Internal Use Only

TR-I-0368

# Results of the ABDUCK Disambiguation System
# ABDUCK 会話理解システムの概要

John K. Myers

March 12, 1993

*Abstract*

This paper offers a final report of the research results achieved so far by the author in natural-language understanding for automatic interpreting telephone applications. The ABDUCK system is designed to accept ambiguous surface-semantic feature-structure candidates output from the parser, understand and disambiguate the utterance, and give as output directly to the transfer module feature-structure representations of the utterance at the surface-semantic, deep-semantic ("real meaning"), and illocutionary-force ("speech act") levels of representation. This results in a significantly more natural translation. For instance, the system can understand unagi-da sentences and the different meanings of "hai" and "wakarimashita". A previous paper has discussed the theory of the system. This report presents the results of the ABDUCK understanding/disambiguation system.

# Results of the ABDUCK Disambiguation System for Conversations A,B, and 1-10

John K. Myers

ATR Interpreting Telephony Research Laboratories
Sanpeidani, Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
myers@atr-la.atr.co.jp

## Abstract

This report offers a final report of the research results achieved so far by the author in natural language understanding for automatic interpreting telephone applications. The ABDUCK system is designed to accept ambiguous surface-semantic feature-structure candidates that are output from the parser, understand and disambiguate the utterance, and give as output directly to the transer module feature-structure representations of the utterance at the surface-semantic, deep-semantic ("real meaning"), and illocutionary-force ("speech-act") levels of representation. This results in a significantly more natural translation. For instance, the system acan understand unagi-da sentences and the different meanings of "hai" and "wakarimashita". A previous paper has discussed the theory of the system [Mye92a]. This report presents the the results of the ABDUCK understanding/disambiguation system.

# Contents

# 1 Introduction

This report presents the current version of the results of the ABDUCK understanding and disambiguation system. ABDUCK is a system that is designed to fit between the parser and the transfer module in the current interpreting telephone design. It can take ambiguous multiple feature-structure output directly from the parser, and give disambiguated feature-structure output directly to the transfer module.

The theory to the ABDUCK system is presented in the international symposium paper, "An Agent-Based Approach to Natural-Language Understanding of Conversations for an Interpreting Telephone". This theory will not be repeated here, and it is assumed that the reader has read this paper already.

Because the system is constantly changing and being updated, the results are also in a state of flux. This is version 3.0 of the system, after a major change in the representation system and the illocutionary force classifications. All 12 conversations are accounted for. The scores are probably not as high as version 1.8's results for A, B, and 1-5 presented in the ABDUCK paper, because another two tuning passes are needed to bring the system up to 3.2 before this representation becomes more stable.

# 2 Discussion of Contents

The results are presented in a number of sections.

First, the Master Training Data is presented. This shows the set of "illocutionary force" speech acts used by the system, along with the sets of deep semantic patterns, surface semantic patterns, and inference rules. This data is important because it represents the ontology of the system.

Next, the conversations used for training are presented. These use the macros presented in the Master Training Data. This data is important because it represents a classification of all 12 conversations into speech-act types and deep semantic meanings. Many comments are put in the middle.

After this, the analysis of the output is presented for the closed training set. This analysis is automatically compiled by the system. Only the parts that the system gets wrong are presented. This automatic analysis aids significantly in maintaining the system.

Finally, the full output for the closed training set is presented, in Section 17. It would probably be useful to glance at this first, to get an idea of what the system does. For each utterance, multiple surface-semantic utterance candidates are read in by the system; the system then attempts to disambiguate these by choosing the correct surface-semantic candidate, and understand these by in addition outputting the deep-semantic and the illocutionary-force ("speech act") meanings for that utterance.

# 3 The Philosophy of ABDUCK

The ABDUCK system was designed to improve the state-of-the-art and to address concrete problems in machine translation. Instead of trying to take information out of the system, the conscious design is to add as much information into the system as possible. The current ABDUCK system uses information at the surface semantic, deep semantic, and speech-act ("illocutionary force") levels. Designs for future extensions include adding information at the agent-simulation, plan-recognition, script-understanding, and dialogue-understanding levels.

It is important to use as much information as possible for understanding and disambiguation. Computer hardware designs become 10 times as fast every three to five years. In less than 15 years our computers will be at least 1000 times as fast as they are now. Current efforts to make translation systems run faster by taking knowledge out of the system are misguided. Instead, we have to ask, "What kinds of powerful algorithms will we run on a computer that is 1000 times as fast and 1000 times as large in the near future?"

# 4 Use of the Program

## 4.1 Bringing Up the Program

The entire system is brought up by loading file `LM01:>myers>abduck>aaa-dis-load-system`. This loads in the entire system and all associated files, and all data and training files. It then trains the system. After this file has finished loading, the system can be run.

## 4.2 Run Modes

There are currently three modes in which to run the system: normal mode, demo mode, and filedump mode.

Normal mode is invoked by calling function `(reset-normal)`. In normal mode, the output goes to stream `OS`, which is normally bound to T, i.e. the user's terminal input/output window. However, with function `(use-file my-filename)`, the stream can be redirected to write out into a file. A summary is printed of the mistakes at the end.

Demo mode is invoked by calling function `(reset-demo)`. In demo mode, the output goes to a special demonstration window which is invoked on the Lisp Machine by pressing the two-key sequence `SELECT A`. In demo mode, the left column shows the ambiguous inputs to the system; the right column shows the various expected predictions made by the system; and the middle column shows the resulting understood illocutionary force, deep semantics, surface semantics, and output documentation string. An interactive Lisp Listener is provided in the top center to drive the demonstration. For instance, the user can type `(ABDUCK 3)` here and the system will process the third conversation. See Figure 1.

Filedump mode creates and sends output to a file in a special format that is compatible with the input to Mark Seligman's transfer program.

## 4.3 A Brief Map of the System

The system consists of: (1) The *code* for the program itself; (2) A special memory space in the code for *trained trajectories*, which is used by the trajectory-based predictor (the current main part of ABDUCK) to predict possible inputs; (3) A set of large *arrays* in memory space used to passively store conversations, training conversations, etc. All of these must be read in from disk and loaded into memory before they can be worked with. The system requires that the training conversations be *trained* by running routine `(HARD-RESET-MEMORY)`; it is insufficient to only load the data into memory. This routine is called automatically by the system setup routines; (4) A set of *Master Macros* that define illocutionary force ("speech act") types, deep semantic pattern types, and surface semantic pattern types. These *must* be read in before the following: (5) A set of *training conversations* that give the ideal interpretation of each utterance, in illocutionary-force, deep-, and surface-semantic spaces. These are defined using the previous Master Macros. It is possible to have multiple suggested surface-semantic patterns per utterance; however, the programmer must guarantee that the first one in the set is the correct one. (6) A set of *input conversations* that will be processed, referred to by their code numbers. Each input utterance may have more than one possible candidate for disambiguation; however, the programmer must guarantee that the first candidate in the set is the correct one. This information is used by the automatic scoring system, among other things.

## 4.4 Running the Program

There are two other major modes for running the program, "demonstration mode" (closed-set training), or "examination mode" (open-set training). These are determined by setting the system flag `*Use-Jackknife*` to NIL or T.

The system can be run using the `(ABDUCK 1)` through 10, or 'A, 'B, commands. This runs one conversation.

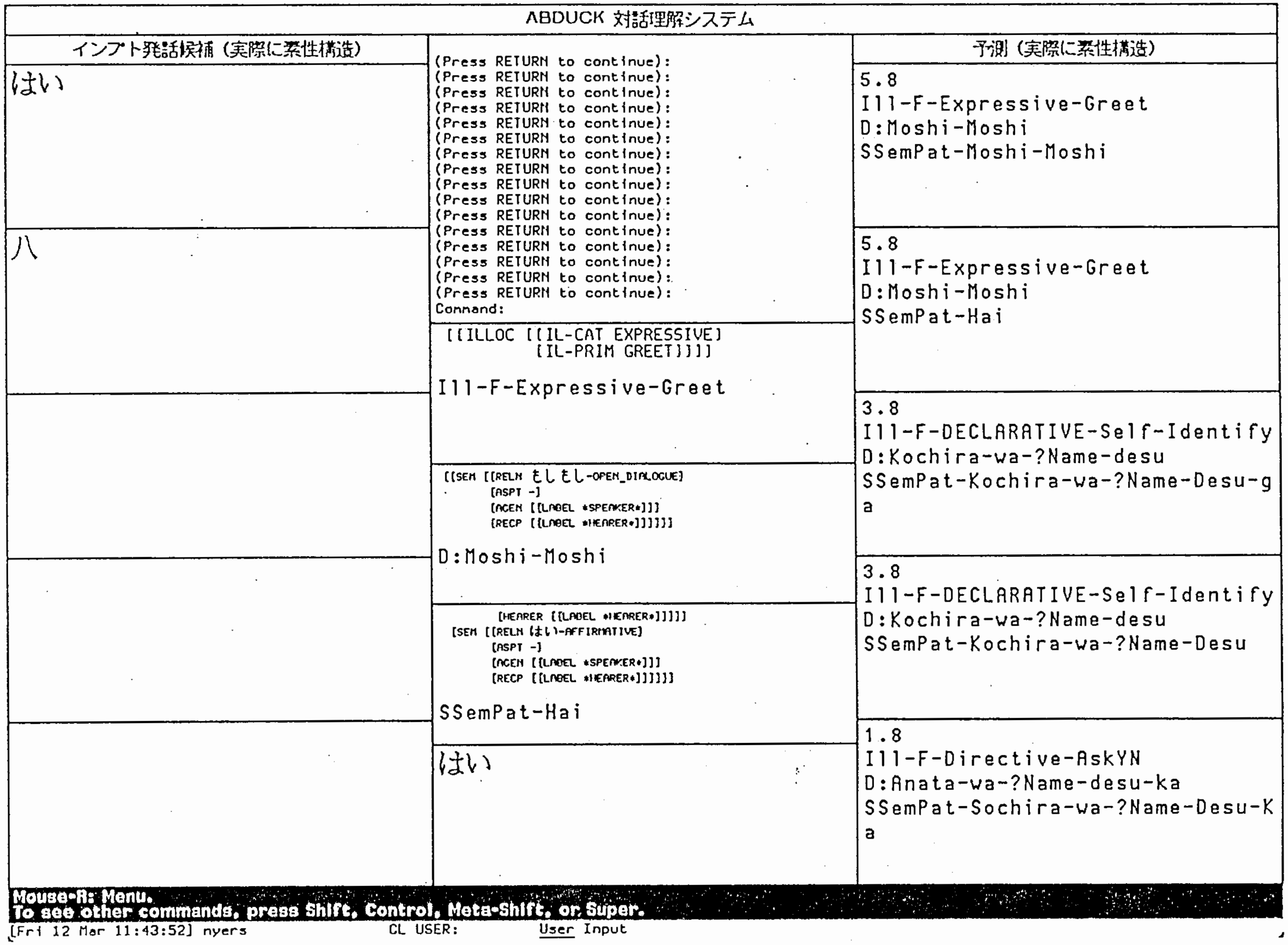


Figure 1: The Demo Mode Interaction Screen

The system can be tested on all conversations in demonstration mode by using the function (ABDUCK-TEST), and in examination mode by (ABDUCK-JTEST). These functions automatically create files of the short and long versions of the output under directory LM01:>myers>ABDUCK>*. Both of these functions are called by (ABDUCK-FULL-TEST).

The system can also be run using the command (ABDUCK-FILEDUMP). This automatically switches the system into filedump mode, creates a single file LM01:>myers>ABDUCK>filedump.txt, and sends the output of all twelve processed conversations to this file.

Finally, (ABDUCK-QUICK-TEST) runs all twelve conversations without opening any files and without changing any modes.

All of these run commands can be found in the file LM01:>myers>ABDUCK>dis-prog.lisp .

## 4.5 Notes on the Program

The ABDUCK system is currently contained in directory LM01:>myers>ABDUCK>*. This contains all of the source to the ABDUCK system proper. The ABDUCK system also uses certain features of the old Hasegawa RWS system, in particular the definition of a feature structure, the utility for reading feature structures, and parts of the functions for matching feature structures. There is also a facility for normalizing feature structures into a standard form. The function for printing out feature structures has been taken over by the ABDUCK function my-pprint, which intelligently prints out feature structures, lists of feature structures, lists of other structures mixed with feature structures, recursive lists, etc. The main RWS features used by ABDUCK are kept in files Declare, Basic, Matcher, and Read-and-Print in the directory LM01:>myers>ENGINE>*. The system is also currently loading the entire RWS system from directory ln:/usr7/myers/engine/* (it is also kept under directory LM01:>myers>ENGINE>* and ATR-SQ:/usr2/myers/engine/*); this is probably not necessary and will be trimmed out in the future. The system also requires the Lisp extension routines found in LM01:>myers>system.

The system uses input data contained in LM01:>myers>convn>rough-1 , -10, -A, -B for conversations to process.

The system also uses training data contained in master definition files LM01:>myers>convn>real-rough-illoc-macros, LM01:>myers>convn>real-rough-deep-macros, and LM01:>myers>convn>real-rough-surf-macros. These are the romaji documentation versions. Kanji documentation versions are found in LM01:>myers>convn>deep-macros and LM01:>myers>convn>surf-macros. Watch to make sure that the versions are consistent between these.

The system also uses conversation training data contained in files LM01:>myers>convn>real-rough-1 , -10, -A, -B. These must be loaded after the master definition macro files have been loaded. They are used to train the trajectory prediction part of the system.

Loading the files does not perform training; the files are simply loaded into data structures in memory. Training is done by function (HARD-RESET-MEMORY). If the system is running in "demonstration" closed-set training mode, it is only necessary to train the system once at the beginning of the session, and then process as many conversations as is required. If the system is running in "examination" open-set training mode, it is necessary to retrain the system with all of the training files except the file for that conversation, before processing each conversation. This is done by calling (HARD-RESET-MEMORY 1) through 10, or (HARD-RESET-MEMORY A-code) or B-code. A-code is currently set to 0, B-code to 11. This function is called automatically by the system high-level (ABDUCK 1), 10, 'A, 'B, function, if the system flag *Use-Jackknife* has been set to non-nil.

The system also uses file LM01:>myers>convn>real-unagi-da-rules to load the rules used by the inference engine for predicting special types of utterances.

If many data files have been changed, it is necessary to reload the data using command (reload-conversations). Individual data files may also be reloaded by hand.

If any of the master macro files have been changed, it is necessary to reload and retrain *all* of the training conversations. This is accomplished by calling function (reload-training-data).

If the system gets modified, and in particular if the definitions-and-initializations file gets modified and recompiled, it is necessary to reload the entire system. This is most easily accomplished by calling function (`dis-compile-system`), which compiles and reloads the entire system, also reloading the conversation and training data.

Macro file versions are contained in global variables `*Illoc-Macros-Version*`, `*Deep-Macros-Version*`, `*Surf-Macros-Version*`, and `*Unagi-Da-Rules-Version*`. These are modified when the files are loaded. It is possible to keep an eye on the differences between the Kanji versions and the Romaji versions using this feature.

# 5 Why is the "Unagi-Da" Inference Engine Necessary?

The current implementation of the system is based on rewriting and not unification. Using the normal system, it is possible to build recognition patterns at training time that are used to construct rewriting rules, which can recognize and reason with fixed patterns at run-time. However, there are certain kinds of utterances, such as "unagi-da" interactions or short answer interactions, that contain *constants* that must be specifed at *run-time*. Although a unification system is powerful enough to handle this, a normal rewriting system with a fixed set of rules cannot. It is therefore necessary to have a special facility that dynamically creates new prediction patterns containing special constants at run-time. This is the "unagi-da" inference engine.

The use of the engine is transparent to the user. Currently the inference engine is implemented so that the rules each only take one antecedent, the current utterance, and produce only one consequent, an expectation of the next utterance at all (three) levels (with bound variables instantiated from the antecedent). Of course, the expectation is entered into the expectation heap for the next utterance. Thus, the current implementation basically consists of a training rule that is two utterances long, rather than one that is only one utterance long.

# 6 Problems and Things Learned from the Research

Up until now, interpretation of conversational utterances has concentrated mostly on the translation of vocabulary and syntactical/semantical grammar. However, natural translation requires that the *pragmatics* of utterances be translated in a skillful fashion as well–including such things as "speech-act type" and "illocutionary force". Instead of crisp black and white definitions, these areas are still grey, poorly defined, and poorly understood. Much further research is required before the theory of pragmatics can be well-defined.

A first step towards building such a theory is to gather relevant information. This can then be used to form classification schemes, or refute older systems as inadequate.

The next few sections present and discuss information that was learned while building the third version of ABDUCK, a system for Agent-Based Dialogue Understanding of utterance candidates using Commonsense Knowledge. The information comes mostly from observing where systems based on previous technology would break down, and inferring what is necessary to fix in order to make the system work correctly. It can be argued that this kind of rough but practical information is the most useful result of performing research of this type.

## 6.1 Reverse Dictionary

The system needs a Reverse Dictionary, for the implementor to use when extending the system. This would take a Surface, Deep, or Illoc pattern form as input, and provide a list of all the Illoc-Deep-Surface utterance training forms that match it/use it. It would be best to implement this using hash-tables, and not try to do the actual matching. This would probably require the training conversations to explicitly take the macros as input, and then take another pass through to install the macro FSs, instead of implicitly taking the macros as input but automatically using their FSs. More research is required in the practical matter of managing the system.

# 7 Problems with the Program: Inference Rules

There is a latent bug in the Hasegawa feature structure instantiator that takes ?rest variables and quietly deletes them in the rewritten instantiation if they are unbound. This means that currently the inference engine cannot support rules with ?rest variables in the consequent. Unfortunately, this representation method is almost required to be able to represent anything of serious complexity. Thus, it is very difficult to write good inference rules for the current version. This feature must get repaired soon in a later version.

The current version only supports predictions from one utterance to the next. This means that questions and answers which are based on three or four predictable utterances, instead of just two, cannot be supported yet with this system.[1] However, in the current conversations, the answers, "Hai, sou desu", and "Iie, mada desu", are each broken up into two utterances, e.g. "Hai." "Sou desu.". This problem is properly attacked by making trajectories more powerful, and able to remember context from previous utterances and apply it using logical inferences. A related problem, that of trajectories at the application level, may be attacked by building a predictor based on application-level script fragments.

## 7.1 Actor: Agent vs. Experiencer

There is a problem in that in some grammatical constructs, the actor is represented as the AGEN (agent) of the utterance. In other types of grammatical constructs, the actor is represented as the EXPR (experiencer) of the utterance. Because these are different representations, it is difficult to bind a section of a feature structure that contains one representation to a variable, and then instantiate it in a different feature-structure pattern. This makes it difficult to write rules for the understanding system. It would be useful to standardize the semantic representation.

## 7.2 Problem on the representation of Zero Pronouns

In one question, the office asks, Will [] attend?, with a zero pronoun. In this case, the pronoun resolves to You. However, in another, the guest asks, Will others attend? and there is no need to resolve the Zero Pronoun. The current system needs to make two rules for these separate cases; however, if a variable is included for the subject slot, the variable will match the zero-subject case as well as the subject-present case. This could be solved by making a representation with negative variables (match everything *except* null) or typed variables. However, it is in general bad to make two rules for these two separate cases; it would be nice to have rules that are powerful enough to handle disjunctions.

## 7.3 Representation problem: Deep Trees vs. Flat Frames

One of the big problems with the current utterance representation system is the representation of parses in deep trees, instead of in flat frames. Because of this, sentences that are essentially identical must be represented in completely different structures. This means that understanding rules must be unnecessarily duplicated for each possible type of structure, which causes much work and makes the system brittle. With the current representation method, only the general types of sentences that are known beforehand by the programmer can be recognized; anything else gets dropped as a non-match. For instance, "Kaigi ni moshikomitai desu" and "Kaigi ni moshikomitai desu ga" require two completely separate recognition rules, because the first sentence is represented as a tree based on "desu",[2] whereas the second sentence is recognized as a tree based on "ga", with the "desu" tree being a subargument. If other grammatical parse representations are added to the parser, such as "Kaigi ni moshikomitai no de", or a construct using a "deshou" instead of a "desu", a completely new parse tree is created with a new top level that is unrecognizable by the understanding rules and the understanding system will break.

As another example, A no B is currently represented as a NO, with arguments A and B, instead of being represented as B, modified by an A, with a No attached. This makes type-checking very difficult,

[1] The previous NP system could support these, using a different technology. See a companion paper on "Short Answer" theory.

[2] Actually, the "desu"/"da"/"de gozaimasu" is eliminated and represented by a "tai-desire" in this case in the actual parser.

as even if a fuzzy type checker (for instance, "Place Names") could be built, there would always have to be two rules: one for the noun itself, and one for the noun as part of a "NO" form. Adjectival phrases are not represented this way, we don't have RED which has an argument of a BARN, it's always BARN which is modified by being RED. Nouns modifying nouns should work this way too.

There is a problem in utterance (8-15) with representing the deep meaning of the utterances of people who give their company names. The company name should not go inside the name: obviously "My name is 'John from Bell Labs'" is a bad semantic representation. What is needed is a semantic relationship such as "works-at" or even "from". But there is still a problem as to where to hang this on the feature structure. It looks like a semantic representation is needed similar to, "My name is 'John', and I'm from Bell Labs". A good feature-structure pattern would be able to represent optional information like this, instead of requiring two separate patterns (suppose he only says "My name is 'John'"?).

A better way to represent the feature structures is to hang auxiliary features, such as mood, possibility, and sentence modifiers, *underneath* the main verb instead of *on top of* it. For instance, in these examples, the "ga" politeness/uncertainty marker, the "no de" explanation marker, the "deshou" possibility marker, etc., can all be put in a frame slot underneath the main verb. This would allow the tree-pattern-based recognition system to understand the main parts of the sentence, even if new modifiers have been added to the parser.

## 7.4 Significant information being dropped by the parser

Some significant information is simply dropped by the parser and not represented in the output parse trees. In particular, if the sentence uses the particle "mo" instead of "wa", there is no marker for this. Also, if the sentences ends with "no desu ga" or "n desu ga", the "no" or the "n" is dropped, and the representation is the same as if the sentence were simply "desu ga". This information again should be put in a frame slot in a feature underneath the main information, as discussed in Section 7.3.

## 7.5 Recognizing a Commitment

Recognizing when a commitment speech-act is occurring is a difficult but important problem. It is important to first look at what a commitment is. According to some people, a commitment exists when a speaker says that he or she will do something, and it is important to the hearer, and the hearer relies upon the speaker to perform the action. There is a problem in speech-act recognition with the system thinking that the speaker is uttering a commitment every time that the speaker says that he will do something. This is too strong. If additional constraints are put on the recognition, such that the system recognizes that the action is for the hearer's benefit, then perhaps a commitment can be recognized better. This would probably require some sort of rule-based inference system in order to be effective.

However, commitments are slightly more complicated than this. For instance, in one case the speaker makes a commitment by saying that "I will come to the meeting place at 10:30". This is not for the benefit of the hearer, who must go to a lot of trouble to meet the speaker there at that time. However, if the speaker were *not* to show up, then that would seriously inconvenience the hearer. This is tied into the concept of relying on what the speaker said. It appears that relying on something, rather than being a question of benefit, is a question of not getting a negative benefit. This appears difficult to represent and reason with.

## 7.6 Suggested Extensions

### 7.6.1 Negative variables needed

One construct that would be useful in the understanding system's representation system would be the implementation of "negative variables". A negative variable is a variable that could match anything *except* a (set of) specified constant(s). For instance, it would be useful to have a negative variable that would match anything except "desu", or anything except "desu" or "aru". There are no designs for implementing such a variable at this time.

#### 7.6.2 Hierarchical Set Typed Variables needed

It would be extremely useful to be able to specify a typed variable instead of a general variable. This would also allow disjunctions. For instance, if there is a natural set "Hai" and "Iie", a typed variable *Hai-or-Iie* could recognize either of these but nothing else. This would allow multiple rules to be combined together into one rule, and create a seriously more powerful system.

#### 7.6.3 Space-saving flag

It would probably be convenient to somehow make a flag for deep semantic patterns that are exactly the same as the surface semantic patterns.

#### 7.6.4 Maintenance Test Tool needed

It is difficult to maintain surface syntactic and deep syntactic patterns, because there are too many of them. A tool should be built that tests all existing syntactic patterns against a new utterance to see whether a new syntactic pattern is required or not. An advanced version would suggest patterns that almost match closely but fail.

## 8 Interesting Speech-Act Sequences

```
Request? (I'd like you to send me a form so I can fill it out.)
Hai, wakarimashita.
```

In this case, the "Hai" is an acknowledgement (O.K.,), which also means assent a little bit, and the "wakarimashita" is a commitment, a volitional expression. I've translated it as "O.K.. I'll do that.". Note that this is different from the other wakarimashitas.

```
Please tell me about the conference      (Info-Request)
The attendance fee is 100 en      (Informing Act)
If you want to make a presentation, submit a summary (Instructing Act!)
```

The office volunteers instructions here, even though not asked, because the office thinks that the caller will probably need them. I'll send the Announcement, so please have a look at it.

```
I have applied for the Conference.    (Stage-setting Informing Act.)
I'd like to cancel my attendance.     (Statement of Want-to-Do.)
Can I get your name?
```

Normally the office would Acknowledge the Want-to-Do statement, and maybe say something like "I can help you" or "I'll try to help you", before asking the name. But in this sequence, the office skips all that and jumps directly to the question.

```
We can't make refunds after September. (Informing Act.)
We'll send you the form later.         (Promise.)
```

At this point, it would be expected that the hearer should respond with something like, "Thank you", to acknowledge the Promise of the office. However, the caller has a lot on his mind, and chooses to ignore the promise at this point (actually continuing a previous dialogue segment): Well, then, can someone attend the conference instead of me?

# 9 The Conversational Process

## 9.1 Acknowledgements

I heard what you said, and I understand it. Necessary when an Active Offer is made. Useful when a Promise or Commitment is made. Includes Thanking.

## 9.2 Meta-Communication: Asking for Permission to Converse

These possibilities were generated using the theory presented in [Mye92b].

```
* I'd like to talk with you about X.
* I wanted to ask you some questions about X.
* Is it O.K. if I   talk with you about X?
                  discuss X with you?
                  speak with/to you about X?
                  ask [you [about [X]]]?
                  ask you [some questions/things] about X?
* I need to /I have to/ I must
* I feel the need to//feel like I need/have to
* I should/ think that I should/feel that I should/ought to
* I would like to
* I wanted to/want to
* I thought/felt it would be {nice/fun/a good thing/important/useful/interesting/cool} to
* I was interested in
* I was thinking about
* I was planning to
* I was deciding whether to
* I was looking at
* I was thinking about the possibility/chance of
* I was wondering if it is O.K. to
* I was thinking about what would happen if I
* I was trying to schedule/think when I should
* I was trying to decide how I should
* I feel uncertain/anxious/Y about
* I was going to
* I had always wanted to
* I had forgotten to
* I was waiting to
* I thought/had decided that sometime I should

* I'm doing Y and //wanted to//
* I heard/read/was told   that it would be a good thing if I
* Y asked/requested/told/ordered/directed/instructed //would like/wanted/etc.// me to


There's something I'd like to
I have something I'd {like} to
     ...if you don't mind.
     ...if it's alright/O.K./not inconvenient.
```

## 9.3 Possible Response to Asking Permission to Discuss

```
* Yes?
* What is it?
```

* Go ahead
* Uh huh?
* Yes, please go ahead./ Yes, how can I help you?
* No problem.
* I'm listening
* Shoot.
* Sure. What would you like to talk about? (if subject not brought up).

* Yeah?
* And?

* So?

* I'm the wrong person to talk with. You need to see X.
* I'm busy right now/ I don't want to talk with you

* So what.
* Who cares.

## 9.4 Inviting: Offering Permission to Converse

How can I help you?
What would you like to talk [to me] about?
Can/May I help you?

Is there anything else [that you'd like to talk about/mention/go over]?
No, there isn't.
Yes, there is. [I wanted to talk about] XXX.
Note that this is also an answer to the previous meta-communication question.

### 9.4.1 Patterns for Giving Speeches or Instructions

* I'd like to say a few things about X.
* I'm going to say three things about X.
* I'm going to give you an introduction, and talk about X, Y, and Z.

# 10 Speech Act Types

Stage-setting. Explaining a situation, to lead up to something else.

Statement of a problem.

Ask for permission to converse
Give permission to converse (inviting)
Stage-setting information

Different from:
Asking an information question (Askq)
Receiving an answer (Inform)
Although "how can I help you?", "May I help you?", "What's the problem?" and "What is it?" are all Questions, the reply is not a literal answer.
"Can I help you?" "I certainly hope so."

Ask for information

```
*Deny the question (You don't need to ask that./You already know the answer.)
*Deny the ability to answer (I can't answer that because I'm:
  not qualified to/not allowed to /don't know the answer/ I don't have the
  answer for that right now/I don't have time to answer

Ask for favor
Say, I can't do that ==> is halfway an apology, or requires apologizing in addition.
     Or, may be a hard rejection, if not interested in being polite.

Call attention to, focus on, topicalize.
  Note that this can be information which is already known to the hearer.
```

The difference between a Direct-ion(2)(order) and a Request seems to be the power situation. If the speaker has power over the hearer in the subject matter of the content, then it is a Direct-ion(2); otherwise, it is a Request. These differ in that Requests can be ignored or rejected while only risking being thought impolite or unfeeling, whereas ignoring or rejecting a Direct-ion(2) calls into question the power relationship, and the director will have to respond to this or risk losing power. Direct-ion(2) entails that the director will be upset or inconvenienced if the direction is not taken.

A Direction(1) differs from a Direct-ion(2) in that a Direction(1) is: If you want to have this happen, then you should take these steps. It is part of an Informing action and does not require that the speaker desire that the actions be taken.

A third type halfway inbetween (1) and (2) is an informing act of a sequence of actions that the listener should take if he or she wants to, and the speaker prefers that the listener take these actions, but will not become upset if the listener does not take the actions.

Possible sequences are thus:

```
Request/Direct
Accept

Request/Direct
Reject

Request/Direct
  Same person provides more required information.
  Other person eventually responds.
```

# 11 Example Input Data

Conversations A, B, and 1-5 are ambiguous and have multiple possibilities per utterance. Conversations 6-10 do not have ambiguous parsing results available for input yet, and so are processed only for understanding purposes, not disambiguation.

## 11.1 Example Input from Conversation 5

```
(START-CONV-#  5  )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CONV5;;;;;;;;;;;;;;;;;;;;;;;
;--------------------------------------------------------------------------------------
;     はい                                                    (p =   0.7656723)
;     二人                                                    (p =   0.0119635)
;
;

;;      recognized                                   =  はい
;;      probability                                  =  0.765672251
;;      score                                        = -0.434327749

(PROBABLE-ALTERNATIVES                               ; Utt #1


(Probable-Utt 0.765     "はい"

[[SEM [[RELN  はい-AFFIRMATIVE]
       [AGEN !X03[[LABEL  *SPEAKER*]]]
       [RECP !X04[[LABEL  *HEARER*]]]
       [ASPT  -]]]
 [PRAG [[SPEAKER !X03]
        [HEARER !X04]]]]

)


(Probable-Utt 0.017     "二人"

[[SEM [[RELN 人]
       [RESTR [[RELN NUMBER]
       [COUNT  二]]]]]]

)


)



;--------------------------------------------------------------------------------------
;     こちら会議事務局でございます                          (p =   0.0000026)
;  --> こちらは会議事務局でございます
;
;
(PROBABLE-ALTERNATIVES                               ; Utt #2


(Probable-Utt 0.765     "こちらは会議事務局でございます"

[[SEM !X10[[RELN   だ-IDENTICAL]
           [OBJE !X07[[LABEL  *SPEAKER*]]]
           [IDEN [[PARM !X06[]]
...

;;;;;;;;;;;;;;
(END-CONV)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CONV5;;;;;;;;;;;;;;;;;;;;;;;
```

# 12 Master Training Data

## 12.1 Illocutionary Force ("Speech Act") Training Data

This is kept in file LM01:>myers>convn>real-rough-illoc-macros. In the interests of space, only the most relevant information from this file is presented here; interested people may print the file out.

```
  MACRO NAMES:
   Ill-F-Expressive-Greet
   Ill-F-Expressive-Acknowledge      ;Includes Phatic Sou-desu-ka.
                     Also probably phatic Ii desu yo!(Great!),and Matte-kudasai.
   Ill-F-Expressive-Thank                 ;Includes Yoroshiku-onegaishimasu.
   Ill-F-Expressive-Ack-Thank
   Ill-F-Expressive-Say-GoodBye
   Ill-F-Declarative-Self-Identify
   Ill-F-Declarative-Confirm   ;Now use YNReply and Short-Answer-Reply.
                          Only for information repetition replies.
   Ill-F-Commissive-Promise                 ;I [will] send you the form.
   Ill-F-Commissive-Commit   ;Normally should use Promise.
   Ill-F-Commissive-Passive-Offer
             ;If you have any questions, please ask, and I'll take care of it.
   Ill-F-Commissive-Active-Offer
      ;I can send you the form. [Would you like that? {Deep: Off; Surf: YN Info}]
   Ill-F-Commissive-Permission-to-Discuss-Offer
                          How can I help you? Can I help you? What is it? etc.
   Ill-F-Commissive-Accept
   Ill-F-Commissive-Reject
   Ill-F-Assertive-Inform
         ;Stage-setting. Replies to info-reqs. Elaborations/explanations of YN
            replies, and instructions.
   Ill-F-Assertive-Want-to-Ask-About
              ;  Includes Want-To-Talk, Onegai ga aru.
   Ill-F-Assertive-Want-to-Do
   Ill-F-Assertive-YNReply
                       ;includes replies to Request-Confirm ne questions.
   Ill-F-Assertive-Short-Answer-Reply        ;includes Mada
                       desu and Sou desu, also Dekimasen, besides Arimasen.
   Ill-F-Assertive-WHReply          ;Only for WH questions.
   Ill-F-Assertive-Permission-Reply
                    ;Includes granting and denying.  Includes "No problem".
   Ill-F-Directive-AskYN
                       ;Ask-identity YN is a special case of this.
   Ill-F-Directive-AskNY
                          ;Negative YN question.  Requires explanation.
   Ill-F-Directive-AskWH             ;Does NOT include HOW questions.
   Ill-F-Directive-Request                 ;includes Can you please send ...
            The request is from the speaker for the hearer to please execute an action.
            A different type of request is when the speaker selects a choice,
               perhaps that has been offered.
              The current system does not yet differentiate between these,
              but it's needed to disambiguate wakarimashita
               I'll Do that (alright) and Alright (sounds good).
   Ill-F-Directive-Info-Request                 ;includes Please tell me about...
   Ill-F-Directive-Request-Instructions
                           ;includes HOW should I..., WITH WHAT should I...,
   Ill-F-Directive-Request-Permission
   Ill-F-Directive-Instruct
   Ill-F-Directive-Direct           ;This will normally be
                                         Instruct, for our conversations.
   Ill-F-Directive-Request-Confirm    "XYZ, right?" -ne questions.


;;NOT YET IMPLEMENTED, FOR FURTHER RESEARCH:
  ;directive: Permit [Permit should be a declarative...]
Commissive: Offer  (Can utterance)  I Can do this.
  "May I help you?"  Offer; invitation to define problem.

Dialogue:  Backgrounding.  Stage-setting.

"The titles of the papers which will be presented at
```

```
  the Conference are listed in the second version of the Announcement."
indirect Directive: Direct (If you want to know what to do, then look there.)

Commissive: Accept/Consent
  Accept is doing something yourself.
  Consent is allowing someone else to do something.

Commissive: Promise
(because hearer receives benefit from the action.  Otherwise a Commitment.)

 "I think I would like to do this."  At start of trajectory is
                                     ;indirect request for help/directions.
                                     ; In middle is informing act of a choice.
Assertive: Inform
indirect request for help


Please wait a minute...  I am going to stop talking.

 I Will Check to see whether we have one or not.
          Now I am starting to do X.  Commitment???
Assertive: Inform
Commissive: Commit    (This is a special AM GOING TO DO RIGHT NOW assertion.)



Macros are of the format:

(ABDUCK-FS-MACRO
         Ill-F-Directive-Info-Request          ;includes Please tell me about...
         "Ill-F-Directive-Info-Request"

        [[ILLOC  [[IL-CAT     Directive]
                  [IL-PRIM   Info-Request]]]]
)
```

## 12.2 Deep Semantics Training Data

Relevant information can be found in the file LM01:>myers>convn>real-rough-deep-macros. The deep semantics information used by the system is summarized here:

```
 MACRO NAMES:
  DeepSemPat-Moshi-Moshi
  DeepSemPat-Arigatou
  DeepSemPat-Dou-Itashi-Mashite
  DeepSemPat-Yoroshiku-onegaishimasu-Thanking
  DeepSemPat-/?Connective-Rest/-?Object-o-Yoroshiku-Onegaishimasu-Request
  DeepSemPat-Iie-No
  DeepSemPat-Iie-You-re-Welcome
  DeepSemPat-Ii-desu-yo-Great
  DeepSemPat-Nan-deshou-ka?
  DeepSemPat-Mada-Desu
  DeepSemPat-/Chotto/-Matte-kudasai
  DeepSemPat-/?Connective-Rest/-Matte-imasu
  DeepSemPat-/?Connective-Rest/-Shitsurei-Shimasu
                                         ;Now bound to Sayounara.
  DeepSemPat-/?Connective-Rest/-Sayounara
  DeepSemPat-Hai-Well-Lets-See
  DeepSemPat-Hai-OK     ;(Acknowledge, not permission)
                                ;After request (also >indirect Can request).
  DeepSemPat-Hai-Yes    ;After -ne ?
  DeepSemPat-Hai-Right
  DeepSemPat-Wakarimashita-Great
  DeepSemPat-Wakarimashita-Alright        ;after Request (choice)? Wakarimashita.
  DeepSemPat-Wakarimashita-I-ll-do-that   ;after Request? Hai, wakarimashita.
  DeepSemPat-Wakarimashita-OK                ;After indirect AskWH question.
  DeepSemPat-Wakarimashita-I-See    ;after WHReply-informing statement
  DeepSemPat-Wakarimashita-OK-Got-It
  DeepSemPat-Sou-Desu-Confirmation
```

```
DeepSemPat-Sou-Desu-ka-I-See      ;After assertive-inform.
DeepSemPat-Arimasen
DeepSemPat-Dono-youna-?Verb-desu-ka?             ;See the next two.
DeepSemPat-Dono-youna-?Noun-de-?Verb-Phrase-eba-ii-desu-ka?
                                              ;Where did the "de" go to in the parse?
DeepSemPat-Dono-youni-?Verb-yoi-desu-ka?         ;See previous and next.
DeepSemPat-Dou-?Verb-eba-yoroshii-desu-ka?        ;See previous two.
DeepSemPat-/?Connective-Rest/-Anata-no-gojuusho-wa-nan-desu-ka?
DeepSemPat-/?Connective-Rest/-Anata-no-denwa-bango-wa-nan-desu-ka?
DeepSemPat-Anata-no-gojuusho-wa-nan-desu-ka-to-anata-no-onamae-wa-nan-desu-ka?
DeepSemPat-Ninzuu-wa-nan-desu-ka-to-anata-no-onamae-wa-nan-desu-ka?
DeepSemPat-Anata-wa-?Name-desu-ka
DeepSemPat-Watashi-wa-/?Place-Or-Object-Rest/-ni-?Apply-Attend-Verb-tai-desu
DeepSemPat-Anata-wa-/?time-rest/-?Object-o-motte-imasu-ka?
DeepSemPat-Watashi-wa-anata-ni-?Form-Noun-o-okurimasu
DeepSemPat-/?Conditional-Rest/-?Verb-Phrase-kudasai
DeepSemPat-Anata-wa-watashi-ni-?Noun-Ni-Tsuite-/Object-o/-oshiete-kudasai
DeepSemPat-Watashi-no-juusho-wa-?Address-desu
DeepSemPat-Watashi-no-denwa-bango-wa-?Number-desu
DeepSemPat-Kochira-wa-?Name-desu
DeepSemPat-Watashi-no-namae-wa-?Name-Rest-desu
DeepSemPat-Wakaranai-ten-ga-areba-o-kiki-kudasai
DeepSemPat-/?Time-Rest/-Touroku-youshi-de-tetsuzuki-o-shite-nakute-wa-ikemasen
DeepSemPat-/?Conditional-Rest/-?Object-wa-?Wh-Word-desu-ka?         ;OBSOLETE
DeepSemPat-/?Conditional-Rest/-?Wh-Word-?Verb-ka?
DeepSemPat-/?Conditional-Rest/-?Wh-Word-ga-chikai-desu-ka?
DeepSemPat-/?Conditional-Rest/-?Object-wa-?Wh-Word-kakarimasu-ka?
DeepSemPat-/?Conditional-Rest/-?Object-wa-?Wh-Word-?Arg-?Identity-desu-ka?
DeepSemPat-/?Conditional-Rest/-?Cost-kakarimasu
DeepSemPat-/?Conditional-Rest/-?Object-wa-?Identity-desu
DeepSemPat-/?Conditional-Rest/-?Object-wa-?Time-desu
DeepSemPat-?Object-wa-?Need-Verb-desu-ka?
DeepSemPat-?Object-wa-arimasu-ka?
DeepSemPat-?As-Fee-?Money-wa-?Need-Verb
DeepSemPat-?Object-wa-?Container-ni-fukumarete-imasu     ;See next
DeepSemPat-?Object-wa-?Location-ni-fukunde-imasu         ;See prev.
                                 ;COMBINE THESE?
DeepSemPat-/?Connective-Rest/-/?Agent/-?Discount-wa-okonatte-imasen
DeepSemPat-Watashi-wa-/?Manner-Rest/-?Object-o-wakarimasen
DeepSemPat-/?Connective-Rest/-?Object-o-youi-shite-imasu
DeepSemPat-/?Time-Rest/-?Location-?Object-wa-kaisaisaremasu
DeepSemPat-?Statement-ne
DeepSemPat-/?Connective-Rest/-Onegai-ga-arimasu-ga
                                 ;Includes ga-aru-no-desu-ga
DeepSemPat-Anata-wa-?Verb-Phrase-ga-dekimasu-ka?
DeepSemPat-/?Connective-Rest/-?Verb-Phrase-ga-dekimasu-ka?
DeepSemPat-/?Connective-Rest/-?Object-o-Dekimasu
DeepSemPat-/?Connective-Rest/-?Object-o-Dekimasen
DeepSemPat-/?Connective-Rest-ni/-kaite-imasu-ga
DeepSemPat-/?Connective-Rest/-ni-notte-imasu
DeepSemPat-/?Conditional-Rest/-?Object-o-oshirase-imasu
DeepSemPat-/?Conditional-Rest/-?Object-o-kengaku-shimasu
DeepSemPat-?Agent-wa-?Location-ni-Sanka-shimasu-ka?
DeepSemPat-?Verb-phrase-yotei-desu
DeepSemPat-Watashi-wa-/?Connective-Rest/-Sanka-shimasu
DeepSemPat-Okurisaki-wa-?Address-Rest--?Name-Rest-desu
DeepSemPat-/?Conditional-Rest/-Kikitai-?Object-ga-aru-desu-ga
          ;Doesn't pick up the SURU, NO, or DESU. Object is Mono or Koto, (NULL).
DeepSemPat-Review-?Review-Rest-and-Send-?Send-Rest
DeepSemPat-/?Conditional-Rest/-?Object-o-doufuu-shimasu
DeepSemPat-/?Connective-Rest/-?Arg1-no-?Arg2-Onegaishimasu
DeepSemPat-/?Connective-Rest/-?Time-Onegaishimasu
DeepSemPat-/?Connective-Rest/-Watashi-wa-?Location-ni-imasu
DeepSemPat-/?Connective-Rest/-?Object-shirabemasu
DeepSemPat-/?Connective-Rest/-?Object-o-otori-shimasu
```

Patterns are represented in the following format:

```
(ABDUCK-FS-MACRO

  DeepSemPat-Dono-youna-?Verb-desu-ka?         ;See the next two.
  "D:Dono-youna-?Verb-desu-ka?"
```

```
[[SEM [[RELN  S-REQUEST]
       [AGEN !X06[[LABEL  *SPEAKER*]]]
       [RECP !X05[[LABEL  *HEARER*]]]
       [OBJE [[RELN  INFORMREF]
              [AGEN !X05]
              [RECP !X06]
              [OBJE [[PARM !X08[[RELN  どのような-1]
                                 [ARG-1 [[PARM !X01[]]
                                         [RESTR [[RELN  ?verb]
                                                 [ENTITY !X01]]]]]]]
                ;Deshou resolved to desu.  This representation is not so good, and it is unclear
                ;just why the da-identical is needed in the first place.  Other similar sentences
                ;are represented in quite a different manner.
                     [RESTR  [[RELN  だ-IDENTICAL]
                              [OBJE !X07[]]
                              [IDEN !X08]]]

                   ]]]]]]
 ]
)
```

## 12.3 Surface Semantics Training Data

The full information is found in `LM01:>myers>convn>real-rough-surf-macros`.

```
MACRO NAMES:
        SSemPat-Moshi-Moshi
        SSemPat-Hai
        SSemPat-Iie
        SSemPat-Ii-desu-yo
        SSemPat-Sou-Desu
        SSemPat-Sou-Desu-Ka
        SSemPat-Mada-Desu
        SSemPat-Nan-deshou-ka?
        SSemPat-Nan-no-go-youken-deshou-ka?
        SSemPat-Motte-imasen
        SSemPat-Wakarimashita
        SSemPat-/?Degree-Rest/-Arigatou
        SSemPat-Dou-Itashi-Mashite
        SSemPat-Shitsurei-Shimasu      ;Also includes Soredewa, Shitsurei-Shimasu; Dewa, Sh-; and Doumo Sh-.
        SSemPat-/Chotto/-Matte-kudasai
        SSemPat-/?Connective-Rest/-Matte-imasu
        SSemPat-/?Connective-Rest/-Sayounara
        SSemPat-/?Connective-Rest/-Yoroshiku-Onegaishimasu
             ;Also might include Douzo Yoroshiku Onegaishimasu
        SSemPat-/?Connective-Rest/-?Object-o-Yoroshiku-Onegaishimasu
        SSemPat-Kochira-wa-?Name-Desu
        SSemPat-Sochira-wa-?Name-Desu-Ka
        SSemPat-Dono-youna-?Verb-/?Deshou-Rest/-ka?  ;See next two.
        SSemPat-Dono-youna-?Noun-de-?Verb-Phrase-eba-ii-desu-ka?
    ;Where did the "de" go to in the parse?
        SSemPat-Dono-youni-?Verb-yoi-desu-ka?        ;See previous and next.
        SSemPat-Dou-?Verb-eba-yoroshii-desu-ka?
         ;Current parser does not pick up "deshou".  See prev two.
        SSemPat-/?Connective-Rest/-Go-Juushou-to-O-Namae-Onegaishimasu  ;See next
        SSemPat-/?Connective-Rest/-O-Namae-to-Go-Juushou-Onegaishimasu  ;See prev
        SSemPat-/?Connective-Rest/-Denwa-Bango-Onegaishimasu
              ;Note parser does not pick up the "MO".
        SSemPat-/?Connective-Rest/-Okurisaki-o-Onegaishimasu
        SSemPat-/?Connective-Rest/-?Arg1-no-?Arg2-Onegaishimasu
        SSemPat-/?Connective-Rest/-?Time-Onegaishimasu
        SSemPat-/?Connective-Rest/-O-Namae-to-Ninzuu-Onegaishimasu      ;Number of people in your party
        SSemPat-Juusho-wa-?address-Desu
        SSemPat-Denwa-Bango-wa-?number-Desu
        SSemPat-?Owner-Name-no-?Name-Rest-Desu   ;See next
        SSemPat-?Name-Rest-Desu                        ;See prev
        SSemPat-Named-?Name-Rest-Desu
        SSemPat-Namae-wa-?name-rest-Desu
        SSemPat-/?Place-Or-Object-Rest/-ni-?Apply-Attend-Verb-tai-desu-ga                  ;See next.
        SSemPat-/?Place-Or-Object-Rest/-ni-?Apply-Attend-Verb-tai-to-omotte-iru-desu-ga
               ;Parser does not pick up "no".  See prev.
        SSemPat-/?Place-Or-Object-Rest/-ni-?Apply-Attend-Verb-tai-to-omoimasu
                   Parser does not pick up "no".
        SSemPat-/?Conditional-Rest/-?Object-o-kikitai-desu-ga
                     ;Doesn't pick up the NO.  See next.
        SSemPat-/?Conditional-Rest/-?Object-o-ukagai-dekimasu-deshou-ka?
                                    ;See prev.
        SSemPat-/?Conditional-Rest/-Kikitai-?Object-ga-aru-desu-ga
                        ;Doesn't pick up the SURU, NO, or DESU.
              This is different in that it doesn't really have an object as an arg, whereas the previous two do.
        SSemPat-?Noun-Ni-Tsuite-/?Object-o/-oshiete-itadakitai-desu-ga
                ;See next two.  Doesn't pick up the NO.
        SSemPat-?Noun-Ni-Tsuite-/?Object-o/-oshiete-kudasai
                ;See prev and next.  Doesn't pick up the NO.
        SSemPat-/?Conditional-Rest/-?Noun-Ni-Tsuite-shitsumon-shitai-desu-ga
                 ;Doesn't pick up the NO.  See prev two.
        SSemPat-/?Conditional-Rest/-?Noun-Ni-Tsuite-tazune-shitai-desu-ga
                 ;Doesn't pick up the NO in NO DESU GA.
        SSemPat-?Object-wa-/?time-rest/-omochi-deshou-ka?
                 ;See next pattern too
        SSemPat-?Object-wa-/?time-rest/-omotte-imasu-ka?
                 ;See previous pattern too
```

```
SSemPat-/?Conditional-Rest/-?Verb-Phrase-kudasai
SSemPat-/?Conditional-Rest/-?Object-o-?Verb-itadakEnai-deshou-ka?
SSemPat-/Soredewa/-?Form-Noun-o-okurimasu
        ;See next pattern too.  Also includes o-okuri-itashimasu.
SSemPat-/Soredewa/-?Form-Noun-o-okurasete-itadakimasu
        ;See previous pattern too
SSemPat-Wakaranai-ten-ga-areba-o-kiki-kudasai
        ;Includes wakaranai ten ga gozaimashitara itsudemo o-kiki-kudasai

SSemPat-/?Time-Rest/-Touroku-youshi-de-tetsuzuki-o-shite-itadakanakute-wa-narimasen-ga
        ;From B-5
SSemPat-?Verb-Phrase-o-modoshite-itadakEmasu-ka?
SSemPat-/?Connective-Rest/-?Verb-Phrase-ga-dekimasu-ka?
SSemPat-/?Conditional-Rest/-?Object-wa-?Wh-Word-desu-ka? ;OBSOLETE
SSemPat-/?Conditional-Rest/-?Wh-Word-?Verb-ka?
SSemPat-/?Conditional-Rest/-?Wh-Word-ga-chikai-desu-ka?
SSemPat-/?Conditional-Rest/-?Object-wa-?Wh-Word-kakarimasu-ka?
SSemPat-/?Conditional-Rest/-?Object-wa-?Wh-Word-?Arg-?Identity-desu-ka?
SSemPat-/?Conditional-Rest/-?Cost-kakarimasu
SSemPat-/?Conditional-Rest/-?Object-wa-?Identity-desu
SSemPat-/?Conditional-Rest/-?Object-wa-?Identity-desu-ga
SSemPat-?Object-wa-?Need-Verb-desu-ka?
        ;Note the parser currently drops the deshou.
SSemPat-?Object-wa-nai-desu-ka?
       ;Note the parser currently drops the "no".  See next.  Includes ?cond-rest.
SSemPat-?Object-wa-arimasu-ka?                    ;See prev.  Includes ?cond-rest.
SSemPat-?As-Fee-?Money-wa-?Need-Verb
SSemPat-?Object-wa-?Container-ni-fukumarete-imasu      ;See next.
SSemPat-?Object-wa-?Location-ni-fukunde-imasu        ;See prev.
SSemPat-/?Connective-Rest/-/?Agent/-?Discount-wa-okonatte-imasen
SSemPat-Watashi-wa-/?Manner-Rest/-?Object-o-wakarimasen-ga
SSemPat-/?Connective-Rest/-?Object-o-youi-shite-imasu
       ;includes youi-shite-orimasu
SSemPat-/?Time-Rest/-?Location-?Object-wa-kaisaisaremasu
SSemPat-?Statement-ne
SSemPat-/?Connective-Rest/-Onegai-ga-arimasu-ga
        ;Includes ga-aru-no-desu-ga
SSemPat-/?Connective-Rest/-?Object-o-Dekimasen     ;See next
SSemPat-/?Connective-Rest/-?Object-o-Dekimasu      ;See prev
SSemPat-/?Connective-Rest-ni/-kaite-imasu-ga
SSemPat-/?Connective-Rest/-ni-notte-imasu-ga
SSemPat-/?Connective-Rest/-ni-notte-imasu
SSemPat-/?Content-Rest/-mondai-nai
SSemPat-/?Conditional-Rest/-?Object-o-oshirase-imasu
SSemPat-/?Conditional-Rest/-?Object-ga-aru-sou-desu-ga
SSemPat-/?Conditional-Rest/-?Object-o-kengaku-shimasu
SSemPat-?Agent-wa-?Location-ni-Sanka-shimasu-ka?
SSemPat-?Verb-phrase-yotei-desu
SSemPat-/?Connective-Rest/-Sanka-shimasu
SSemPat-Okurisaki-wa-?Address-Rest--?Name-Rest-desu
SSemPat-Review-?Review-Rest-and-Send-?Send-Rest
SSemPat-/?Conditional-Rest/-?Object-o-doufuu-shimasu
SSemPat-/?Connective-Rest/-?Object-?Goal-chikai-desu-ga
SSemPat-/?Connective-Rest/-?Object-shirabemasu
SSemPat-/?Connective-Rest/-?Object-o-otori-shimasu
```

Surface macros are of the form:
(includes normal variables and "rest" variables)

```
(ABDUCK-FS-MACRO

  SSemPat-Dono-youna-?Noun-de-?Verb-Phrase-eba-ii-desu-ka?      ;Where did the "de" go to in the parse?
  "SSemPat-Dono-youna-?Noun-de-?Verb-Phrase-eba-ii-desu-ka?"

[[SEM !X20[[RELN  S-REQUEST]
          [AGEN !X07[[LABEL  *SPEAKER*]]]
          [RECP !X08[[LABEL  *HEARER*]]]
          [OBJE [[RELN  INFORMREF]
                 [AGEN !X08]
                 [RECP !X07]
                 [OBJE [[PARM !X11[[RELN  どのような-1]
                                  [ARG-1  [[PARM !X05[]]
```

```
                                          [RESTR [[RELN  ?noun]
                                                  [ENTITY !X05]]]]]]
                      [RESTR !X33[[RELN  ばいい-SHOULD]
                                  [AGEN !X09[]]
                                  [OBJE ?verb-phrase]
                                  [ASPT  STAT]]]]]]]]]
 ?prag-rest
 ]

)
```

## 12.4 "Unagi-Da" Inference Rules Training Data

This example is taken from file `LM01:>myers>convn>real-unagi-da-rules`.

Note that there is a latent bug in the H. feature-structure instantiator: although regular variables that are unbound are properly left alone, "rest" variables that are unbound are nulled out during instantiation. This causes a major difficulty when trying to partially instantiate the unagi-da patterns: basically, with the current instantiator, "rest" variables can't be used. The instantiator should be fixed in the next version. (Duplicate it in case old code uses the previous feature.)

This example shows the form of the unagi-da rules.

```
(START-TRAINING-UNAGI-DA-RULES)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;UNAGI-DA RULES;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;  This pattern is applied to the RESULTS of the BEST RECOGNITION,
;;    not to the INPUT.
;;  This means that the First part of the pair (the question)
;;    must be recognized properly in order for the unagi-da rule to trigger.
;;
;;  Matching (verification) is performed based on the Illoc Force and the Deep Semantics.
;;   The Surface Semantics of the Antecedent Rule is not used.

(START-NEXT-UNAGI-DA-ANTECEDENT)       ; Rule #1 antecedent, used for conv #2 utt #13

(Illoc-Force
       Ill-F-Directive-AskYN
)

(Deep-Semantic-Pattern
[[SEM     [[RELN  S-REQUEST]
           [AGEN !X03[[LABEL  *SPEAKER*]]]
           [RECP !X04[[LABEL  *HEARER*]]]
           [OBJE [[RELN  INFORMREF]
                  [AGEN !X04]
                  [RECP !X03]
                  [OBJE [[PARM !X09[[PARM !X07[]]
                                    [RESTR [[RELN  どのように-1]
                                            [ENTITY !X07]]]]]
                         [RESTR !X33[[RELN  たらよい-SHOULD]
                                     [AGEN !X10[]]
                                     [ASPT  STAT]
                                     [OBJE !X11[[RELN  ?action]
                                                [AGEN !X10]
                                                [MANN !X09]
                                                [OBJE ?object]]]]]]]]]
  ]
 ]
)


;Following is currently not used:
(Surface-Semantic-Pattern
[[SEM     [[RELN  S-REQUEST]
           [AGEN !X03[[LABEL  *SPEAKER*]]]
           [RECP !X04[[LABEL  *HEARER*]]]
           [OBJE [[RELN  INFORMREF]
                  [AGEN !X04]
                  [RECP !X03]
                  [OBJE [[PARM !X09[[PARM !X07[]]
                                    [RESTR [[RELN  どのように-1]
                                            [ENTITY !X07]]]]]
                         [RESTR !X33[[RELN  たらよい-SHOULD]
                                     [AGEN !X10[]]
                                     [ASPT  STAT]
                                     [OBJE !X11[[RELN  ?action]
                                                [AGEN !X10]
                                                [MANN !X09]
                                                [OBJE ?object]]]]]]]]]
  ]
  ?prag-rest
 ]
)
```

```
;---------------------------------------------------------------------------
(START-NEXT-UNAGI-DA-CONSEQUENT)        ; Rule #1 consequent

(Illoc-Force
        Ill-F-DIRECTIVE-DIRECT
)

(Deep-Semantic-Pattern
[[SEM [[RELN  下さい-REQUEST]
       [ASPT  UNRL]
       [AGEN !X07[[LABEL  *SPEAKER*]]]
       [RECP !X08[[LABEL  *HEARER*]]]
       [OBJE

          [[RELN  ?action]
           [AGEN  !X08]
           [OBJE  ?object]
           [MANN  ?method]
          ]

      ]]
  ]
 ]
)

; Here's the recognizer.  This gets mostly instantiated by the program before it's
;applied to the input.
(Surface-Semantic-Pattern
[[SEM     [[RELN  だ-IDENTICAL]
           [ASPT  STAT]
           [OBJE ?object]
           [IDEN ?method]
          ]
  ]
;;;; ?new-prag-rest
;;;;  Bug in FS instantiator:  Rest variables are nulled out if unbound.  Fix it later.
  [PRAG ?prag]
  [WH   ?wh]
]
)

...

(END-TRAINING-UNAGI-DA-RULES)
```

# 13 Conversations Used for Training

Only a brief example is given in the interests of space. Interested users can check the listings of the other conversations on disk, in directory `LM01:>myers>convn>*`.

## 13.1 Example: Training Conversation Number A

```
(START-TRAINING-CONV-# 'A )
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CONVA;;;;;;;;;;;;;;;;;;;;;;;;
;--------------------------------------------------------------------------
;      もしもし
(START-NEXT-TRAINING-UTTERANCE)         ; Utt #1

(Illoc-Force
        Ill-F-Expressive-Greet
)

(Deep-Semantic-Pattern
        DeepSemPat-Moshi-Moshi
)

(Surface-Semantic-Pattern
  SSemPat-Moshi-Moshi
)
;other alternative patterns:
(Surface-Semantic-Pattern
  SSemPat-Hai
)

;--------------------------------------------------------------------------
;      そちら会議事務局ですか
(START-NEXT-TRAINING-UTTERANCE)         ; Utt #2

(Illoc-Force
        Ill-F-Directive-AskYN
)

(Deep-Semantic-Pattern
        DeepSemPat-Anata-wa-?Name-desu-ka     ;Anata resolved
)

(Surface-Semantic-Pattern
  SSemPat-Sochira-wa-?Name-Desu-Ka
)
;other alternative patterns:
(Surface-Semantic-Pattern
  SSemPat-?Name-Desu-Ka
)


...

(END-TRAINING-CONV)
```

# 14 Verification of Training Conversations

In order to have the system be able to score itself automatically, the first surface semantic pattern of each training conversation utterance must be guaranteed to match the correct utterance. And, the correct utterance must be guaranteed to be the first utterance presented in the (unordered) list of alternatives. Using this information, the system can check to make sure afterwards how accurate it is.

The only problem is with the utterances that are the output of the "unagi-da" rule. This does not have to match, because the correct one is instantiated at run-time.

Here are the results of the check. The data should be checked every time a new version is implemented, to make sure there are no gross errors. Checking is done using `(filedump-check-all-surf-patterns)`, which dumps the result to `LM01:>myers>ABDUCK>surf-checks-results.txt`, or using `check-all-surf-patterns`, which dumps the results to stream OS (typically bound to T, indicating the user's terminal). The functions are in file `dis-checkout` which is loaded automatically with the system.

Note that conversation A is #0, conversation B is #11, and the unagi-da rules are conversation #12. Also note that the one "mistake" in conversation #2 is where the unagi-da rule fills in the actual pattern at run-time.

```
 CHECKING SURFACE PATTERNS AGAINST INPUTS
Only the first pattern and the first input for each utterance is checked.
These are guaranteed to be correct (except for unagi-da bun) by the designer.


Conv 0.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19
Conv 1.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
Conv 2.    :  1  2  3  4  5  6  7  8  9  10  11  12  13

MISMATCHED PATTERN IN CONVERSATION 2, UTTERANCE # 14.


PATTERN:  NIL
 [[SEM [[RELN 下さい-REQUEST]
       [ASPT UNRL]
...

INPUT: 参加料は銀行振り込みです
 [[SEM !X3[[RELN だ-IDENTICAL]
          [ASPT STAT]
...

 15  16  17  18  19  20  21
Conv 3.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16
Conv 4.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22
Conv 5.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22
Conv 6.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
Conv 7.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21
Conv 8.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25
Conv 9.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17
Conv 10.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25
  26  27  28  29  30  31  32  33  34  35  36  37  38
Conv 11.    :  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18
```

# 15 Example Analysis of Output (Closed Training)

## 15.1 Analysis of Output (Closed Training) for Conversation Number A

```
FINAL SCORE FOR CONVERSATION *A*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 0     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  19  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  145.5 SECONDS
   FOR AN AVERAGE OF 7.6578946 SECONDS PER UTTERANCE.
NO MISTAKES.  NO ANALYSIS FOLLOWS.
```

## 15.2 Analysis of Output (Closed Training) for Conversation Number B

```
FINAL SCORE FOR CONVERSATION *B*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 3     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  18  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  681.5 SECONDS
   FOR AN AVERAGE OF 37.86111 SECONDS PER UTTERANCE.


ANALYSIS OF MISTAKES FOLLOWS:


 ANSWER FOR UTTERANCE # 4  WAS  ***(deep) Incorrect***

***Deep Pattern Used By System To Answer:***
 [[SEM [[RELN S-REQUEST]
        [AGEN !X2[[LABEL *SPEAKER*]]]
        [RECP !X1[[LABEL *HEARER*]]]
        [OBJE [[RELN INFORMREF]
               [AGEN !X1]
               [RECP !X2]
               [OBJE [[PARM !X4[[RELN どのような-1]
                                [ARG-1 [[PARM !X3[]]
                                        [RESTR [[RELN ?VERB]
                                                [ENTITY !X3]]]]]]]
                      [RESTR [[RELN だ-IDENTICAL]
                              [OBJE []]
                              [IDEN !X4]]]]]]]]]]

***Actual, Correct Deep Pattern To Form Answer:***
 [[SEM [[RELN S-REQUEST]
        [AGEN !X2[[LABEL *SPEAKER*]]]
        [RECP !X1[[LABEL *HEARER*]]]
        [OBJE [[RELN INFORMREF]
               [AGEN !X1]
               [RECP !X2]
               [OBJE [[PARM !X4[[PARM !X3[]]
                                [RESTR [[RELN どう-1]
                                        [ENTITY !X3]]]]]
                      [RESTR [[RELN ばよい-SHOULD]
                              [ASPT STAT]
                              [AGEN !X2]
```

```
                              [OBJE [[RELN ?VERB]
                                     [AGEN !X2]
                                     [OBJE []]
                                     [MANN !X4]]]]]]]]]]]]]

 ANSWER FOR UTTERANCE # 6  WAS  ***[SURFACE] INCORRECT***

***System's [Surface] Answer:***
  [SEM [[RELN S-REQUEST]
        [AGEN !X5[[LABEL *SPEAKER*]]]
...

***Actual, Correct [Surface] Answer:***
 [[SEM [[RELN S-REQUEST]
        [AGEN !X2[[LABEL *SPEAKER*]]]
...
```

## 15.3 Analysis of Output Scores (Closed Training) for Conversations Number 1-10

```
FINAL SCORE FOR CONVERSATION *1*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 0    ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  20  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  165.0 SECONDS    FOR AN AVERAGE OF 8.25 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *2*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 6    ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 3    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  21  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  209.25 SECONDS    FOR AN AVERAGE OF 9.964286 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *3*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 2    ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  16  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  129.5 SECONDS    FOR AN AVERAGE OF 8.09375 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *4*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 5    ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  22  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  460.0 SECONDS    FOR AN AVERAGE OF 20.90909 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *5*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 4    ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  22  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  185.0 SECONDS    FOR AN AVERAGE OF 8.409091 SECONDS PER UTTERANCE.
```

```
FINAL SCORE FOR CONVERSATION *6*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 1     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  23  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  158.0 SECONDS   FOR AN AVERAGE OF 6.869565 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *7*:

 2 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 7     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 4     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  21  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  565.5 SECONDS   FOR AN AVERAGE OF 26.928572 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *8*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 3     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  25  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  235.0 SECONDS   FOR AN AVERAGE OF 9.4 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *9*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 5     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  17  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  295.5 SECONDS   FOR AN AVERAGE OF 17.382353 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *10*:

 0 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 1     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  38  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  249.0 SECONDS   FOR AN AVERAGE OF 6.5526314 SECONDS PER UTTERANCE.
```

# 16 Example Analysis of Output (Open Training)

## 16.1 Analysis of Output (Open Training) for Conversation Number A

```
FINAL SCORE FOR CONVERSATION *A*:

 2 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 3     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  19  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  592.25 SECONDS
   FOR AN AVERAGE OF 31.171053 SECONDS PER UTTERANCE.


ANALYSIS OF MISTAKES FOLLOWS:


 ANSWER FOR UTTERANCE # 5  WAS  ***(illoc) Incorrect***

***Illoc Force Pattern Used By System To Answer:***
 [[ILLOC [[IL-CAT ASSERTIVE]
         [IL-PRIM INFORM]]]]

***Actual, Correct Illoc Force Pattern To Form Answer:***
 [[ILLOC [[IL-CAT ASSERTIVE]
         [IL-PRIM WANT-TO-DO]]]]


 ANSWER FOR UTTERANCE # 16  WAS  ***(illoc) Incorrect***  ***(deep) Incorrect***  ***GUESSED***

***Illoc Force Pattern Used By System To Answer:***
NIL

***Actual, Correct Illoc Force Pattern To Form Answer:***
 [[ILLOC [[IL-CAT COMMISSIVE]
         [IL-PRIM PASSIVE-OFFER]]]]

***Deep Pattern Used By System To Answer:***
NIL

***Actual, Correct Deep Pattern To Form Answer:***
 [[SEM [[RELN 下さい-REQUEST]
       [ASPT UNRL]
       [AGEN !X2[[LABEL *SPEAKER*]]]
       [RECP !X1[[LABEL *HEARER*]]]
       [OBJE [[RELN 聞く-3]
...
```

## 16.2 Analysis of Output Scores (Open Training) for Conversations Number B-10

Note that these were run under a heavily loaded machine, and the processing times are slightly high.

```
FINAL SCORE FOR CONVERSATION *B*:

 3 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 7     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 4     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  18  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
```

```
TOOK  2226.75 SECONDS
   FOR AN AVERAGE OF 123.708336 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *1*:

 2 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 3     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  20  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  674.625 SECONDS
   FOR AN AVERAGE OF 33.73125 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *2*:

 2 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 8     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 6     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  21  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  1707.75 SECONDS
   FOR AN AVERAGE OF 81.32143 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *3*:

 5 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 10     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 6     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  16  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  1099.125 SECONDS
   FOR AN AVERAGE OF 68.69531 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *4*:

 3 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 11     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 5     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1     SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  22  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  2160.875 SECONDS
   FOR AN AVERAGE OF 98.22159 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *5*:

 5 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 12     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 7     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.
```

```
 1    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  22  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  2138.625 SECONDS
   FOR AN AVERAGE OF 97.21023 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *6*:

 9 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 15     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 10     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  23  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  1775.125 SECONDS
   FOR AN AVERAGE OF 77.179344 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *7*:

 9 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 15     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 11     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  21  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  1749.375 SECONDS
   FOR AN AVERAGE OF 83.30357 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *8*:

 9 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 15     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 11     DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 0    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  25  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  1938.25 SECONDS
   FOR AN AVERAGE OF 77.53 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *9*:

 6 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.

 12     ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 6    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 1    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  17  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  1439.0 SECONDS
   FOR AN AVERAGE OF 84.64706 SECONDS PER UTTERANCE.


FINAL SCORE FOR CONVERSATION *10*:

 9 UTTERANCES WERE NOT RECOGNIZED AND HAD TO BE GUESSED USING INPUT PROBABILITIES.
```

```
 24    ILLOCUTIONARY FORCE UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 17    DEEP-SEMANTIC UTTERANCES WERE INCORRECT, INCLUDING GUESSES.

 2    SURFACE-SEMANTIC UTTERANCES WERE INCORRECT.

 UTTERANCE PROCESSING FOR  38  UTTERANCES, NOT INCLUDING TRAINING INITIALIZATION TIME,
TOOK  2434.25 SECONDS
   FOR AN AVERAGE OF 64.05921 SECONDS PER UTTERANCE.
```

# 17 Example Full Output (Long Version)

This section has been cut in the interests of space, to only give the results for conversation number 2. Conversation 2 is presented because it is perhaps the most interesting, containing the one unagi-da utterance in the corpus.

## 17.1 Full Output for Conversation Number 2

```
;  ABDUCK SYSTEM RUN FOR CONVERSATION #  2  .
;Illocutionary Force Macros Version:    RR5.2.
;Deep Semantics Macros Version:         RR5.2.
;Surface Semantics Macros Version:      RR4.8.
;Unagi-Da etc. Inference Rules Version: 2.2.


STARTING TIMING FOR THE RUN.


*************STARTING UTTERANCE # 1 ***************


Input candidate:
(0.7 はい [[SEM [[RELN はい-AFFIRMATIVE]
        [AGEN !X1[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]
        [ASPT -]]]
  [PRAG [[SPEAKER !X1]
         [HEARER !X2]]]])


Input candidate:
(0.12八 [[SEM [[RELN 八-NUMBER]
        [ASPT -]
        [AGEN !X1[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]]]
  [PRAG [[SPEAKER !X1]
         [HEARER !X2]]]])


*****REPORTING ANSWER FOR UTTERANCE # 1 .*****


RECOGNIZED ILLOC FORCE is:
 [[ILLOC [[IL-CAT EXPRESSIVE]
          [IL-PRIM GREET]]]]

THE RESOLVED DEEP SEMANTIC STRUCTURE  is:
 [[SEM [[RELN もしもし-OPEN_DIALOGUE]
        [ASPT -]
        [AGEN [[LABEL *SPEAKER*]]]
        [RECP [[LABEL *HEARER*]]]]]]

THE RESULT (score = 4.0599594) is:
 [[PRAG [[SPEAKER [[LABEL *SPEAKER*]]]
         [HEARER [[LABEL *HEARER*]]]]]
  [SEM [[RELN はい-AFFIRMATIVE]
        [ASPT -]
        [AGEN [[LABEL *SPEAKER*]]]
        [RECP [[LABEL *HEARER*]]]]]]


*************STARTING UTTERANCE # 2 ***************


Input candidate:
(0.189 こちらは会議事務局です [[SEM !X3[[RELN だ-IDENTICAL]
          [OBJE !X2[[LABEL *SPEAKER*]]]
```

```
          [IDEN [[RESTR [[RELN NAMED]
                         [IDEN 会議事務局-1]
                         [ENTITY !X1[]]]]
                 [PARM !X1]]]
          [ASPT STAT]]]
 [PRAG [[RESTR [[IN []]
                [OUT []]]]
        [SPEAKER !X2]
        [HEARER [[LABEL *HEARER*]]]
        [TOPIC [[IN [[REST []]
                     [FIRST [[FOCUS !X2]
                             [TOPIC-MOD HA]
                             [SCOPE !X3]]]]]
                [OUT []]]]
        [PRSP-TERMS [[IN []]
                     [OUT []]]]
        [ASPE [[IN []]
               [OUT []]]]]]
 [WH []]])


Input candidate:
(0.002 こちらは会議事務局ですか [[SEM [[RELN S-REQUEST]
        [AGEN !X2[[LABEL *SPEAKER*]]]
        [RECP !X1[[LABEL *HEARER*]]]
        [OBJE [[RELN INFORMIF]
               [AGEN !X1]
               [RECP !X2]
               [OBJE !X5[[RELN だ-IDENTICAL]
                         [ASPT STAT]
                         [OBJE !X2]
                         [IDEN [[PARM !X3[]]
                                [RESTR [[RELN NAMED]
                                        [ENTITY !X3]
                                        [IDEN 会議事務局-1]]]]]]]]]]]
 [PRAG [[RESTR [[IN [[FIRST [[RELN POLITE]
                             [AGEN !X2]
                             [RECP !X1]]]
                     [REST !X4[]]]]
                [OUT !X4]]]
        [TOPIC [[IN [[FIRST [[FOCUS !X2]
                             [TOPIC-MOD HA]
                             [SCOPE !X5]]]
                     [REST []]]]
                [OUT []]]]
        [PRSP-TERMS [[IN []]
                     [OUT []]]]
        [SPEAKER !X2]
        [HEARER !X1]]]]])


*****REPORTING ANSWER FOR UTTERANCE # 2 .*****


RECOGNIZED ILLOC FORCE is:
 [[ILLOC [[IL-CAT DECLARATIVE]
          [IL-PRIM SELF-IDENTIFY]]]]

THE RESOLVED DEEP SEMANTIC STRUCTURE  is:
 [[SEM [[ASPT STAT]
        [OBJE [[LABEL *SPEAKER*]]]
        [IDEN [[RESTR [[IDEN 会議事務局-1]
                       [RELN NAMED]
                       [ENTITY !X1[]]]]
               [PARM !X1]]]
        [RELN だ-IDENTICAL]]]]

THE RESULT (score = 0.7181856) is:
 [[WH []]
  [PRAG [[RESTR [[IN []]
                 [OUT []]]]
         [SPEAKER !X1[[LABEL *SPEAKER*]]]
```

```
        [HEARER [[LABEL *HEARER*]]]
        [TOPIC [[IN [[REST []]
                      [FIRST [[FOCUS !X1]
                              [TOPIC-MOD HA]
                              [SCOPE [[RELN だ-IDENTICAL]
                                      [OBJE !X1]
                                      [IDEN [[RESTR [[RELN NAMED]
                                                     [IDEN !X3 会議事務局-1]
                                                     [ENTITY !X2[]]]]
                                             [PARM !X2]]]
                                      [ASPT STAT]]]]]]]
                [OUT []]]]
        [PRSP-TERMS [[IN []]
                      [OUT []]]]
        [ASPE [[IN []]
               [OUT []]]]]]
  [SEM [[RELN だ-IDENTICAL]
       [IDEN [[PARM !X4[]]
              [RESTR [[IDEN !X3]
                      [RELN NAMED]
                      [ENTITY !X4]]]]]
       [OBJE [[LABEL *SPEAKER*]]]
       [ASPT STAT]]]]

*************STARTING UTTERANCE # 3 ***************

...
```

# 18 Conclusion

The third verson of the ABDUCK system is still very primitive, but it is able to disambiguate utterance candidates and understand the deep meanings of simple utterances and of simple unagi-da sentences. It does this by applying as much knowledge as possible to the problem. Future versions need to integrate a plan-recognition system, a script understanding system, typed fuzzy feature structures, and true agent-mind simulators in order to come up with a system that can offer truly powerful, natural understandings and translations of conversational utterances.

# References

[MT90] John K. Myers and Takashi Toyoshima. Known current problems in automatic interpretation: Challenges for language understanding. Technical Report TR-I-0128, ATR Interpreting Telephony Research Laboratories, Kyoto, Japan, January 1990.

[Mye92a] John K. Myers. An agent-based approach to natural-language understanding of conversations for an interpreting telephone. In *International Symposium on Natural Language Understanding and AI (NLU +AI), as a part of the International Symposia on Information Sciences (ISKIT'92)*, pages 211–218, Kyuushuu Institute of Technology, July 1992.

[Mye92b] John K. Myers. A theory of intentional action under uncertainty as applied to physical and communicative actions. In Hisato Kobayashi, editor, *Proceedings of the IEEE Workshop on Robot and Human Communication: RO-MAN '92*, pages 305–310, Hosei University, Tokyo, September 1992.