Learning Attribute Selections for Non-Pronominal Expressions

Pamela Jordan
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA
jordan@isp.pitt.edu

Marilyn Walker
AT&T Labs—Research
180 Park Avenue
Florham Park, NJ 07932-0971 USA
walker@research.att.com

Abstract

A fundamental function of any task-oriented dialogue system is the ability to generate nominal expressions that describe objects in the task domain. In this paper, we report results from using machine learning to train and test a nominal-expression generator on a set of 393 nominal descriptions from the COCONUT corpus of task-oriented design dialogues. Results show that we can achieve a 50% match to human performance as opposed to a 16% baseline for just guessing the most frequent type of nominal expression in the COCONUT corpus. To our surprise our results indicate that many of the central features of previously proposed selection models did not improve the performance of the learned nominal-expression generator.

1 Introduction

A fundamental function of any task-oriented dialogue system is the ability to generate nominal expressions that describe objects in the task domain. For example, consider the excerpt of a task-oriented dialogue from the COCONUT corpus in Figure 1 (Di Eugenio et al., 2000) The conversants in this dialogue are attempting to collaboratively construct a solution for furnishing a two room house. Each conversant starts the task with a set of furniture items that can be used in the solution. In the process of negotiating the solution, they generate nominal expressions (shown in italics) describing the items of furniture.

(Partial solution to problem already agreed upon in prior dialogue)
G: That leaves us with 250 dollars. I have *a yellow rug for 150 dollars.* Do you have any other furniture left that matches for 100 dollars?"
S: No, I have no furniture left that costs $100. I guess you can buy *the yellow rug for $150.*
G: Okay. I'll buy *the rug for 150 dollars.* I have *a green chair* that I can buy for 100 dollars that should leave us with no money.
S: That sounds good. Go ahead and buy *the yellow rug* and *the green chair.*
G: I'll buy *the green 100 dollar chair.* Design Complete?
S: Sounds good, do you want *the green chair* in the dining room with *the other chairs*? I put *the yellow rug* in the living room. Then the design is complete.
G: Sounds good. Hit the design complete

Figure 1: Excerpt of a COCONUT dialogue illustrating variable selection of attributes for nominal descriptions

Each furniture type in the COCONUT task domain has four associated attributes: color, price, owner and quantity. A nominal expression generator must decide which of these four attributes to include in the generated expression. For example, the task domain objects under discussion in the dialogue in Figure 1 are a $150 yellow rug owned by Garrett (G) and a $100 dollar green chair owned by Steve (S). In the dialogue excerpt in Figure 1 the yellow rug is described first as *a yellow rug for 150 dollars* and then subsequently as *the yellow rug for 150 dollars, the rug for 150 dollars, the yellow rug.* It could also have been described by any of the following non-pronominal expressions: *the rug, my rug, my*

yellow rug, my $150 yellow rug, the $150 yellow rug, the $150 rug. The content of these descriptions varies depending on which attributes are included in the description. How does the speaker decide which attributes to include?

The problem of content selection for nominal expressions has been the focus of much previous work and a large number of models have been proposed (Clark and Wilkes-Gibbs, 1986; Brennan and Clark, 1996; Dale and Reiter, 1995; Passonneau, 1995; Jordan, 2000) *inter alia.* The factors that these models utilize include the discourse structure, the attributes used in the last mention, the recency of last mention, the frequency of mention, the task structure, the inferential complexity of the task, and ways of determining salient objects and the salient attributes of an object. In this paper we utilize a set of factors considered as important for three of these models, and empirically compare the utility of these factors as predictors in a machine learning experiment. The factor sets we utilize are:

- CONTRAST SET factors, inspired by the INCREMENTAL MODEL of Dale and Reiter (1995);
- CONCEPTUAL PACT factors, inspired by the models of Clark and colleagues (Clark and Wilkes-Gibbs, 1986; Brennan and Clark, 1996);
- INTENTIONAL INFLUENCES factors, inspired by the model of Jordan (2000).

Dale and Reiter's INCREMENTAL MODEL focuses on the production of near-minimal descriptions that allow the hearer to reliably distinguish the task object from similar task objects. Following Grosz and Sidner (1986), Dale and Reiter's algorithm utilizes discourse structure as an important factor in determining which objects the current object must be distinguished from. The model of Clark, Brennan and Wilkes-Gibbs is based on the notion of CONCEPTUAL PACTS, i.e. the conversants attempt to coordinate with one another by establishing a conceptual pact for describing an object. Jordan's INTENTIONAL INFLUENCES model is based on the assumption that the underlying task-related inferences required to achieve the task goals are an important factor in content selection for non-minimal descriptions. We describe these models in more detail below.

We compare the predictive power of the factors utilized in these models by using machine learning to train and test a nominal-expression generator on a set of 393 nominal descriptions from the corpus of COCONUT dialogues. We provide the machine learner with distinct sets of features motivated by the models above, in addition to discourse features representing given-new distinctions, and dialogue specific features such as the speaker of the nominal expression, its absolute location in the discourse, and the problem that the conversants are currently trying to solve.

We evaluate the nominal-expression generator by comparing its predictions against what humans said at the same point in the dialogue. We provide a rigorous test of the nominal-expression generator by only counting as correct those nominal expressions which exactly match the content of the human generated nominal expressions.[1] We also quantify the contributions of each feature set to the performance of the nominal-expression generator. Our results show that nominal-expression generators based on a combination of the given-new, and dialogue specific and INTENTIONAL INFLUENCES features can achieve 50% accuracy at matching human performance, a significant improvement over the majority class baseline of 16% in which the generator simply guesses the most frequent property combination. In addition, to our surprise, the results indicate that the CONCEPTUAL PACT features and the CONTRAST SET features make no significant contribution to performance.

[1] While this approach is controversial, we believe that human performance is currently the only reasonable standard against which we can evaluate natural language generators (Oberlander, 1998). Note that the more attributes a discourse entity has, the harder it is to achieve an exact match to a human description, i.e. for our problem the nominal-expression generator must correctly choose among 16 possibilities represented by the power set of the four attributes.

Section 2 describes the COCONUT corpus, the encoding of the corpus and the features used in machine learning in more detail. Section 3 presents the quantitative results of testing the learned rules against the corpus, discusses the features that the machine learner identifies as important, and provides examples of the rules that are learned. Section 4 summarizes our results and discusses future work.

2 Corpus, Data, Methods

Our experiments utilize the rule learning program RIPPER (Cohen, 1996) to learn a nominal-expression generator from the nominal expressions in the COCONUT corpus. Although we had several learners available to us, we chose RIPPER primarily because the if-then rules that are used to express the learned nominal generator model are easy for people to understand and thus facilitate comparison with the theoretical models we are trying to evaluate. Like other learning programs, RIPPER takes as input the names of a set of *classes* to be learned, the names and ranges of values of a fixed set of *features*, and *training data* specifying the class and feature values for each example in a training set. Its output is a *classification model* for predicting the class of future examples. In RIPPER, the classification model is learned using greedy search guided by an information gain metric, and is expressed as an ordered set of if-then rules.

Thus to apply RIPPER, the nominal expressions in the corpus must be encoded in terms of a set of classes (the output classification) and a set of input features that are used as predictors for the classes. As mentioned above, we are trying to learn which of a set of content attributes should be included in a nominal expression. The features we encode for each nominal expression are motivated by factors claimed in the literature to be important predictors of the content of a nominal expression.

Below we describe our corpus of nominal expressions, the assignment of classes to each nominal expression, the extraction of features from the dialogue in which each expression occurs, and our learning experiments.

2.1 Corpus

The COCONUT corpus is a set of 24 computer-mediated dialogues consisting of a total of 1102 utterances. The dialogues were collected in an experiment where two human subjects collaborated on a simple design task, that of buying furniture for two rooms of a house (Di Eugenio et al., 1998). An excerpt of a COCONUT dialogue was given in Figure 1. The participants' main goal is to negotiate the purchases; the items of highest priority are a sofa for the living room and a table and four chairs for the dining room. The participants also have specific secondary goals which further constrain the problem solving task. Participants are instructed to try to meet as many of these goals as possible, and are motivated to do so by rewards associated with satisfied goals. The secondary goals are: 1) match colors within a room, 2) buy as much furniture as you can, 3) spend all your money. The participants are told which rewards are associated with achieving each goal.

Each participant is given a separate budget and inventory of furniture. Neither participant knows what is in the other's inventory or how much money the other has. By sharing information during the conversation, they can combine their budgets and select furniture from each other's inventories. The participants are equals and purchasing decisions are joint. In the experiment, each set of participants solved one to three scenarios with varying inventories and budgets. The problem scenarios varied task complexity by ranging from tasks where items are inexpensive and the budget is relatively large to tasks where the items are expensive and the budget relatively small.

After the corpus was collected it was annotated by human coders for two types of features. The DISCOURSE ENTITY LEVEL annotations provide discourse reference information from which initial representations of discourse entities and updates to them can be derived, and explicit attribute usage information that reflects how each discourse entity

was evoked. For example, the initial representation for "I have a yellow rug. It costs $150." would include type, quantity, color and owner following the first utterance. Only the quantity attribute is inferred. After the second utterance the entity would be updated to include price. The UTTERANCE LEVEL ANNOTATIONS capture the problem solving state in terms of goals, constraint changes and the size of the solution set for the current constraint equations as well as current variable assignments. The utterance level discourse features encode when an offer is made and the level of a speaker's commitment to a proposal under consideration, i.e. conditional or unconditional.

In order to derive some of the discourse information the task structure must be identified. The COCONUT corpus was encoded via a set of instructions to coders to record all domain goals. Changes to a different domain goal or action were used as a cue to derive the non-linguistic task structure (Terken, 1985; Grosz and Sidner, 1986). Each domain action provides a discourse segment purpose so that each utterance that relates to a different domain action or set of domain actions defines a new segment. The encoded features all have good intercoder reliability (Di Eugenio et al., 1998; Jordan, 2000).

Our experimental data is 393 non-pronominal nominal descriptions from 13 dialogues of the COCONUT corpus as well as features constructed from the annotations described above. We explain how we use the annotations to construct the features in more detail below.

2.2 Class Assignment

The corpus of nominal expressions is used to construct the machine learning classes as follows.

We are trying to learn which subset of the four attributes, color, price, owner, quantity, should be included in a nominal expression. We encode each nominal expression in the corpus as a member of the category represented by the set of properties expressed by the nominal expression. This results in 16 classes representing the power set of the four attributes.

2.3 Feature Extraction

The corpus is used to construct the machine learning features as follows. In RIPPER, feature values are continuous (numeric), set-valued, or symbolic. We encoded each non-pronominal description in terms of a set of 58 features that were either directly annotated by humans as described above, derived from annotated features or inherent to the dialogue (Di Eugenio et al., 1998; Jordan, 2000). The dialogue context in which each description occurs is represented in the encodings.

- what is mutually known: type-mk, color-mk, owner-mk, price-mk, quantity-mk
- reference-relation: one of `initial, coref, inference`

Figure 2: Given-New Feature Set.

The GIVEN-NEW features in Figure 2 encode fundamental attributes of the entity that is to be described by the nominal expression (Clark and Marshall, 1981; Prince, 1981). We encode what is mutually known about the discourse entity at the point at which it is to be described (*type-mk, color-mk, owner-mk, price-mk, quantity-mk*). We utilize a *reference-relation* feature to encode whether the entity is new (`initial`), given (`coref`) or discourse inferred (`inference`) relative to the discourse history. The types of inferences supported by the annotation are set, subset, class and common noun anaphora (e.g. one and null anaphora) (Jordan, 2000).

The INHERENT FEATURES in Figure 3 are a specific encoding of particulars about the discourse situation, such as the speaker, the task, and the entity's known attributes (*type, color, owner, price, quantity*). While we don't expect this feature set to generalize to other dialogue situations, it allows us to examine whether there are individual differences in attribute selection algorithms (*speaker, speaker-pair*), or whether specifics about the properties of the object, the location within the dialogue (*utterance-number*), and the prob-

- utterance-number, speaker-pair, speaker, problem-number
- attribute values:
  - type: one of `sofa, chair, table, rug, lamp`
  - color: one of `red, blue, green, yellow`
  - owner: one of `self, other`
  - price: ranged from $50 to $600
  - quantity: ranged from 0 to 4.

Figure 3: INHERENT Feature Set: Task, Speaker and Discourse Entity Specific features.

- interactions with other discourse entities: distance-last-ref, distance-last-ref-in-turns, number-prev-mentions, speaker-of-last-ref
- previous description: color-in-last-exp, type-in-last-exp, owner-in-last-exp, price-in-last-exp, quantity-in-last-exp, type-in-last-turn, color-in-last-turn, owner-in-last-turn, price-in-last-turn, quantity-in-last-turn, initial-in-last-turn

Figure 4: CONCEPTUAL PACT Feature Set.

lem difficulty (*problem-number*) play significant roles in attribute selection.

The CONCEPTUAL PACT model suggests that dialogue participants negotiate a description that both find adequate for describing an object (Clark and Wilkes-Gibbs, 1986; Brennan and Clark, 1996). The speaker generates trial descriptions that the hearer modifies based on which object he thinks he is suppose to identify. The negotiation continues until the participants are confident that the hearer has correctly identified the intended object. The additional features suggested by this model include the previous description since that is the description that will be modified, and how long ago the description was made. If the description were made further back in the dialogue, that would indicate that the negotiation process had been completed. Furthermore, the model suggests that, once a pact has been reached, that the dialogue participants will continue to use the description that they previously negotiated. This aspect of the model is also similar to Passonneau's LEXICAL FOCUS model (Passonneau, 1995).

The CONCEPTUAL PACT features in Figure 4 encode how the current description relates to previous descriptions of the same entity. We encode when the entity was last described in terms of number of utterances and turns (*distance-last-ref, distance-last-in-turns*), how frequently it was described (*number-prev-mentions*), who last described it (*speaker-of-last-ref*), and how it was last described in terms of turn and expression since the description may have been broken into several utterances (*color-in-last-exp, type-in-last-exp, owner-in-last-exp, price-in-last-exp, quantity-in-last-exp, type-in-last-turn, color-in-last-turn, owner-in-last-turn, price-in-last-turn, quantity-in-last-turn, initial-in-last-turn*).

- ONE UTTERANCE Distractors: type-distractors, color-distractors, owner-distractors, price-distractors, quantity-distractors
- SEGMENT Distractors: type-distractors, color-distractors, owner-distractors, price-distractors, quantity-distractors

Figure 5: CONTRAST SET Feature Sets

The INCREMENTAL MODEL builds a description incrementally by considering the other objects that are currently expected to be in focus for the hearer(Dale and Reiter, 1995). These other objects are called *distractors*. The basic idea is to add attributes as necessary until any distractors are ruled out as competing co-specifiers. Based on these ideas, we developed a set of features we call CONTRAST SET features, as in Figure 5. The goal of our encoding is to represent whether there are distractors present in the focus space which might motivate the inclusion of a particular attribute.[2]

[2]This representatation only approximates the model since the INCREMENTAL model utilizes a preferred salience ordering of attributes and eliminates distractors as attributes are added to a description. For example, adding the attribute *type* when the object is a chair, eliminates any distractors that aren't chairs. Our encoding treats attributes instead of objects as distractors. This view has the advantage that the preferred ordering of attributes could adjust according to the focus space and this interpretation of Dale and Reiter's model was shown in (Jordan, 2000) to perform similarly to the strict model. However, the feature representation is still impoverished with respect to (Jordan, 2000) since it doesn't capture what the most salient attribute values for the focus space are.

An open issue with deriving the distractors is how to define a focus space (Walker, 1996). We use two focus space definitions, one based on recency, and the other on intentional structure. See Figure 5. For intentional structure we utilize the task goal segmentation encoded in the COCONUT corpus as discussed above (SEGMENT). For recency, we simply consider the entities from the previous utterance as possible distractors (ONE UTTERANCE). For each focus space definition, we encode whether the attribute value of the item to be described is the same as at least one other item in the focus space (*type-distractors, color-distractors, owner-distractors, price-distractors, quantity-distractors*).

- task situation: goal, colormatch, colormatch-constraintpresence, pricelimit, pricelimit-constraintpresence, priceevaluator, priceevaluator-constraintpresence, colorlimit, colorlimit-constraintpresence, priceupperlimit, priceupperlimit-constraintpresence
- agreement state: influence-on-listener, commit-speaker, solution-size, prev-influence-on-listener, prev-commit-speaker, prev-solution-size, distance-of-last-state-in-utterances, distance-of-last-state-in-turns, ref-made-in-prev-action-state, speaker-of-last-state
- solution interactions: color-contrast, price-contrast

Figure 6: Intentional Influences Feature Set.

Jordan (Jordan, 2000) proposed a model to select attributes for nominals called the INTENTIONAL INFLUENCES model. This model posits that the task-related inferences and the agreement process for task negotiation are important factors in selecting attributes. The features used to approximate Jordan's model are in Figure 6. The task situation features encode inferrable changes in the task situation that are related to item attributes. The agreement state features encode critical points of agreement during problem solving. For example, if a dialogue participant is *accepting* a proposal, she may want to verify that she has the same item and the same entity description as her partner. These are features that (Di Eugenio et al., 2000) found to be indicative of agreement states and include DAMSL features (*influence-on-listener, commit-speaker, prev-influence-on-listener, prev-commit-speaker*) (Allen and Core, 1997), progress towards a solution (*solution-size, prev-solution-size, ref-made-in-prev-action-state*), and features inherent to an agreement state (*speaker-of-last-state, distance-of-last-state-in-utterances, distance-of-last-state-in-turns*). The solution interactions features represent situations where multiple proposals are under consideration which may contrast with one another in terms of solving color-matching goals (*color-contrast*) or price related goals (*price-contrast*).

2.4 Learning Experiments

The final input for learning is training data, i.e., a representation of a set of nominal expressions in terms of feature and class values. In order to induce rules from a variety of feature representations, our training data is represented differently in different experiments. First, examples are represented using only the GIVEN-NEW features in Figure 2 to establish a performance baseline for given-new information. Then other feature sets are added in to examine their individual contribution, culminating with the full feature set.

The output of each machine learning experiment is a model for nominal expression generation for this domain and task, learned from the training data. To evaluate these models, the error rates of the learned models are estimated using 25-fold cross-validation, i.e. the total set of examples is randomly divided into 25 disjoint test sets, and 25 runs of the learning program are performed. Thus, each run uses the examples not in the test set for training and the remaining examples for testing.

3 Experimental Results

Table 1 summarizes our experimental results. For each feature set, we report accuracy rates and standard errors resulting from cross-validation.[3] It is clear that performance de-

[3]Accuracy rates are statistically significantly different when the accuracies plus or minus twice the standard error do not overlap (Cohen, 1995), p. 134.

pends on the features that the learner has available. The 16.3% MAJORITY CLASS BASELINE accuracy rate in the first row is a standard baseline that corresponds to the accuracy one would achieve from simply choosing the description type that occurs most frequently in the corpus, which in this case means that the nominal-expression generator would always use the color, price and quantity to describe a domain entity.

Feature Sets Used	Accuracy (SE)
MAJORITY CLASS BASELINE	16.3 %
GIVEN-NEW	18.2% (2.3)
GIVEN-NEW,SEG	18.5% (2.5)
GIVEN-NEW,CP	20.3% (2.5)
GIVEN-NEW,IINF	33.1% (2.7)
GIVEN-NEW,IINF,CP,SEG	33.6 % (2.2)
GIVEN-NEW, INH	42.6% (2.7)
GIVEN-NEW,IINF,INH	47.9% (2.0)
GIVEN-NEW,IINF,INH,CP	50.1% (2.2)
GIVEN-NEW,IINF,INH,CP,SEG	48.2% (2.9)
GIVEN-NEW,IINF,INH,CP,1UTT	46.0% (1.9)

Table 1: Accuracy rates for the Nominal Generator using different feature sets, SE = Standard Error. CP = the CONCEPTUAL PACT features. IINF = the INTENTIONAL INFLUENCES features. INH = the INHERENT features. SEG = the CONTRAST-SET, SEGMENT features. 1UTT = the CONTRAST SET, ONE UTTERANCE features.

The row of Table 1 labelled GIVEN-NEW shows that providing the learner with information about whether the values of the attributes for a discourse entity are mutually known does not, in and of itself, improve performance over the baseline. Similarly, the rows labelled GIVEN-NEW,SEG and GIVEN-NEW,CP show that providing the features for contrast set and conceptual pact does not statistically improve performance over the baseline. The GIVEN-NEW,IINF and GIVEN-NEW,IINF,CP,SEG rows show that adding INTENTIONAL INFLUENCES features **does** provide a significant performance improvement, but allowing the learner to learn rules that would combine features from the INTENTIONAL INFLUENCES features, the CONTRAST SET features and the CONCEPTUAL PACT features does not significantly improve performance over just having the INTENTIONAL INFLUENCES features alone. Figure 7 shows the rules that are learned for the generation of nominal expressions given the GIVEN-NEW and INTENTIONAL INFLUENCES features.

The row labelled GIVEN-NEW,INH in Table 1 shows that, if we were only interested in doing well in this domain, that adding discourse entity and task specific information **does** improve the performance of the learned nominal-expression generator. This is at the cost of losing generality in the rules that are learned. Figure 8 shows that the generation rules learned given access to the INHERENT feature set make use of many discourse entity, task, and speaker specific features. The speaker-pair feature alone is used in seven of the learned rules.

The GIVEN-NEW,IINF,INH row in Table 1 suggests that interesting rules can be learned by adding the INTENTIONAL INFLUENCES features to the INHERENT features, but the performance improvement over the INHERENT feature set is not significant.

The remainder of the table shows that the ability to utilize all of the features provides a slight performance improvement which however is not statistically significant. The last two rows suggest that adding in features representing various views of discourse segmentation do not contribute to performance. Figure 8 shows the generation rules learned with the best performing features set shown in the row labelled GIVEN-NEW,IINF,INH,CP. As mentioned above, many task, entity and speaker specific features are used in these rules. However, this rule set performs at 50% accuracy, as opposed to 33.6% accuracy for our most general feature set (shown in the row labelled GIVEN-NEW,IINF,CP,SEG).

4 Discussion and Future Work.

While previous research in natural language generation has applied machine learning to accent placement, cue-word selection, text planning, and determining the form of a nominal expression (Hirschberg, 1993; Moser and Moore, 1995; Mellish et al., 1998; Poesio, 2000; Strube and Wolters, 2000), we know of

Say POQ **if** (priceupperlimit-constraintpresence=IMPLICIT) ∧ (goal=SELECTCHAIRS)
Say PO **if** (pricelimit=yes)
Say PO **if** (priceupperlimit-constraintpresence=IMPLICIT) ∧ (goal=SELECTSOFA)
Say CQ **if** (color-mk=yes) ∧ (influence-on-listener=na) ∧ (distance-of-last-state-in-utterances≥4) ∧ (distance-of-last-state-in-utterances≤4)
Say CO **if** (colorlimit=yes)
Say CO **if** (price-mk=yes) ∧ (prev-solution-size=INDETERMINATE) ∧ (ref-made-in-prev-action-state=no) ∧ (solution-size=INDETERMINATE)
Say CO **if** (distance-of-last-state-in-utterances≥9) ∧ (distance-of-last-state-in-turns≤1)
Say O **if** (influence-on-listener=info-request) ∧ (distance-of-last-state-in-turns≤0)
Say O **if** (prev-influence-on-listener=open-option) ∧ (quantity-mk=yes) ∧ (prev-solution-size=INDETERMINATE)
Say CP **if** (solution-size=INDETERMINATE) ∧ (influence-on-listener=na) ∧ (price-contrast=yes) ∧ (distance-of-last-state-in-turns≥2)
Say T **if** (prev-solution-size=DETERMINATE) ∧ (color-contrast=no) ∧ (distance-of-last-state-in-utterances≥3) ∧ (prev-influence-on-listener=na)
Say T **if** (prev-solution-size=DETERMINATE) ∧ (colormatch-constraintpresence=EXPLICIT)
Say T **if** (ref-made-in-prev-action-state=yes) ∧ (distance-of-last-state-in-turns≤0) ∧ (color-mk=yes)
Say T **if** (influence-on-listener=info-request)
Say T **if** (priceevaluator=yes)
Say CPOQ **if** (distance-of-last-state-in-utterances≥5) ∧ (speaker-of-last-state=OTHER) ∧ (color-contrast=no)
Say CPOQ **if** (goal=SELECTCHAIRS) ∧ (prev-solution-size=INDETERMINATE) ∧ (ref-made-in-prev-action-state=no) ∧ (distance-of-last-state-in-utterances≤3)
Say CPO **if** (influence-on-listener=action-directive) ∧ (reference-relation=initial)
Say CPO **if** (goal=SELECTSOFA) ∧ (distance-of-last-state-in-utterances≥2)
Say CPO **if** (ref-made-in-prev-action-state=no) ∧ (goal=SELECTTABLE) ∧ (distance-of-last-state-in-turns≥1)
Say CPO **if** (prev-influence-on-listener=action-directive) ∧ (goal=SELECTTABLE)
Say CPO **if** (influence-on-listener=open-option) ∧ (distance-of-last-state-in-utterances≤0)
default **Say** CPQ

Figure 7: Rules Learned Using GIVEN-NEW and INTENTIONAL INFLUENCES Features. The classes encode the four attributes, e.g CPOQ = Color,Price,Owner and Quantity, T = Type only

no other work applying machine learning to determining the content of a nominal expression. In our experiments, we train a nominal-expression generator to learn which attributes of an entity to include in a nominal description from a corpus of dialogues. Our results show that:

- Our best performing learned nominal-expression generator can achieve a 50% match to human performance as opposed to a 16% baseline;
- The intentional influences features developed to approximate Jordan's intentional influences model do significantly improve performance;
- Features specific to the task, speaker and discourse entity also provide significant performance improvements;
- Surprisingly, the use of given-new, contrast set and conceptual pact features do not improve performance.

We might have expected to achieve a best-performing accuracy higher than 50% but as this is the first study of this kind, there are several issues to consider. First, the nominal expressions in the corpus may represent just **one** way to describe the entity at that point in the dialogue, so that using human performance as a standard against which to evaluate the learned nominal-expression generators provides an overly rigorous test (Oberlander, 1998). Furthermore, we do not know whether humans would produce identical nominal expressions given the same discourse situation. A previous study of anaphor generation in Chinese showed that rates of match for human speakers averaged 74% for that problem (Yeh and Mellish, 1997), and our results show that including speaker-specific features improves performance significantly. Our conclusion is that it may be important to quantify the best performance that a human could achieve at matching the nominal expressions in the corpus, given the complete discourse context and

Say PO **if** (color=unk) ∧ (owner=SELF) ∧ (speaker-pair=GARRETT-STEVE)
Say OQ **if** (color=unk) ∧ (quantity≥2)
Say OQ **if** (distance-of-last-state-in-turns=2) ∧ (distance-last-ref-in-turns≥31) ∧ (type=CHAIR)
Say COQ **if** (quantity≥2) ∧ (price=unknown)
Say COQ **if** (quantity≥2) ∧ (type-in-last-exp=no)
Say CQ **if** (speaker-pair=DAVE-GREG) ∧ (distance-last-ref-in-turns≥15) ∧ (distance-last-ref≥4)
Say C **if** (prev-commit-speaker=commit) ∧ (utterance-number≥43)
Say C **if** (price-in-last-turn=no) ∧ (utterance-number≤21)
Say CO **if** (price=unknown) ∧ (utterance-number≥16)
Say CO **if** (quantity-in-last-exp=no) ∧ (distance-last-ref-in-turns≤18) ∧) (utterance-number≥18) ∧ (price≥325)
Say CO **if** (price-in-last-exp=yes) ∧ (speaker-pair=JILL-PENNY)
Say CO **if** (priceupperlimit=yes)
Say O **if** (speaker-pair=GARRETT-STEVE) ∧ (speaker-of-last-state=SELF) ∧ (color-contrast=no)
Say O **if** (color=unk) ∧ (owner=OTHER) ∧ (price≤300)
Say T **if** (prev-solution-size=DETERMINATE) ∧ (price≥250) ∧ (color-contrast=no)
Say T **if** (color=unk)
Say T **if** (distance-last-ref-in-turns≥10) ∧ (speaker-pair=KATHY-MARK) ∧ (distance-last-ref≤2)
Say CP **if** (utterance-number≤5) ∧ (utterance-number≥3) ∧ (quantity≤1)
Say CP **if** (distance-of-last-state-in-utterances='4') ∧ (problem≥2) ∧ (price-contrast=yes)
Say CP **if** (quantity≤-1)
Say CP **if** (speaker=KRISTI) ∧ (reference-relation=inference)
Say CPOQ **if** (goal=SELECTCHAIRS) ∧ (prev-solution-size=INDETERMINATE)
Say CPOQ **if** (speaker-pair=KATHY-MARK) ∧ (prev-solution-size=INDETERMINATE) ∧) (reference-relation=initial)
Say CPOQ **if** (goal=SELECTCHAIRS) ∧ (problem≤1)
Say CPO **if** (goal=SELECTSOFA)
Say CPO **if** (problem≤1) ∧ (utterance-number≥7) ∧ (reference-relation=initial) ∧ (price≥150)
Say CPO **if** (speaker-pair=KATHY-MARK) ∧ (quantity≤1)
Say CPO **if** (utterance-number≥77)
default **Say** CPQ

Figure 8: The best performing rule set, learned using the combination of the GIVEN-NEW, INTENTIONAL INFLUENCES, INHERENT, and CONCEPTUAL PACT feature sets. The classes encode the four attributes, e.g., CPOQ = Color,Price,Owner and Quantity, T = Type only.

the identity of the referent. In addition, the difficulty of this problem depends on the number of attributes available for describing an object in the domain; our nominal expression generator has to correctly make four different decisions to achieve an exact match to human performance. Finally, the COCONUT corpus is publicly available, and other researchers can now attempt to improve on our results.

One of the most surprising results of our study is the finding that many of the theoretically motivated features previously proposed in the literature do not improve performance on our task. However, in previous work, Jordan (Jordan, 2000) utilized the COCONUT corpus to develop a rule-based model of nominal expression generation for *redescriptions*, i.e. the subset of our data where the reference relation is `coref`. Jordan also found that varying how contrast sets are derived made no significant difference in performance and that the CONCEPTUAL PACTS model was significantly worse than the INTENTIONAL INFLUENCES model. In future work, we plan to perform similar experiments on different corpora with different communications settings and problem types (e.g. planning, scheduling, designing) to determine whether our findings are specific to the genre of dialogues that we examine here, or whether they are more general. We also intend to develop other feature sets to provide additional approximations to these models.

References

James Allen and Mark Core. 1997. Draft of DAMSL: Dialog act markup in several layers.

Susan E. Brennan and Herbert H. Clark. 1996. Lexical choice and conceptual pacts in conversation. *Journal of Experimental Psychology: Learning, Memory And Cognition.*

Herbert H. Clark and Catherine R. Marshall. 1981. Definite reference and mutual knowledge. In Joshi, Webber, and Sag, editors, *Elements*

of Discourse Understanding, pages 10–63. CUP, Cambridge.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

Paul R. Cohen. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press, Boston.

William Cohen. 1996. Learning trees and rules with set-valued features. In *Fourteenth Conference of the American Association of Artificial Intelligence*.

Robert Dale and Ehud Reiter. 1995. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, 19(2):233–263, Apr–June.

Barbara Di Eugenio, Pamela W. Jordan, Johanna D. Moore, and Richmond H. Thomason. 1998. An empirical investigation of collaborative dialogues. In *ACL-COLING98, Proceedings of the Thirty-sixth Conference of the Association for Computational Linguistics*, Montreal, Canada, August.

Barbara Di Eugenio, Pamela W. Jordan, Richmond H. Thomason, and Johanna D. Moore. 2000. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogues. *To Appear in International Journal of Human-Computer Studies*.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204.

Julia B. Hirschberg. 1993. Pitch accent in context: predicting intonational prominence from text. *Artificial Intelligence Journal*, 63:305–340.

Pamela W. Jordan. 2000. *Intentional Influences on Object Redescriptions in Dialogue: Evidence from an Empirical Study*. Ph.D. thesis, Intelligent Systems Program, University of Pittsburgh.

Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O'Donnell. 1998. Experiments using stochastic search for text planning. In *Proceedings of International Conference on Natural Language Generation*, pages 97–108.

Margaret G. Moser and Johanna Moore. 1995. Investigating cue selection and placement in tutorial discourse. In *ACL 95*, pages 130–137.

Jon Oberlander. 1998. Do the right thing...but expect the unexpected. *Computational Linguistics*, 24(3):501–508.

Rebecca J. Passonneau. 1995. Integrating Gricean and Attentional Constraints. In *Proceedings of IJCAI 95*.

Massimo Poesio. 2000. Annotating a corpus to develop and evaluate discourse entity realization algorithms: issues and preliminary results. In *Proc. Language Resources and Evaluation Conference, LREC-2000*.

Ellen F. Prince. 1981. Toward a taxonomy of given-new information. In *Radical Pragmatics*, pages 223–255. Academic Press.

Michael Strube and Maria Wolters. 2000. A probabilistic genre-independent model of pronominalization. In *Proceedings of the North American Meeting of the Association for Computational Linguistics*, pages 18–25.

J. M. B. Terken. 1985. *Use and Function of Accentuation: Some Experiments*. Ph.D. thesis, Institute for Perception Research, Eindhoven, The Netherlands.

Marilyn A. Walker. 1996. Limited attention and discourse structure. *Computational Linguistics*, 22-2:255–264.

Ching-Long Yeh and Chris Mellish. 1997. An empirical study on the generation of anaphora in chinese. *Computational Linguistics*, 23-1:169–190.