H. Poon and P. Domingos. 2008. Joint Unsupervised Coreference Resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 649–658, Honolulu, HI. ACL.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Norbert Reithinger and Elisabeth Maier. 1995. Utilizing statistical dialogue act processing in verbmobil. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, pages 116–121.

Norbert Reithinger. 1995. Some experiments in speech act prediction. In *In Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse*.

Klaus Ries. 1999. Hmm and neural network based speech act detection. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.

Vasile Rus, Sidney K. D'Mello, Xiangen Hu, and Arthur C. Graesser. 2013. Recent advances in conversational intelligent tutoring systems. *AI Magazine*, 34(3):42–54.

Vasile Rus, Nobal Niraula, Nabin Maharjan, and Rajendra Banjade. 2015. Automated labelling of dialogue modes in tutorial dialogues. In *Proceedings of the Florida Artificial Intelligence Research Society Conference*, pages 205–210.

Borhan Samei, Li Li, Haiying, Fazel Keshtkar, Vasile Rus, and Arthur C. Graesser. 2014. Context-based speech act classification in intelligent tutoring systems. In *Proceeedings of The 12th International Conference on Intelligent Tutoring Systems*, pages 236–241.

Somdeb Sarkhel, Deepak Venugopal, Parag Singla, and Vibhav Gogate. 2014. Lifted MAP inference for Markov Logic Networks. *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS-14)*.

John R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

B. Selman, H. Kautz, and B. Cohen. 1996. Local Search Strategies for Satisfiability Testing. In D. S. Johnson and M. A. Trick, editors, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, pages 521–532. American Mathematical Society, Washington, DC.

Riccardo Serafin and Barbara Di Eugenio. 2004. Flsa: Extending latent semantic analysis with features for dialogue act classification. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 692–699.

David Sontag and Amir Globerson. 2011. Introduction to Dual Decomposition for Inference. *Optimization for Machine Learning*.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373.

Deepak Venugopal and Vibhav Gogate. 2012. On lifting the gibbs sampling algorithm. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1664–1672.

Deepak Venugopal and Vibhav Gogate. 2014. Evidence-based clustering for scalable inference in markov logic. In *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*, pages 258–273.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. Relieving the Computational Bottleneck: Joint Inference for Event Extraction with High-Dimensional Features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 831–843. ACL.