Analyzing and Predicting Patterns of DAMSL Utterance Tags

Mark G. Core
Department of Computer Science
University of Rochester
Rochester, NY 14627
mcore@cs.rochester.edu

Abstract

We have been annotating TRAINS dialogs with dialog acts in order to produce training data for a dialog act predictor, and to study how language is used in these dialogs. We are using DAMSL dialog acts which consist of 15 independent attributes. For the purposes of this paper, infrequent attributes such as Unintelligible and Self-Talk were set aside to concentrate on the eight major DAMSL tag sets. For five of these eight tag sets, hand constructed decision trees (based solely on the previous utterance's DAMSL tags) did better than always guessing the most frequent DAMSL tag values. This result suggests that it is possible to automatically build such decision trees especially if other sources of context are added. Our initial efforts to address our second goal (studying language use in the TRAINS dialogs) consist of measuring DAMSL tag co-occurrences and bigrams. Some interesting patterns have emerged from this simple analysis such as the fact that signaling non-understanding is often done through questions. These patterns suggest that we should also be considering an n-gram dialog act model for use in predicting DAMSL tags.

Machine learning techniques have been successfully applied to speech recognition, part of speech tagging, word sense disambiguation, and parsing. One reason for this success is that researchers have been able to define "correct" answers even for word sense disambiguation (Miller *et al.* 1993) and parsing (Marcus, Santorini, & Marcinkiewicz 1992). DAMSL (Dialog Act Markup in Several Layers) is a system for labeling the "correct" dialog acts for an utterance. We are annotating dialogs with DAMSL tags in order to produce training data for a dialog act predictor, and to study how language is used in our dialog corpus.

We start this paper with a brief description of DAMSL as well as a discussion of its suitability for encoding the "correct" dialog acts for an utterance. The second section of the paper discusses the motivation for using machine learning to automatically predict DAMSL tags, and the third section describes an initial attempt at constructing decision trees to predict DAMSL tags. The fourth section covers interesting tag co-occurrences and bigrams showing a few ways language is used in these dialogs and suggesting that a n-gram dialog act model may be the best approach for predicting DAMSL tags.

DAMSL

Searle's taxonomy of illocutionary acts (Searle 1975) has been highly influential on the set of dialog acts recognized by computational systems. Two examples are the VERBMOBIL dialog system (Reithinger & Maier 1995) and the TRAINS dialog system (Ferguson, Allen, & Miller 1996). Both have dialog acts fitting under Searle's *Directives* label such as *Request-Suggest* in VERBMOBIL and *Request* in TRAINS. However, with other dialog acts such as VERBMOBIL's *Digress* act or TRAIN's *Acknowledge Apology* it is difficult to see the parallelism between the systems or their relation to Searle's taxonomy.

Different types of dialogs, being analyzed for different purposes will lead to specialized dialog acts such as Digress and Acknowledge Apology. However, research groups can agree to a common set of higher-level dialog acts so they can share annotated dialogs, increasing the amount of training data available for dialog act recognizers. For example, TRAINS and VERBMOBIL might agree to a general acknowledgment dialog act under which Acknowledge Apology would fit as well as VERBMOBIL's *Feedback* dialog act.

The Multiparty Discourse Group was formed to develop such a set of dialog acts. It has meet twice at meetings of the Discourse Research Initiative (DRI).[1] The DAMSL utterance annotation scheme was developed from this effort and includes traditional illocutionary acts as well as a label of the utterance's general topic and how it responds to previous utterances. The annotation manual describing the la-

[1]See the DRI home page for more details: http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html

bels of DAMSL and when they apply is available at "ftp://ftp.cs.rochester.edu/pub/packages/dialog-annotation/manual.ps.gz". Note this is a work in progress; the manual has been changed based on annotation work done with DAMSL and should not be taken as the final word on the DRI scheme. Continued work with DAMSL as well as subsequent DRI meetings will likely result in refinement of the scheme.

DAMSL has three main layers: Forward Communicative Functions, Backward Communicative Functions, and Information Level. The Forward Communicative Functions consist of a taxonomy in a similar style as the actions of traditional speech act theory. The Backward Communicative Functions indicate how the current utterance relates to the previous dialog, such as accepting a proposal, confirming understanding, or answering a question. Information Level encodes whether the utterance deals with the dialog task, the communication process, or metalevel discussion about the task.

Forward Communicative Function

The main Forward Communicative Functions are shown below. Each function is labeled independently of the others, so, for example, an utterance can be a statement as well as a question. Statements are defined as claims about the world, and are further subdivided based on whether the speaker is trying to affect the beliefs of the hearer, or is repeating information for emphasis or acknowledgment. Directives fit under the more general category, Influencing-Addressee-Future-Action, which includes all utterances that discuss potential actions of the addressee. Directives are subdivided into two categories: *Info-Request*, which consists of questions and requests such as "tell me the time", and *Action-Directive*, which covers requests for action such as "please take out the trash" and "close the door". Influencing-Addressee-Future-Action also includes *Open-Option* where a speaker gives a potential course of action but does not show preference toward it, "how about going to Joey's Pizza". Commissives are given the more descriptive name, Committing-Speaker-Future-Action, and are subdivided into *Offers* and *Commit*(ments). The *Other Forward Function* category is a default choice for communicative actions that influence the future of the dialog in a way not captured by the other categories. Sentence initial words such as "okay" are often separated into separate utterances and marked as Other Forward Function. These words may have Forward Communicative Functions such as signaling a repair or change in topic or holding the turn (while the person is thinking) as well as Backward Communicative Functions such as Accepting and Acknowledging. Future work in this annotation effort will include developing classes of Other Forward Functions.

- Statement
  - Assert
  - Reassert
  - Other-Statement
- Influencing Addressee Future Action
  - Open-option
  - Directive
    - Info-Request
    - Action-Directive
- Committing Speaker Future Action
  - Offer
  - Commit
- Other Forward Function

Backward Communicative Function

The Backward Communicative Functions of DAMSL are shown below. The classes Agreement, Understanding, and Answer are independent so an utterance may simultaneously accept information and acknowledge that the information was understood as well as answer a question.

Agreement has several subclasses; *Accept* and *Reject* refer to fully accepting or rejecting an utterance or set of utterances. *Accept-Part* and *Reject-Part* refer to partially accepting or rejecting a proposal. *Hold* refers to utterances such as clarification questions that delay the listener's reaction to a proposal or question. *Maybe* refers to cases where the listener refuses to make a judgment at this point. The examples in figure 1 illustrate each type of agreement in response to the offer "Would you like the book and its review?".

```
Context:     A: Would you like the book and
                its review?
Accept       B: Yes please.
Accept-Part  B: I'd like the book.
Maybe        B: I'll have to think about it
                (intended literally)
Reject-Part  B: I don't want the review.
Reject       B: No thank you.
Hold         B: Do I have to pay for them?
```

Figure 1: Example annotations using the Agreement Label

The Understanding dimension concerns whether the listener understood the speaker. The listener may signal understanding or non-understanding or attempt to correct the speaker (showing that they either did not understand or that they did understand but that the speaker misspoke). Non-understanding can be indicated by utterances such as "huh?", clarification questions ("To Dansville?") and by explicit questions about what the speaker said or meant. Understanding can be indicated by acknowledgments such as "right" or "okay", by repeating some of the speaker's utterance, or by continuing or completing the speaker's sentence.

The Answer dimension indicates that an utterance is supplying information explicitly requested by a previous Info-Request act. This is a highly specific function that you might expect could be generalized into some other form of response, but we have not as yet been able to identify what the generalization would be.

- Agreement
  - Accept
  - Accept-Part
  - Maybe
  - Reject-Part
  - Reject
  - Hold
- Understanding
  - Signal-Non-Understanding
  - Signal-Understanding
    - Acknowledge
    - Repeat-Rephrase
    - Completion
  - Correct-Misspeaking
- Answer

Information Level

The third part of DAMSL is the Information Level annotation and encodes whether the utterance deals with the dialog task, the communication process, or metalevel discussion about the task. This dimension eliminates the need to have tags such as Communication-Info-Request, for utterances such as "What did you say?", and Task-Info-Request for utterances such as "What times are available?". With this information, we can identify three independent subdialogs within a single dialog. The topic motivating the dialog is developed and discussed in the Task part of the dialog. The Task-Management part of a dialog involves explicit planning and monitoring of how well the task is being accomplished. The physical requirements of the dialog (such as being able to hear one another) are maintained in the Communication-Management part of the dialog. Note that in some sense all utterances have a Communication-Management component. It is only marked, however, when the utterance has no Task or Task Management component.

- Information Level
    - Task
    - Task Management
    - Communication Management
    - Other

Inter-Annotator Reliability

Two goals of labeling dialogs with DAMSL tags are to provide training data for dialog systems and to gain a better knowledge of how phenomena such as "acceptances" are used. Such uses require reliably labeled dialogs; otherwise there will not be consist patterns to identify. The kappa statistic (Siegel & Castellan 1988) for inter-annotator reliability is becoming the standard for annotations of this type. According to (Carletta 1996) even for tentative conclusions to be drawn, kappas must be above 0.67 with above 0.8 being considered reliable.

(Core & Allen 1997) shows kappa scores from annotations of eight TRAINS dialogs by University of Rochester (undergraduate and graduate) students. The TRAINS dialogs which are also used in the experiments of this paper are discussions between humans on solving transportation problems involving trains (Heeman & Allen 1995). Most of the kappa scores of these eight TRAINS dialogs are close to or above the 0.67 limit discussed above. (Core & Allen 1997) discusses reasons for annotator disagreement; most of these issues have been addressed in the newest version of the DAMSL annotation manual. The next round of annotations should have higher kappa scores as the manual now contains examples and instructions for cases where annotators would often differ in labeling. In the meantime, some tentative conclusions can be drawn about the TRAINS dialogs based on the current data.

Automatically Predicting DAMSL Tags

Can machine learning techniques predict DAMSL tags well enough to be useful in a dialog system? (Cohen & Perrault 1979) introduced the notation of formulating dialog acts as plan operators. Such an architecture allows the system to perform plan recognition to determine the appropriate dialog acts for an utterance. The major problem with such an architecture is that the planner must sort through a large space of communicative plans and may also have to perform planning related to the domain (the dialog may involve constructing plans). (Allen & Perrault 1980) talk about heuristics to guide a planner through this space of plans so that the system does not waste its time considering implausible plans. Plans are given weights based on heuristics, but these heuristics are all general purpose and apply to any set of plan operators. Simply knowing the frequency with which a dialog act operator occurs would help a planner give more accurate weights to its plans. Even more useful would be estimating the probability of a dialog act operator given the current context: previous dialog acts, other dialog acts of the current utterance, cue words and phrases, the sentence form (declarative, imperative, etc.), and prosodic cues (including change of turns and knowing whether the utterance to be tagged is an interruption).

Machine learning techniques have been used successfully in similar applications where there is a large amount of background information and it is unclear what the relevant pieces are. Connectionist models have been used in parsing (Wermter & Weber 1994), while inductive learning methods have been used in parsing (Magerman 1995), part of speech tagging, and speech repair correction (Heeman 1997).

To predict an utterance's DAMSL tags, one could simply use the highest frequency values for each label. Hopefully, one can do better by using machine learning techniques to identity contexts where alternative values are more likely. The syntactic form of the sentence will likely prove important as declarative sentences are usually *Asserts*; imperative sentences, commands; and so on. Common-sense notions about dialog such as:

- questions end in a rising pitch
- you answer questions
- you accept/reject commands and statements
- no often signals a reject

will prove important in predicting DAMSL tags. However, it will be up to machine learning techniques to find more complex patterns in the sentence forms, cue words, and DAMSL tags of dialogs. The context around each utterance could correspond to a 28 or more dimension space, a difficult puzzle to sort out by hand.

Preliminary Experiments

In the process of building a system to predict dialog acts, it is important to measure the frequency of these acts. These frequencies create a baseline for a tag predictor to beat; if the predictor is worse than this baseline then we are better off always guessing the most frequent tag. Table 1 shows the highest frequency tag values for the eight major DAMSL tag sets.

The DAMSL tag frequencies (discussed in the following subsection) and the decision tree testing data came from a corpus of 19 annotated TRAINS dialogs containing 1524 utterances. These 19 dialogs were annotated by pairs of University of Rochester (undergraduate and graduate) students; initially the students worked separately and then collaborated to produce a final annotation.

Nature of the Corpus

Using the probabilities of the tag values for the 8 major DAMSL tag sets, cross entropy and perplexity can be computed. In this case, we are using cross entropy to measure how random the distributions of a tag's values are (given no context). The formula used here for cross entropy is:

$$H = -(1/K)\sum_{i=1}^{N} c_i * log_2(p(v_i)) \tag{1}$$

where v_i indicates a tag value, K is the size of the corpus (1524), c_i is how often v_i appears in the corpus, and $p(v_i) = c_i/K$. Perplexity is simply 2^H; an intuitive view of perplexity is that it gives the average number of choices faced by a tag prediction model. In the case of a two valued tag such as Other Forward Function, the worst perplexity is two; the values are completely random and each is equally likely. The best perplexity will always be one; only one value is possible.

Table 2 shows the cross entropy and perplexity of the 8 major DAMSL tag sets. The results show that the high frequency tag values keep the perplexity of tag prediction low.

Initial Decision Trees

Our plan is to use decision trees to find exceptions to cases where high frequency tag values such as "Not an answer" apply. A preliminary test of this plan involved hand constructing decision trees that were better than always guessing the most frequent tag. Table 3 shows the accuracies (percent right/total) of these trees and figure 2 describes their structure. Notice Info-level, Influencing Addressee Future Action, and Other-forward-function are not listed in the table. Our current decision tree system only uses the last DAMSL tags as context. Thus, it is difficult to recognize questions and commands (Influencing Addressee Future Action).

Tag Set	Most Freq. Value	Prob
Other-ForF	No Other-ForF	0.926
Info-level	Task	0.772
Statement	Not Statment	0.541
Infl-Addressee	No Influence	0.726
Committing-Spk	No Commitment	0.762
Agreement	No Agreement	0.636
Understanding	No Understanding	0.684
Answer	Not an Answer	0.853

Table 1: Highest Frequency Tag Values and Their Probabilities

Tag Set	Cross Entropy	Perplexity	# Tags
Other-ForFunct	0.381	1.30	2
Info-level	0.957	1.94	5
Statement	1.4	2.64	4
Infl-Addressee	1.2	2.29	4
Committing-Spk	0.97	1.96	3
Agreement	1.25	2.37	7
Understand	1.19	2.29	6
Answer	0.6	1.52	2

Table 2: Entropy and Perplexity of Current Annotations

Other-forward-function and Info-level are both complicated by the frequency of utterances such as *okay* that serve as acknowledgments and signals of repairs and topic shifts. Sometimes, however, *okay* serves as an acceptance and has a different set of Other-forward-function and Info-level tags. If we gave the decision tree the ability to recognize cue words such as *okay*, it is unclear whether this would be an advantage as it would still have to decide whether it was an acceptance or not. However, in 5 (6 if we assume that more context will help us predict Influencing Addressee Future Action) out of 8 cases, it looks like decision trees will allow us to do better than always guessing the most frequent tag.

Tag Set	Accuracy
Statement	63.2
Committing-Spk	78.8
Agreement	70.0
Understand	70.1
Answer	93.8

Table 3: Accuracies of Hand Coded Decision Trees

Statement DT: **if** previous utterance is info-request-directive **then** label assert **else** label not statement

Committing Speaker Future Action DT: **if** previous utterance is action-directive **then** label commit **else** label no commitment

Agreement DT: **if** previous utterance is action-directive, open option, or check question (marked by other statement) **then** label accept **else** label no agreement

Understanding DT: **if** previous utterance is action-directive, open option, or check question (marked by other statement) **then** label acknowledgment **else** label no understanding feature

Answer DT: **if** previous utterance is info-request-directive **then** label answer **else** label not answer

Figure 2: Hand crafted decision trees

Conditional Probability	Value	Unigram
$P(Assert_i \mid \text{Task-management}_i)$	0.586	0.373
$P(Assert_i \mid Answer_i)$	0.955	0.373
$P(Assert_i \mid \text{Other-Statement}_{i-1})$	0.841	“
$P(\text{Info-request}_i \mid \text{Other-statement}_i)$	0.889	0.152
$P(Assert_i \mid \text{Info-request}_{i-1})$	0.791	0.373
$P(Assert_i \mid \text{Signal-non-understand}_{i-1})$	0.647	“
$P(\text{Info-request}_i \mid \text{Signal-non-understand}_i)$	0.882	0.152
$P(Hold_i \mid \text{Signal-non-understand}_i)$	0.706	0.033
$P(Assert_i \mid Hold_{i-1})$	0.740	0.373
$P(Assert_i \mid \text{Open-option}_i)$	0.806	0.373
$P(Offer_i \mid \text{Open-option}_i)$	1	0.052
$P(Assert_i \mid \text{Action-directive}_i)$	0.558	0.373
$P(Assert_i \mid Reject_i)$	0.971	“
$P(Assert_i \mid Competition_i)$	0.909	“
$P(Accept_i \mid \text{Action-directive}_{i-1})$	0.699	0.306
$P(Accept_i \mid \text{Other-statement}_{i-1})$	0.683	“
$P(Accept_i \mid Repeat_i)$	0.641	“
$P(Accept_i \mid Acknowledge_i)$	0.797	“

Table 4: Useful Conditional Probabilities

Corpus Analysis

The unigram probabilities of table 1 do not tell us much about our corpus of dialogs. The fact that 54.1% of utterances are not statements is not surprising since we would expect a mixture of commands and questions as well as statements.[2] To investigate more interesting probabilities, we measured conditional probabilities that indicate cases where a tag is more probable than the most likely tag (given no context). These probabilities are shown in table 4 with the third column showing the relevant unigram tag probability. The subscripts refer to utterance numbers; some of the conditional probabilities involve tags of the same utterance while others refer back to a previous utterance.

The first set of probabilities in the table involves task management. Task management utterances in DAMSL are defined as meta-level discussion about planning and experimental procedure: "let's work on the orange juice problem first", "Are we done?". 34 of the 58 task management utterances in the experiment were Asserts. Many of these utterances state that the task is completed: "we are done". The rest are various assertions about the problem solving process: "we need to do this part first", "I added wrong", "either solution is fine". Since in general Asserts only occur 37.3% of the time, this information is valuable and gives us insight into meta-level planning discussion.

The second set of probabilities in the table involves questions and answers. Check-questions are statements about the world as well as questions. An example is "there are oranges in Corning right". They are usually labeled Other-statement because they generally do not introduce new information but are not repeating anything either since they deal with assumptions. Other-statement is highly co-related to Info-requests, and Asserts are highly co-related to answers. Thus, the probability of an Assert is higher after an Other-statement. Interestingly, the conditional probability of an Assert following an Info-request is lower. Some answers are imperatives and are not marked as Asserts. Answers to check questions are more likely to be "okay" or "yes" and marked as Asserts.

Signaling non-understanding is commonly done through questions; answers are likely to follow so that the next utterance is likely an Assert. When a speaker makes a request or assertion, the listener is obligated to at least implicitly accept or reject the utterance. Sometimes the listener will ask for clarification or otherwise delay their judgment. These utterances are marked *Holds*. Since clarification is needed if a listener does not understand the previous utterance, Holds and Signaling non-understanding co-occur. Answers that are Asserts often follow Holds.

The third set of probabilities in the table concerns Open-options and Action-directives. *Open-options* are weak suggestions by the speaker. These are likely to be Asserts ("there are oranges at the Corning warehouse you could use"). The annotation manual states Open-options are usually Offers explaining why annotators have so far marked every Open-option as an Offer. *Action-directives* are requests by the speaker that the listener perform an action. In the TRAINS domain, 55.8% of the Action-directives are also Asserts. Since the TRAINS domain is about planning, these are utterances such as "the train will go through Corning" that serve as statements but are also requests that the listener accept the action as part of a plan. A common way to make a rejection is through an Assert, and completions of another speaker's utterance are often marked Asserts as well.

The last set of probabilities in the table involves the frequency of acceptances. In TRAINS, the two speakers have roughly the same expertise although one speaker has travel time information and the other does not. 70% of the responses to Action-directives were Accepts because the listener is not likely to immediately to see a problem in a speaker's request since they both have similar levels of expertise and knowledge. The same is true of check questions indicated by their Other-statement tag. Given that an utterance is an acknowledgment ("okay", "right") or an acknowledgment made through repetition, it is likely that the utterance is also an accept.

Using a sophisticated statistical tool such as a decision tree should bring out additional results based on looking at more complex patterns of DAMSL tags. The addition of cue words and sentence forms to the context should bring new developments as well. Another area for exploration is the use of data mining techniques to extract interesting patterns from the corpus. (Dehaspe & Raedt 1997) tries such an approach on part of speech tags in a corpus of Wall Street Journal text.

Conclusions

Results on hand coded decision trees (given only the context of the last utterance's DAMSL tags) suggest that automatically derived decision trees should be effective DAMSL tag predictors. The conditional probabilities taken from the labeled corpus not only provide interesting insights into these dialogs, but suggest that an n-gram model of dialog acts could also be an effective DAMSL tag predictor. (Jurafsky *et al.* 1997) uses such an n-gram dialog act model as a component of

[2]Commands and questions can have statement aspects to them but typical examples such as "take the Corning path" or "how far is that?" do not.

their 3 part system which also includes word models and a decision tree using prosodic features as context. Different n-gram word models are created for each dialog act and one the best fitting the input is chosen for each utterance. (Mast *et al.* 1996) compares using decision trees (having the utterance's words as context) to using word models like (Jurafsky *et al.* 1997) and finds that the word models are better. Thus as work proceeds on predicting DAMSL tags in TRAINS dialogs, it may turn out that word models and n-gram dialog act models are used.

Having the best possible estimates of the probability of each DAMSL tag being present in the current utterance, it would be interesting to feed these into a fully connected network containing nodes for all the DAMSL tags. The connections would be weighted according to the probabilities of tag co-occurrence. Node probabilities could be recomputed based on these link values and the probabilities of the nodes they were connected to. This should give a better estimate of tag probabilities and take advantage of the unique nature of DAMSL's layers of independent tags.

Acknowledgments

This work was supported in part by National Science Foundation grants IRI-95-03312 and IRI-95-28998, the latter of which was under a subcontract from Columbia University. The Rochester annotation effort is headed by James Allen with the help of Teresa Sikorski, George Ferguson, and myself. Thanks to James Allen and Lucian Galescu for their help with this paper.

References

Allen, J. F., and Perrault, C. R. 1980. Analyzing intention in utterances. *Artificial Intelligence* 15:143–178.

Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics* 22(2).

Cohen, P. R., and Perrault, C. R. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science* 3:177–212.

Core, M. G., and Allen, J. F. 1997. Annotating dialogs with the damsl annotation scheme. Presented at the AAAI Fall Symposium on Communicative Action in Humans and Machines. Available at http://www.cs.rochester.edu/u/mcore.

Dehaspe, L., and Raedt, L. D. 1997. Mining a natural language corpus for multi-relational association rules. In Daelemans, W.; van den Bosch, A.; and Weijters, A., eds., *Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks.* 35–48.

Ferguson, G.; Allen, J.; and Miller, B. 1996. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proc. of the 3rd International Conference on AI Planning Systems (AIPS-96).*

Heeman, P., and Allen, J. 1995. the TRAINS 93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester, Rochester, NY 14627-0226.

Heeman, P. A. 1997. *Speech Repairs, Intonational Boundaries and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialog.* Ph.D. Dissertation, Department of Computer Science, University of Rochester.

Jurafsky, D.; Bates, R.; Coccaro, N.; Martin, R.; Meteer, M.; Ries, K.; Shriberg, E.; Stolcke, A.; Taylor, P.; and Van Ess-Dykema, C. 1997. Automatic detection of discourse structure for speech recognition and understanding. In *Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding.*

Magerman, D. M. 1995. Statistical decision-tree models for parsing. In *Proc. of the 33rd annual meeting of the Association for Computational Linguistics (ACL-95).*

Marcus, M.; Santorini, B.; and Marcinkiewicz, M. 1992. *Building a large annotated corpus of English: the Penn Treebank.* Penn Treebank Project.

Mast, M.; Niemann, H.; Nöth, E.; and Schukat-Talamazzini, E. G. 1996. Automatic classification of dialog acts with semantic classification trees and polygrams. In Wermter, S.; Riloff, E.; and Scheler, G., eds., *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence. New York: Springer. 217–229.

Miller, G.; Leacock, C.; Tengi, R.; and Bunker, R. 1993. A semantic concordance. In Bates, M., ed., *Proc. of the ARPA Human Language Technology Workshop*, 303–308. San Francisco: Morgan Kaufmann Pub. Inc.

Reithinger, N., and Maier, E. 1995. Utilizing statistical dialogue act processing in verbmobil. In *Proc. of the 33rd annual meeting of the Association for Computational Linguistics (ACL-95).*

Searle, J. R. 1975. A taxonomy of illocutionary acts. In Gunderson, K., ed., *Language, Mind, and Knowledge. Minnesota Studies in the Philosophy of Science.* Minneapolis, Minnesota: University of Minnesota Press. 344–369.

Siegel, S., and Castellan, N. J. 1988. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill, second edition.

Wermter, S., and Weber, V. 1994. Learning fault-tolerant speech parsing with screen. In *Proc. of the National Conference on Artificial Intelligence (AAAI94).* MIT Press.