Human dialogue modelling using annotated corpora

Yorick Wilks, Nick Webb, Andrea Setzer, Mark Hepple, 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: first we describe a Dialogue Manager which uses a representation of stereotypical dialogue patterns that we call Dialogue Action Frames and which, we believe, generate strong and novel constraints on later access to incomplete dialogue topics. Secondly, 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.

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. 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. However, many of these low-functionality systems have 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. This method has greater scope for reacting to user input and correcting problems as they occur, but has never placed emphasis on either implementation or evaluation.

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 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 ultimately 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.

A 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.

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 ngram-based corpus analysis used so far, and that the current impasse, if it is one, can only be solved by realising 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. 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

The development of our Dialogue Management strategies has occured 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).

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

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

as the planning mechanism employed by systems such as TRAINS.

The argument between these two views, at bottom, is about how much stereotopy 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? Stereotopy 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 stereotopy 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 WITAS system (Lemon et al., 2001) was initially 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.

2.1. DAFs: A proposed model for DAM

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. 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. There will be DAFs corresponding to each of the system-driven sub-tasks 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) 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 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.

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).

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[2] 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.

Two key annotated corpora, which have formed the basis for work on dialogue act modelling are of particular relevance here: first, the VERBMOBIL corpus , 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'.

3.1. Creating a naive classifier

As 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.

3.2. Corpus, data sets and experiments

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

[2] See `http://www.dcs.shef.ac.uk/nlp/amities/`

3300 utterances from SWITCHBOARD, and divided this for 10-fold cross-validation. We evaluated the naive tagging approach using these two data sets, 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.

Further 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 models sensitive to utterance length. 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 cross validated score for our naive tagger of 61.92% over the larger, 50k data set (with a high of 65.03%). 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).

3.3. N-Best Dialogue Act Classification

Our most recent experiment shows interesting promise. We built a classifier using the 45k utterance training set, and tested it on the 5k utterance test set. However, rather than attempting to find the single best match from the classifier, we tagged each utterance with the top 5 possible utterances, as indicated by the classifier on the basis of the predictivity of the n-grams the utterance contained. On a cross-validation of the corpus, we calculated that 86.74% of the time, the correct dialogue act was contained in the 5-best output of the classifier. In order to create some baseline measure, this experiment was repeated using the top 5 n-grams occurring by frequency in the SWITCHBOARD corpus. The tagging accuracy of this experiment was 71.09%.

This would appear to confirm our belief in a limit on the potetial resoution to this classification problem using ngram-based corpus analysis. However, we can an ordered list of possible alternatives to some higher level structure (the DM), where in this case the complexity of the choice is reduced from some 200 to 5.

3.4. Future Work

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 tranformation 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 resiliance to such a high number of features, so by removing the need to incorporate them, we are hopeful that we can use a wider range of machine learning 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.

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.

4. References

Allen, J. F., L. K. Schubert, G. Ferguson, P. Heeman, C. Hee Hwang, T. Kato, M. Light, N.G. Martin, B.W. Miller, M. Poesio, and D.R. Traum, 1995. The trains project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:07–48.

Hardy, H., K. Baker, L. Devillers, L. Lamel, S. Rosset, T. Strzalkowski, C. Ursu, and N. Webb, 2002. Multi-layered dialogue annotation for automated multilingual customer service. In *Proceedings of the ISLE workshop on Dialogue Tagging for Multimodal Human Computer Interaction*. Edinburgh.

Jurafsky, D., R. Bates, N. Coccaro, R. Martin, M. Meeter, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, and C. Van Ess-Dykema, 1998. Switchboard discourse language modeling. Technical Report Project Report Research Note 30, Center for Speech and Language Processing, Hopkins University.

Lemon, O., A. Bracy, A.R. Gruenstein, and S. Peters, 2001. The witas multi-modal dialogue system i. In *Proceedings of the Seventh European Conference on Speech and Communication Technology (EuroSpeech)*. Aalborg, Denmark.

Reithinger, N. and M. Klesen, 1997. Dialogue act classification using language models. In *Proceedings of the Fifth European Conference on Speech and Communication Technology (EuroSpeech)*. Rhodes, Greece.

Samuel, K., S. Carberry, and K. Vijay-Shanker, 1998. Dialogue act tagging with transformation-based learning. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th Annual International Conference on Computational Linguistics*, volume 2. Montreal, Canada.

Stolcke, A., K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer, 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Woods, W. A., 1970. Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606.