ADVANCES IN NATURAL MULTIMODAL DIALOGUE SYSTEMS

ADVANCES IN NATURAL MULTIMODAL DIALOGUE SYSTEMS

Edited by
JAN VAN KUPPEVELT
The Netherlands

LAILA DYBKJÆR
NISLab, University of Southern Denmark

NIELS OLE BERNSEN
NISLab, University of Southern Denmark

Kluwer Academic Publishers
Boston/Dordrecht/London

Contents

Chapter 1

MACHINE LEARNING APPROACHES TO HUMAN DIALOGUE MODELLING

Yorick Wilks, Nick Webb, Andrea Setzer, Mark Hepple and Roberta Catizone
Natural Language Processing Group
Department of Computer Science
University of Sheffield, UK
{y.wilks, n.webb, a.setzer, m.hepple, r.catizone}@dcs.shef.ac.uk

Abstract We describe two major dialogue system segments: the first is an analysis module that learns to assign dialogue acts from corpora, but on the basis of limited quantities of data, and up to what seems to be some kind of limit on this task, a fact we also discuss. Secondly, we describe a Dialogue Manager which uses a representation of stereotypical dialogue patterns that we call Dialogue Action Frames, which are processed using simple and well understood algorithms, which are adapted from their original role in syntactic analysis role, and which, we believe, generate strong and novel constraints on later access to incomplete dialogue topics.

Keywords: Dialogue management, machine learning, dialogue acts, dialogue modelling, Dialogue Action Frames.

1. Introduction

Computational modelling of human dialogue is an area of NLP where there are still a number of open research issues about how such modelling should best be done. Most research systems so far have been largely hand-coded, inflexible representations of dialogue states, implemented as some form of finite state or other rule-based machine (e.g. TRINDI systems at the most theoretical end [Larsson and Traum, 2000]). These approaches have addressed robustness issues within spoken language dialogue systems by limiting the range of the options and vocabulary available to the user at any given stage in the dialogue. They have, by common agreement, failed to capture much of the flexibility and functionality inherent in human-human communication, and the resulting systems have far less than optimal conversational capability and are neither pleasant nor natural to use.

Meanwhile, many low-functionality systems have, however, been deployed in the market in domains such as train reservations. On the other hand, more flexible, conversationally plausible models of dialogue, such as those based on planning, [Allen et al., 1995] are knowledge rich, and require very large amounts of manual annotation to create. They model individual communication actions, which are dynamically linked together into plans to achieve communicative goals in a tradition of work on trains going back to Perrault and his students [Allen and Perrault, 1980]. This method has greater scope for reacting to user input and correcting problems as they occur. Both types of contemporary theoretical model above have long histories, but have never placed their emphasis on implementation or evaluation.

The only regular forum for the evaluation of dialogue systems has been the Loebner Competition [1990], whose value has not been universally accepted, but which has served the field by keeping the emphasis on evaluation and innovation. Some of the current authors were among the designers of the winning Loebner entry in 1997, and some principles from that system CONVERSE [Levy et al., 1997] have survived in what we propose here, namely: machine learning, knowledge of conversational strategy encoded in flexible script structures rather than full plans, and a mechanism for judging the relative balance of system and user initiatives.

The model we wish to present occupies a position between these two approaches: full planning systems and turned-based dialogue move engines. We contend that larger structures are necessary to represent the content and context provided by mini-domains or meta-dialogue processes (a term we shall explain) as opposed to modelling only turn taking. The traditional problems with our position are: how to obtain the data that such structures (which we shall call Dialogue Action Frames or DAFs) contain, and how to switch rapidly between them in practice, so as not to be stuck in a dialogue frame inappropriate to what a user has just said. We shall explain their functioning within an overall control structure that stacks DAFs, and show that we can leave a DAF in any dialogue state and return to it later if appropriate, so that there is no loss of flexibility, and we can retain the benefits of larger scale dialogue structure. For now, DAFs are hand-coded but we shall show later in the paper how we are seeking to learn them from annotated dialogue corpora. In so doing, we hope to acquire those elements of human-human communication which may make a system more conversationally plausible.

The second major area that remains unsettled in dialogue modelling is the degree to which its modules can be based directly on abstractions from data (abstractions usually obtained by some form of Machine Learning) as significant parts of NLP have been over the last fifteen years. We shall describe a system for learning the assignment of dialogue acts (DAs) and semantic content directly from corpora, while noting the following difficulty: in the five years since Samuel et al. [1998] first demonstrated such a technique based on Transformation-Based Learning (TBL, [Brill, 1995]) the figures obtained [Stolcke et al., 2000] have remained obstinately in the area of 65%+ and not risen towards the Nineties as has been the case in other, perhaps less complex, areas of linguistic information processing, such as part-of-speech tagging.

In the model that follows, we hypothesise that the information content of DAs may be such that some natural limit has appeared to their resolution by the kinds of n-gram-based corpus analysis used so far, and that the current impasse, if it is one, can only be solved by realising that DA training is inherently low quality and that higher level dialogue structures in the DM will be needed to refine the input DAs, that is, by using the inferential information in DAFs, along with access to the domain model. This hypothesis, if true, explains the lack of progress with a purely data-driven research in this area and offers a concrete hybrid model, to employ an overused word in the area of NLP and ML. This process could be seen as one of the correction or reassignment of DA tags to input utterances in a DM, where a higher level structure will be able to chose from some (possibly ordered) list of alternative DA assignments as selected by our initial process.

2. Modality Independent Dialogue Management

Any survey of this field might suggest that we may now be in something of the same position as the field of Information Extraction (IE) when Jerry Hobbs [1993] wrote his brief note on a generic IE system, on the assumption that all functioning IE systems contained roughly the same modules doing the same tasks, and that the claimed differences were largely matters of advertising, plus dubious claims to special theories with trademarked names. However, and as we noted earlier, we may be in a worse position with dialogue systems because, unlike IE, there is substantial disagreement on the core control structure of a dialogue system, and little or no benchmarked performance with which to decide which modules and theoretical features aid robust dialogue performance. The lack of an established evaluation methodology is, by common consent, one of the main features holding back robust dialogue development methodology.

We cannot even appeal to some accepted progression of systems towards an agreed level of maturity, as one can in some areas of NLP like IE: even very primitive dialogue systems from long ago contain features which many would associate with very sophisticated systems: Carbonell's POLITICS [Carbonell et al., 1983] seems to be just a series of questions and answers to a complex knowledge base, but it is clear that he considers the system to deploy coded forms of goals, beliefs and plans, which one might take as a sufficient property for being in a more developed class of systems.

Even PARRY [Colby, 1971] the most developed and robust of the early dialogue systems, but based on no more than fast pattern matching, very clearly had the goal of informing the user of certain things and, even though it had no explicit representation of goals and beliefs, it did have a primitive but explicit model of the user.

2.1 Initial Design Considerations

The development of our Dialogue Management strategies has occurred largely within the COMIC (Conversational Multimodal Interaction with Computers)[1] project whose object is to build a cooperative multi-modal dialogue system which aids the user in the complex task of designing a bathroom, and a system to be deployed in a showroom scenario. A central part of this system is the Dialogue and Action Manager (DAM).

We assumed that a plausible DAM system must be able to have at least the following functionalities:

(a) determine the form of response appropriately, to dialogue turn pairs, where appropriately means in both pragmatic (i.e. dialogue act functional) and semantic terms (i.e. give correct answers to questions, if known).

(b) have some form of representation of a whole dialogue, which means not only opening and closing it appropriately, but knowing when a topic has been exhausted, and also how and when to return to it, if necessary, even though exhausted from the system's point of view.

(c) have appropriate access to a data base if there is to be question-answering on the basis of stored (usually application dependent) knowledge.

(d) have appropriate access to a database that can be populated if information is to be elicited from the user as part of a basic task.

(e) have some form of reasoning, belief/goal/intention representation, user modelling and planning sufficient to perform these tasks, though this need not imply any particular form of representation or mechanism for implementing these functionalities.

(f) have some general and accessible notion of where in the whole dialogue and task performance the system is at any moment.

The key problems for dialogue system performance, and therefore reasons for failure, are:

(i) the inability of a dialogue system to find the relevant structure/frame that encapsulates what is known to the system about the subject under discussion and to use this to switch topics when the user dictates that. This is the main

[1] see http://www.hcrc.ed.ac.uk/comic/

form of what we shall call the frame detection problem in dialogue management, one normally addressed by some level of overlap of terms in the input with indexes attached to particular task frames in the current application.

(ii) another problem for all dialogue systems is recovery from not knowing how to continue in a given dialogue state, and quite different strategies are out there in the field: e.g. the Rochester-style strategy [Allen and Perrault, 1980] of the system taking a definite, and possibly wrong, line with the user, relying on robust measures for revision and recovery if wrong, as opposed to a hesitant (and potentially irritating) system that seeks constant confirmation from the user before deciding on any action. We shall also opt for the former strategy, and hope for sufficiently robust recovery, while building in implicit confirmations for the user wherever appropriate.

We anticipate a core dialogue engine that is both a simple and perspicuous virtual machine (and not a lot of data/links and functionalities under no clear control) and which can capture (given good data structures) the right compromise between push (user initiative) and pull (system initiative) that any robust system must have. Our DAM sketch below, now implemented and integrated into the COMIC project, is intended to capture this combination of perspicuity (for understanding the system and allowing data structures to be written for it) and compromise between the two opposed dialogue initiative directions.

2.2 Choosing a Level of Structure

There is as yet no consensus as to whether a DAM should be expressed simply as a finite-state automaton, a well understood and easy to implement representation, or utilise more complex, knowledge-based approaches such as the planning mechanism employed by systems such as TRAINS [Allen et al., 1995].

The argument between these two views, at bottom, is about how much stereotypy one expects in a dialogue and which is to say, is it how much is it worth collecting all rules relevant to a subtopic together, within some larger structure or partition? Stereotypy in dialogue is closely connected to the notion of system-initiative or top-down control, which is strongest in “form-filling" systems and weakest in chatbots. If there is little stereotypy in dialogue turn ordering, then any larger frame-like structure risks being over-repetitious, since all possibilities must be present at many nodes. If a system must always be ready to change topic in any state, it can be argued, then what is the purpose of being in a higher level structure that one may have to leave? The answer to that it is possible to be always ready to change topic but to continue on if change is not forced: As with all frame-like structures since the beginning of AI, they express no more than defaults or preferences.

The same opposition was present in early AI planning theory between rule-driven planners and systems like SRI's STRIPS that pioneered more structural objects consisting of expected default actions [Fikes and Nilsson, 1971].

The WITAS system [Lemon et al., 2001] was initially, at least, based on networks of ATN (Augmented Transition Network) structures, stacked on one of two stacks. In the DAM described below we also opt for an ATN-like system which has as its application mechanism a single stack (with one slight modification) of DAF's (Dialogue Action Frames) and suggest that the WITAS argument for abandoning an ATN-type approach (namely, that structure was lost when a net is popped) is easily overcome. We envisage DAFs of radically different sizes and types: complex ones for large scale information eliciting tasks, and small ones for dialogue control functions such as seeking to reinstate a topic.

Our argument will be that the simplicity and perspicuity of this (well understood and easily written and programmed) virtual machine (at least in its standard form) has benefits that outweigh its disadvantages, and in particular the ability to leave and return to a topic in a natural and straightforward way. As we shall see below, this is a complex issue, and the need to return to unpopped syntactic ATN networks, so as to ensure completeness of parsing, is quite different in motivation from that of returning to an interrupted topic in dialogue processing. In syntactic parsing one must so return, but in dialogue one can sometimes return in a way that is pragmatically inappropriate and we shall deal with that below, and seek new forms of dialogue constraint and generalization.

2.3 DAFs: A Modest DAM Proposal

We propose a single pop-push stack architecture that loads structures of radically differing complexities but whose overall forms are DAFs. The algorithm to operate such a stack is reasonably well understood, though we will suggest below one amendment to the classical algorithm, so as to deal with a dialogue revision problem that cannot be dealt with by structure nesting.

The general argument for such a structure is its combination of power, simplicity and perspicuity. Its key language-relevant feature (known back to the time of Woods [1970] in syntactic parsing) is the fact that structures can be pushed down to any level and re-entered via suspended execution, which allows nesting of topics as well as features like barge-in and revision with a smooth and clear return to unfinished materials and topics. This is so well known that it has entered the everyday language of computer folk as “stack that topic for a moment". Although, in recursive syntax, incomplete parsing structures must be returned to and completed, in dialogue one could argue that not all incomplete structures should be re-entered for completion since it is unnatural to return to every suspended topic no matter how long suspended, unless, that is, the suspended structure contains information that must be elicited from the user. One experimental question here will be whether there are constraints on such re-entry to suspended networks, analogous to the semantic networks in Grosz's [1977] dialogue systems and the absolute constraints she proposed on long range reference back to open topics.

There will be DAFs corresponding to each of the system-driven sub-tasks (e.g. filling of the forms the bathroom salesman aims to end up with at the end of a client session) which are for eliciting information and whose commands write directly to the output database. There will also be DAFs for standard Greetings and Farewells, and for complex dialogue control tasks like revisions and responses to conversational breakdowns. A higher granularity of DAFs will express simple dialogue act pairs (such as QA) which can be pushed at any time (from user initiative) and will be exhausted (and popped) after an SQL query to the COMIC database.

The stack is preloaded with a (default) ordered set of system initiative DAFs, with Greeting at the top, Farewell at the bottom and such that the dialogue ends with maximum success when these and all the intermediate information eliciting DAFs for this task have been popped. This would be the simplest case of a maximally cooperative user with no initiative whatever; he may be rare but must be catered for if he exists.

An obvious problem arises here, noted in earlier discussion, which may require that we adapt this overall DAM control structure:

If the user proposes an information eliciting task before the system does (e.g. in a bathroom world, we suppose the client wants to discuss tile-colour-choice before that DAF is reached in the stack) then that structure must immediately be pushed onto the stack and executed till popped, but obviously its copy lower in the stack must not be executed again when it reaches the top later on. The integrity of the stack algorithm needs to be violated only to the extent that any task-driven structure at the top of the stack is only executed from its initial state if the relevant part of the database is incomplete.

However, a closely related, issue (and one that caused the WITAS researchers to change their DAM structure, wrongly in our view) is the situation where a user-initiative forces the revision/reopening of a major topic already popped from the stack; e.g. in a bathroom world, the user has chosen pink tiles but later, and at her own initiative, decides she would prefer blue and initiates the topic again. This causes our proposal no problems: the tile-colour-choice DAF structure is pushed again (empty and uninstantiated) but with an entry subnetwork (no problem for DAFs) that can check the data-base, see it is complete, and begin the subdialogue in a way that responses show the system knows a revision is being requested. It seems clear to us that a simple stack architecture is proof against arguments based on the need to revisit popped structures, provided the system can distinguish this case (as user initiative) from the last (a complete structure revisited by system initiative).

A similar device will be needed when a partly executed DAF on the stack is re-entered after an interval; a situation formally analogous to a very long syntactic dependency or long range co-reference. In such cases, a user should be asked whether he wishes to continue the suspended network (to completion). It will be an experimental question later, when data has been generated, whether there are constraints on access to incomplete DAFs that will allow them to be dumped from the top of the stock unexecuted, provided they contain no unfilled requests for bathroom choices.

What has not been touched upon here is the provision, outside the main stack and content-structures, of DAM modules that express models of the user's goals/beliefs/intentions and which reason over these. We shall postpone this discussion as inessential for getting a DAM started and able to generate dialogue data for later learning and modification (see further below), provided what we ultimately propose can transition from simpler to more complex structures and functions without radical redesign. To deploy such capacity for bathroom advice would require an implausible scenario where the advisor has to deal with e.g. a client couple, possibly interviewed separately so that the system has to construct a couple's views of each other's wishes.

We expect later to build into the DAM an explicit representation of plan tasks, and this will give no problem to a DAF since recursive networks can be, and often have been, a standard representation of plans, which makes it odd that some redesigners of DAM's have argued against using ATNs as DAM models, wrongly identifying them with low-level dialogue grammars, rather than, as they are, structures (ATNs) more general than those for standard plans (RTNs). As a model of goals intentions and beliefs of the dialogue participants, we expect to use our procedural ViewGen [Ballim and Wilks, 1991] model.

3. Learning to Annotate Utterances

In the second part of this paper, we will focus on some experiments on modelling aspects of dialogue directly from data. In the joint EU-, US- funded project AMITIES we are building automated service counters for telephone-based interaction, by using large amounts of recorded human-human data.

Initially, we report on some experiments on learning the analysis part of the dialogue engine; that is, that part which converts utterances to dialogue act and semantic units. Subsequently, we will outline the next stage of research, that of learning dialogue structure from a corpus.

3.1 Machine-Learning for Dialogue Act Tagging

There has been an increasing interest in using machine learning techniques to solve problems in spoken dialogue. One thread in this work has addressed dialogue act modelling, i.e. the task of assigning an appropriate dialogue act tag to each utterance in a dialogue. It is only recently, with the availability of annotated dialogue corpora, that empirical research in this area has become possible.

Two key annotated corpora, which have formed the basis for work on dialogue act modelling, are of particular relevance here: first, the VerbMobil corpus [Reithinger and Klesen, 1997], which was created within the project developing the VerbMobil speech-to-speech translation system, and secondly, the Switchboard corpus [Jurafsky et al., 1998]. Of the two, Switchboard has generally been considered to present a more difficult problem for accurate dialogue act modelling, partly because it has been annotated using a total of 42 distinct dialogue acts, in contrast to the 18 used in the VerbMobil corpus, and a larger set makes consistent judgements harder. In addition, Switchboard consists of unstructured non-directed conversations, which contrast with the highly goal-directed dialogues of the VerbMobil corpus.

One approach that has been tried for dialogue act tagging is the use of n-gram language modelling, exploiting ideas drawn directly from speech recognition. For example, Reithinger and Klesen [1997] have applied such an approach to the VerbMobil corpus, which provides only a rather limited amount of training data, and report a tagging accuracy of 74.7%. [Stolcke et al., 2000] apply a somewhat more complicated n-gram method to the Switchboard corpus (which employs both n-gram language models of individual utterances, and n-gram models over dialogue act sequences) and achieve a tagging accuracy of 71% on word transcripts, drawing on the full 205k utterances of the data. Of this, 198k utterances were used for training, with a 4k utterance test set. These performance differences can be seen to reflect the differential difficulty of tagging for the two corpora.

A second approach by Samuel et al. [1998], uses transformation-based learning over a number of utterance features, including utterance length, speaker turn and the dialogue act tags of adjacent utterances. They achieved an average score of 75.12% tagging accuracy over the VerbMobil corpus. A significant aspect of this work, that is of particular relevance here, has addressed the automatic identification of word sequences that would form dialogue act cues. A number of statistical criteria are applied to identify potentially useful n-grams which are then supplied to the transformation-based learning method to be treated as ‘features’.

3.2 Creating a Naive Classifier

As just noted, Samuel et al. [1998] investigated methods for identifying word n-grams that might serve as useful dialogue act cues for use as features in transformation-based learning. We decided to investigate how well n-grams could perform when used directly for dialogue act classification, i.e. with an utterance being classified solely from the individual cue phrases it contains. Two questions immediately arise. Firstly, which n-grams should be accepted as cue phrases for which dialogue acts, and secondly, which dialogue act tag should be assigned when an utterance contains several cues phrases that are indicative of different dialogue act classes. In the current work, we have answered both of these questions principally in terms of *predictivity*, i.e. the extent to which the presence of a certain n-gram in an utterance is predictive of it having a certain dialogue act category, which for an n-gram n and dialogue act category d corresponds to the conditional probability: $P(d|n)$.

A set of n-gram cue phrases was derived from the training data by collecting all n-grams of length 1–4, and counting their occurrences in the utterances of each dialogue act category and in total. These counts allow us to compute the above conditional probability for each n-gram and dialogue act. This set of n-grams is then reduced by applying thresholds of predictivity and occurrence, i.e. eliminating any n-gram whose maximal predictivity for any dialogue act falls below some minimum requirement, or whose maximal number of occurrences with any category falls below a threshold value. The n-grams that remain are used as cue phrases. The threshold values that were used in our experiments were arrived at empirically.

To classify an utterance, we identify all the cue phrases it contains, and determine which has the highest predictivity of some dialogue act category, and then that category is assigned. If multiple cue phrases share the same maximal predictivity, but predict different categories, one category is assigned arbitrarily. If no cue phrases are present, then a default tag is assigned, corresponding to the most frequent tag within the training corpus.

3.3 Corpus and Data Sets

For our experiments, we used the Switchboard corpus, which consists of 1,155 annotated conversations, comprising around 205k utterances. The dialogue act types for this set can be seen in [Jurafsky et al., 1997]. From this source, we derived two alternative datasets. Firstly, we extracted 50k utterances, and divided this into 10 subsets as a basis for 10-fold cross-validation (i.e. giving 45k/5k utterance set sizes for training/testing). This volume was selected as being large enough to give an idea of how well methods could perform where a good volume of data was available, but not too large to prohibit experiments with 10-fold cross-validation from excessive training times. The second data set was selected for loose comparability with the work of Samuel, Carberry and Vijay-Shanker on the VerbMobil corpus, who used training and test sets of around 3k and 300 utterances. Accordingly, we extracted 3300 utterances from Switchboard, and divided this for 10-fold cross-validation.

3.4 Experiments

We evaluated the naive tagging approach using the two data sets just described, in both cases using a predictivity threshold of 0.25 and an occurrence threshold of 8 to determine the set of cue phrases. Applied to the smaller data set, the approach yields a tagging accuracy of 51.8%, which compares against a baseline accuracy of 36.5% from applying the most frequently occurring tag in the Switchboard data set (which is **sd** — statement). Applied to the larger data set, the approach yields a tagging accuracy of 54.5%, which compares to 33.4% from using the most frequent tag.

More recent experiments suggest that we can dramatically improve this score. We introduced start and end tags to every utterance (to capture phrases which serve as cues when specifically in these locations), and trained specific utterance length models. For example, we trained three models — one for utterances of length 1, another for length between 2 and 4 words, and another for length 5 and above. Combining these features, we obtained a maximal score for our naive tagger of 63% over the larger data set. Given that Stolke et al. achieve a total tagging accuracy of around 70% on Switchboard data, we observe that our approach goes a long way to reproducing the benefits of that approach, but using only a fraction of the data, and using a much simpler model (i.e. individual dialogue act cues, rather than a complete n-gram language model).

3.5 Experiments with Transformation-Based Learning

Transformation-based learning (TBL) was first applied to dialogue act modelling by Samuel, Carberry and Vijay-Shanker. They achieved overall scores of 75.12% tagging accuracy, using the VerbMobil corpus. As previously noted, an aspect of their work addressed the identification of potential cue phrases, for use as features during transformation based learning, i.e. so transformation rules can be learned which require the presence of a given cue as a context condition for the rule firing. In that work, the initial tagging state of the training data from which TBL learning would begin was produced by assigning every utterance a default tag corresponding to the most frequent tag over the entire corpus.

In our experiments, we wanted to investigate two issues. Firstly, whether a more effective dialogue act tagging approach could be produced by using our naive n-gram classifier as a pre-tagger generating the initial tagging state over which a TBL tagger could be trained. It seems plausible that the increased accuracy of the initial tag state produced by the naive classifier, as compared to assigning just the most frequent tag, might provide a basis for more effective subsequent training. Secondly, we wanted to assess the impact of using larger volumes of training data with a transformation based approach, given that Samuel et al.'s results are based on a quite small data set from the VerbMobil corpus.

For an implementation of transformation based learning, we used the freely available μ-TBL system of Lager [1999]. The current distribution of μ-TBL provides an example system for dialogue act modelling, including a simple set of templates, which is developed with reference to the Samuel et al. work, and applied to the MapTask domain [Lager and Zinovjeva, 1999]. We have used this set of templates for all our experiments. We should note that the Lager and Samuel et al. template sets differ considerably, e.g. Samuel et al. use thousands of templates (together with a Monte Carlo variant of the training algorithm), whilst the μ-TBL templates are much fewer in number, may refer only to the dialogue act tags of preceding utterances (i.e. not both left and right), and may refer to any unigram or bigram appearing within utterances as part of a context condition, i.e. they are not provided with a fixed set of dialogue act cues to consider.

Our best results over the larger data set from the SwitchBoard corpus are around 66%, applying TBL to an initial data state produced by the naive classifier. Interestingly, our results indicate the naive classifier achieves most of the gain, with TBL consistently adding only 2 or 3%. In further work, we intend to apply other machine learning algorithms to the results of pre-tagging the data using a naive classifier.

4. Future work: Data Driven Dialogue Discovery

Using the same corpus as above, to what extent could we discover the structure of DAFs, and their bounds from segmentations of the corpus, from annotated corpora? We are currently exploring the possibility of deriving the DAF structures themselves by taking a dialogue-act annotated corpus and then annotating it further with an information extraction engine [Larsson and Traum, 2000] that seeks and semantically tags major entities and their verbal relations in the corpus, which is to say, derives a surface-level semantic structure for it. One function of this additional semantic tagging will be to add features for the DA tagging methods already described, in the hope of improving our earlier figures by adding semantic features to the learning process.

The other, and more original possibility, is that of seeking repeated sequences of DA+ semantic triple type (verb, plus agent and object types) and endeavouring to optimise the "packing" of such sequences to fill as much as possible of a dialogue by using some algorithm such as Minimum Description Length, so as to produce reusable, stereotypical, dialogue segments. We anticipate combining this with some corpus segmentation by topic alone, following Hearst's tiling technique [Hearst, 1993].

Given any success at learning the segmentation of dialogue data, we expect to use some form of the Reinforcement Learning approach of Walker [1990] to optimise the DAF's themselves.

5. Discussion

The work in the last section is at a very preliminary stage, but will we hope form part of the general strategy of this paper which is to derive a set of weak, general, learning methods which will be strong in combination (in the spirit of Newell [1990]). This means the effect of the combination of a top down DAM using DAFs learned from corpus data, with a DA+ semantics parser learned from the same data. It is the interaction of these bottom-up and top-down strategies in dialogue understanding (where the former is taken to include the ambiguities derived from the acoustic model) that we seek to investigate. This can perfectly well be seen as part of the program for a full dialogue model laid out in [Young, 2000] in which he envisaged a dialogue model as one where different parts are separately observed and framed before being combined.

We have shown that a simple dialogue act tagger can be created that uses just n-gram cues for classification. This naive tagger performs modestly, but still surprisingly well given its simplicity. More significantly, we have shown that a naive n-gram classifier can be used to pre-tag the input to transformation based learning, which removes the need for a vast number of n-gram features to be used in the learning algorithm. One of the prime motivators for using TBL was its resilience to such a high number of features, so by removing the need to incorporate them, we are hopeful that we can use a range of ML approaches for this task.

In regard to the naive n-gram classifier, we have described how the training of the classifier involves pruning the n-gram by applying thresholds for predictivity and absolute occurrence. These thresholds, which are empirically determined, are applied globally, and will have a greater impact in eliminating possible n-gram cues for the less frequently occurring dialogue act types. We aim to investigate the result of using local thresholds for each dialogue act type, in an attempt to keep a adequate n-gram representation of all dialogue acts types, including the less frequently occurring ones.

Finally, we aim to apply these techniques to a new corpus collected for the AMITIES project, consisting of human-human conversations recorded in the call centre domain [Hardy et al., 2002]. We hope that the techniques outlined here will prove a useful first step in creating automatic service counters for call centre applications.

Although we have described a dialogue analysis approach in one project (AMITIES) and a DAM in another (COMIC), this is merely a side effect of funding strategy and we expect to bring the two together in a single system, along with an appropriate generation component and ASR front end.

With the generation of more data from our already functioning DAM, we hope to derive constraints on stack access and the reopening of all unpopped DAFs. This, if successful, will be an important demonstration of the different functioning of DAFs as contrasted with the use of ATNs in syntactic analysis (e.g. [Woods, 1970]) where non-determinism requires both back tracking and the exhaustion of all unpopped ATN's for completeness and the generation of all valid parsings of a sentence. It should be noted that this is not at all the case here: there is no provision for backtracking in DAFs and we expect to derive strong constraints such that not all unpopped DAFs will be reactivated. Analogous to the early dialogue findings of Grosz [1977] and Reichmann [1985] we expect some unpopped DAFs are not reopenable after substantial dialogue delay, just as they showed that dialogue segments and topics were closed off and became eventually inaccessible. Also, the ATN interpreter, unlike it's use in syntactic processing, is deterministic, since, in every state, there will be a best match between some arc condition and incoming representations. In this paper, we have discussed aspects of our approach to dialogue analysis/fusion and control, but have not touched at all on generation/fission and the role of knowledge rich items, such as belief and planning structures, in that phase. All this we shall leave to a later paper.

6. Acknowledgements

This paper is based on work supported in part by the European Commission under the 5th Framework IST/HLT Program (consortia AMITIES and COMIC), and by the U.S. Defense Advanced Research Projects Agency.

References

Allen, J. F. and Perrault, C. R. (1980). Analyzing Intentions in Utterances. *Journal of Artificial Intelligence*, 15(3):143–178.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Miller, B. W., Poesio, M., and Traum, D. R. (1995). The TRAINS Project: A Case Study in Building a Conversational Planning Agent. *Journal of Experimental and Theoretical AI (JETAI)*, 7:7–48.

Ballim, A. and Wilks, Y. (1991). *Artificial Believers*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Brill, E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, pages 234–246.

Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1983). An Overview of Machine Learning. In Carbonell, J. G., Michalski, R. S., and Mitchell, T. M., editors, *Machine Learning: An Artificial Intelligence Approach*, pages 168–185. Palo Alto, CA: Tioga Pub Co.

Colby, K. M. (1971). Artificial Paranoia. *Journal of Artificial Intelligence*, 2:76–89.

Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the Second International Joint Conference on Artificial Intelligence (IJCAI-7)*, volume 1, pages 111–119.

Grosz, B. (1977). The Representation and Use of Focus in Understanding Dialogs. In Grosz, B., Jones, K. S., and Webber, B. L., editors, *Readings in Natural Language Processing*, pages 56–67. Morgan Kaufmann Publishers Inc.

Hardy, H., Baker, K., Devillers, L., Lamel, L., Rosset, S., Strzalkowski, T., Ursu, C., and Webb, N. (2002). Multi-Layered Dialogue Annotation for Automated Multilingual Customer Service. In *Proceedings of the ISLE Workshop on Dialogue Tagging for Multimodal Human Computer Interaction*, pages 90–99, Edinburgh, UK.

Hearst, M. A. (1993). TextTiling: A Quantitative Approach to Discourse Segmentation. Technical Report UCB:S2K-93-24, Berkeley, CA.

Hobbs, J.R. (1993). The Generic Information Extraction System. In *Proceedings of the Fifth Message Understanding Conference (MUC-5), Journal of Artificial Intelligence*, pages 87–91. Morgan Kaufman Publishers.

Jurafsky, D., Bates, R., Coccaro, N., Martin, R., Meeter, M., Ries, K., Shriberg, E., Stolcke, A., Taylor, P., and van Ess-Dykema, C. (1998). *Switchboard Discourse Language Modeling Project Report Research Note 30*. Center for Speech and Language Processing, Johns Hopkins University, Baltimore, MD.

Jurafsky, D., Shriberg, E., and Biasca, D. (1997). Switchboard-DAMSL Labeling Project Coder's Manual. Technical Report 97-02, University of Colorado, Institute of Cognitive Science, Boulder, CO.

Lager, T. (1999). The μ-TBL System: Logic Programming Tools for Transformation-Based Learning. In *Proceedings of the Third International Workshop on Computational Natural Language Learning*, pages 190–201, Bergen, Norway.

Lager, T. and Zinovjeva, N. (1999). Training a Dialogue Act Tagger with the μ-TBL System. In *Proceedings of the Third Swedish Symposium on Mul-*

timodal Communication, pages 66–87. Linköping University Natural Language Processing Laboratory.

Larsson, S. and Traum, D. (2000). Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. *Journal of Natural Language Engineering*, 6:267–278.

Lemon, O., Bracy, A., Gruenstein, A. R., and Peters, S. (2001). The Witas Multi-Modal Dialogue System I. In *Proceedings of European Conference on Speech Communication and Technology (Eurospeech)*, pages 1559–1562, Aalborg, Denmark.

Levy, D., Catizone, R., Battacharia, B., Krotov, A., and Wilks, Y. (1997). CONVERSE: A Conversational Companion. In *Proceedings of the First International Workshop on Human-Computer Conversation*, pages 27–34, Bellagio, Italy.

Loebner Competition (1990).
http://www.loebner.net/Prizef/loebner-prize.html.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Reichmann, R. (1985). *Getting Computers to Talk Like You and Me*. Cambridge, MA: MIT Press.

Reithinger, N. and Klesen, M. (1997). Dialogue Act Classification Using Language Models. In *Proceedings of European Conference on Speech Communication and Technology (Eurospeech)*, pages 2235–2238, Rhodes, Greece.

Samuel, K., Carberry, S., and Vijay-Shanker, K. (1998). Dialogue Act Tagging with Transformation-Based Learning. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, volume 2, pages 1150–1156, Montreal.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., van Ess-Dykema, C., and Meteer, M. (2000). Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 26(3):339–373.

Walker, M. A. (1990). An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email. *Journal of Artificial Intelligence Research*, 12:387–416.

Woods, W. A. (1970). Transition Network Grammars for Natural Language Analysis. *Communications of the ACM*, 13(10):591–606.

Young, S. J. (2000). Probabilistic Methods in Spoken Dialogue Systems. *Philosophical Transactions of the Royal Society (Series A)*, 358(1769):1389–1402.