Sequential Learning for Dialog Act Classification in Tutorial Dialog

Mahesh Joshi
10-701 Machine Learning (Fall 2006)
Final Project Report
maheshj@cs.cmu.edu

Abstract

Dialog act classification or tagging is the task of assigning labels such as "question", "assertion", "positive feedback" and "negative feedback" to the turns in a dialog. In this project, we study the dialog act classification task as applied to human-human tutoring dialogs in the domain of thermodynamics. We initially establish a baseline by posing the task as a classification problem and applying three supervised machine learning methods. We then view the task as a sequential labeling problem, to make use of previous dialog act labels for predicting future labels and compare the performance of sequential labeling algorithms with the baseline accuracy. The best performing sequential algorithm shows 5.59% improvement in macro-averaged accuracy over the best baseline algorithm.

1 Introduction

Dialog Act (DA) classification or tagging is the task of assigning labels to each of the turns of the speakers (or in general, agents in the case of textual communication such as chat) in a dialog. The set of labels to be assigned is decided in advance and usually corresponds to some high level semantic concepts such as "question", "answer", "offer", "acceptance", which can be used to perform a shallow semantic analysis of the dialog. DA classification can be useful in tasks such as summarization of dialog, where for example one might be interested in identifying the key questions raised in a meeting (multi-party dialog) and the possible set of answers [1].

Tutoring domain dialogs share many characteristics of dialogs in general, such as short responses and informal non-expository language. Additionally, the DAs are customized to suit the specific domain. For example, DAs such as "positive response" and "negative response" are usually included in the set of DAs since they can help assess student learning.

2 Related Work

Dialog Act classification is a well-studied problem in the literature, particularly in the spoken dialog domain. Research has also been done in applying sequential learning algorithms such as Hidden Markov Models (HMMs) to the problem. Here we briefly discuss some of the related work and also previous work related to topic segmentation of the dialog corpus that we use for our experiments.

Surendran and Levow [2] have applied Support Vector Machines (SVMs) and HMMs to the problem of DA classification in the a corpus of two-speaker task-oriented dialogs involving map directions given by a "giver" to the "follower". This is in some respect similar to our dialog corpus where a human tutor is guiding the student towards the solution to a thermodynamics problem, except the goal in tutoring dialog is also to evaluate the student understanding, rather than simply providing directions. Since theirs is a spoken dialog corpus, they have made use of acoustic as well as textual features for the task and shown slight improvement in classification accuracy using sequential HMM labeling.

Galley et. al. [3] have applied Bayesian Networks to the problem of predicting agreement and disagreement in a multi-party dialog corpus of research team meetings at the International Computer Science Institute. They initially apply maximum entropy modeling to predict adjacency pairs in the dialog, which are essentially pairs of coherent correspondences between two speakers – such as a question and an answer. They then use these adjacency pairs to design dependencies among previous and current contributions in the dialog. Their results suggest improvement in classification of agreements and disagreements using knowledge of such dependencies.

Arguello and Rosé [4] have used the tutoring domain corpus that we use in our experiments to perform automatic topic segmentation of dialog, which involves predicting the start of a new topic or a topic-shift in the tutoring dialog. The manually annotated topics included general ones such as *greetings*, *initialization*, *general thermo concepts* as well as task-specific ones such as *sensitivity analysis* and *regeneration*. The features employed in their experiments were lexical features such as unigrams and bigrams, and syntactic features such as Part-of-Speech bigrams. They compared their system to several state-of-the-art systems for segmentation of expository text and demonstrated significant improvement.

3 Data

The data (thermo-study corpus) is a corpus of student-tutor (human-human) dialogs in which the students worked on an optimization problem for the thermodynamics domain, via chat software. This is a locally collected corpus consisting of 22 student-tutor dialogs. Each dialog is composed of multiple turns from the student and the tutor, for a total of 4794 turns. The corpus has been annotated (mostly manually) to include six features for each turn. Briefly, three of them are as follows (with their possible values in parentheses):

- **Agent**: Automatically logged via chat software. (Tutor, Student)
- **Action**: The dialog act, classified as open response question (1), closed response question (2), check for understanding by tutor (3), assertion (4), negative response (5), positive response (6), direction or command (7), meta statements (8) and other (0).
- **Exchange**: A sequence identifier given to a group of turns in the dialog that form one dialog exchange (an Initiative-Response-Feedback segment).

Further details about the other features and the coding scheme used for the annotation can be found in the next section and also online[1].

4 Method

In this project, as in related literature, we initially treat the DA tagging as a supervised classification problem and then as a sequential labeling problem. The motivation for applying sequential learning to this particular tutoring dialog corpus is based on an empirical

[1] http://www.cs.cmu.edu/∼maheshj/Regan-CodingScheme.doc

Table 1: Transition table showing the number of times the dialog acts listed along the columns follow the dialog acts listed along the rows, in the thermo-study tutoring dialog corpus of 22 dialogs, 4794 turns. Bold values indicate the maximum in each row.

	0	1	2	3	4	5	6	7	8
0	**258**	3	24	6	20	3	13	23	5
1	1	17	21	0	**62**	5	1	8	2
2	9	7	106	4	**220**	22	112	85	8
3	3	3	8	1	22	4	**52**	4	1
4	22	34	148	58	**700**	22	297	162	16
5	7	0	16	1	**36**	16	15	29	4
6	16	31	121	12	229	19	148	**279**	24
7	31	20	118	15	159	30	221	**472**	11
8	3	2	11	1	13	3	**21**	15	**21**

analysis of DA transitions from one label to another. Table 1 lists the number of times the DAs listed along the columns follow the DAs listed along the rows, in our entire corpus of 4794 turns. The table shows that certain DAs are more likely to follow a given DA, than others. For example, negative response (5) is more likely to be followed by an assertion (4) or a direction/command (7) than others. This is quite intuitive since a negative response by the tutor is likely followed by some sort of informative assertion or directive statement by the tutor again. Based on this empirical analysis, we hypothesize that the use of predicted DA labels of previous turns will improve accuracy on our DA prediction task.

4.1 Features

We experiment with the following types of features:

Lexical (lex): We use unigrams (single words) and bigrams (two-word sequences) as binary features, while skipping stopwords such as articles and prepositions and stemming the words to their root morphological forms. Unigrams or bigrams that occur fewer than five times in the training set are rejected. We also employ a set of 19 punctuation features such as period, comma, semi-colon and colon. Additionally we include the normalized length of a turn as a numeric feature.

Syntactic (syn): We use Part-of-Speech bigrams as pseudo-syntactic features. No stoplisting or stemming is done for identifying Part-of-Speech tags, but the frequency cutoff criteria of five (as in the case of unigrams and bigrams) applies.

Meta (met): In addition to the automatically extracted set of features above, we also have additional meta-level features for each of the turns (most of which are manually annotated). These are as follows:

- **Agent**: This can take the value *T* for Tutor or *S* for Student. This is automatically tracked via the chat software.
- **Depth**: This is a binary feature indicating whether or not a turn is an explanation. This has been manually annotated.
- **Control**: This indicates the type of a turn in terms of the conversational Initiation-Response-Feedback theory by Sinclair and Coulthard [5]. It can take the following values: *D*–dialog initiation, *T*–task initiation, *R*–response, *F*–feedback, or the following combinations of two of the above four – *R/D*, *R/T*, *F/D* and *F/T*. This is a manually annotated feature.
- **Focus**: Indicates whether the speaker's contribution is self-oriented (S) or other-oriented (O). Manually annotated.
- **Exchange**: This is a binary feature derived from the exchange attribute described in the Data section above. It indicates whether or not a turn is the start of a new

exchange. The exchanges have been manually annotated.

4.2 Machine Learning Algorithms

For the initial baseline experiments involving supervised classification, we have experimented with the Decision Tree learner, the Naïve Bayes classifier and SVMs. The reason to choose these three algorithms is that all of them have been shown to work well on natural language processing tasks in previous literature. We have used the implementations of these algorithms from the MinorThird [6] package.

For sequential learning, we have made use of two of the sequential learning algorithm implementations available in MinorThird – Conditional Random Fields [7] and Collins' Perceptron Learner [8].

5 Experiments

The goal of our experiments is twofold – (1) to perform a feature engineering exercise for the task of DA classification by examining if use of features from previous turns improves accuracy (in other words, are the features of previous turns good proxies for predicted labels of previous turns, as in sequential learning?), and (2) to compare the accuracy of classification and sequential learning algorithms.

5.1 Baseline Experiments

We plan to make use of the exchange boundaries to limit the previous context in our sequential experiments. However, using manually annotated exchange boundaries introduces a bottleneck in the automation of this method. In order to try and avoid this, we performed experiments to automatically predict the exchange boundaries in our corpus. The two sets of features we have employed are: (1) lexical + syntactic, and (2) lexical + syntactic + meta (excluding the exchange feature).

For our evaluation methodology, we have employed a leave-one-dialog-out method in which each one of the 22 dialogs is employed as a test set while the remaining 21 are used as training set. To account for the variable size of our test sets, we report both macro-average accuracy and micro-average accuracy numbers. Macro-average accuracy is the accuracy obtained by averaging the 22 accuracy values obtained on the 22 test sets, while micro-averaged accuracy is calculated by counting the total number of correctly classified turns across all the 22 test sets and then dividing by the total number of turns (4794).

For dialog act classification, we have employed three sets of features: (1) lexical + syntactic, (2) lexical + syntactic + meta (excluding exchange) and (3) lexical + syntactic + meta (including exchange, manually annotated).

5.1.1 Exchange Prediction Results

Table 2 shows the results on exchange boundary prediction. The average majority class baseline for this task is 0.775 since 3717 turns out of the total 4794 are not exchange boundaries. So even if a classifier always predicts a turn as "not an exchange boundary", the micro-average accuracy will still be 0.775. The results are in agreement with those in [4], showing that automatic exchange boundary prediction in dialog is a fairly hard task. It can be observed however that the meta-features indeed seem to increase accuracy significantly above the majority baseline.

Table 2: Accuracy on exchange boundary prediction task.

	Classifier					
	Decision Tree		**Naïve Bayes**		**SVM**	
Feature Set	Macro	Micro	Macro	Micro	Macro	Micro
lex+syn	0.769	0.778	0.678	0.689	0.786	0.790
lex+syn+met	0.800	0.803	0.745	0.751	0.824	0.826

Table 3: Baseline accuracy on DA prediction task.

	Classifier					
	Decision Tree		**Naïve Bayes**		**SVM**	
Feature Set	Macro	Micro	Macro	Micro	Macro	Micro
lex+syn	0.610	0.627	0.675	0.691	0.644	0.659
lex+syn+met(no exch)	0.730	0.732	0.751	0.761	0.706	0.707
lex+syn+met(exch)	0.730	0.732	**0.752**	**0.763**	0.705	0.706

5.1.2 Baseline Results

Table 3 shows the results for DA classification. The majority baseline for DA classification is 0.305, with the DA=4 (assertion) being the most frequent class. The naïve Bayes classifier with the full feature set yielded the best accuracy values (bolded in Table 3). A formal statistical significance analysis has not been done for our results, but in general the accuracies seem significantly higher when using the "Meta" feature set (with or without exchange).

5.2 Features from Previous Turns

In addition to the three feature sets for DA classification mentioned in the baseline experiments section, for each of them we also generated three more feature sets that included the features from previous 1, 2 and 3 turns. At the implementation level, the features from the N^{th} previous turn were prefixed with "PREVN_" so as to distinguish them from the features of the current turn. For the "lexical + syntactic + meta" feature set with exchanges, if any of the previous N turns crossed an exchange boundary, then its features were not added to the current turn.

5.2.1 Previous Turn Features Results

Table 4 shows results of using the augmented feature sets. As can be observed, the accuracy using these augmented feature sets was worse than without using them, for most of the cases. Decision trees were fairly robust to the introduction of these new features and seemed to pick the right subset of features among the larger set. However, naïve Bayes classifier and SVMs suffered a consistent degradation in performance with the introduction of these augmented set of features. We therefore hypothesized that the features from previous turns were overall adding more noise than information and performed experiments with feature selection using information gain criterion, choosing the top 1000 features from each augmented set. The number 1000 was selected based on the fact that our original feature space size was close to 1000. However, the results in our feature selection experiments too showed an accuracy degradation pattern similar to the entire set of augmented features in Table 4. This suggests that the features being added are not merely noisy for our particular task, but strongly misleading.

Table 4: Accuracy on DA prediction task using features from previous 1, 2 and 3 turns (indicated in parentheses after the feature set name).

	Classifier					
	Decision Tree		**Naïve Bayes**		**SVM**	
Feature Set	Macro	Micro	Macro	Micro	Macro	Micro
lex+syn(1)	0.609	0.627	0.632	0.644	0.593	0.612
lex+syn(2)	0.607	0.625	0.598	0.609	0.564	0.578
lex+syn(3)	0.605	0.623	0.570	0.582	0.551	0.567
lex+syn+met(no exch)(1)	0.730	0.733	0.706	0.715	0.681	0.686
lex+syn+met(no exch)(2)	0.730	0.732	0.674	0.681	0.662	0.670
lex+syn+met(no exch)(3)	0.731	0.733	0.641	0.647	0.659	0.664
lex+syn+met(exch)(1)	0.747	0.757	0.706	0.717	0.687	0.690
lex+syn+met(exch)(2)	0.747	0.757	0.662	0.672	0.668	0.674
lex+syn+met(exch)(3)	0.745	0.755	0.636	0.647	0.658	0.664

Table 5: Accuracy on DA prediction task using sequential learning algorithms.

	Classifier			
	CRF		**Collins' Perceptron**	
History Size	Macro	Micro	Macro	Micro
1	0.740	0.749	0.788	0.792
2	0.671	0.678	**0.794**	**0.799**
3	-	-	0.793	0.799

5.3 Sequential Learning

Both CRFs and Collins' Perceptron Learner have two "tunable" parameters among others: (1) the history size, which is the number of previous labels considered while predicting the current label and (2) the number of iterations through training data. The parameter that we varied was the history size, keeping everything else to its default value in the MinorThird implementation. Note that this variation in history size is *not* done using a separate validation set, which would be the principled way of selecting the optimal history size for this dataset. However, our goal here is not choosing or reporting the optimal history size but comparing the accuracy of classification versus sequential learning algorithms under similar conditions. With CRF, we used a history of 1 and 2 labels[2]. For the Collins' Perceptron Learner, we used a history of 1, 2 and 3 labels. We used the full feature set of lexical + syntactic + meta features with manually annotated exchange boundaries. The history of labels of previous turns does not cross an exchange boundary.

5.3.1 Sequential Learning Results

Table 5 shows the results for the two sequential algorithms with different history sizes. Collins' Perceptron with a history size of 2 gave the best accuracy on our task, as measured by macro-averaged and micro-averaged accuracy values. With a history of 3, there was only a slight decrease in the macro-averaged accuracy of Collins' Perceptron. We note again that these experiments do not suggest that a history size of 2 is optimal for this dataset, that will have to be evaluated using a separate validation set. All of the results from Collins' Perceptron learner are well above the best baseline accuracy values of the naïve Bayes classifier. Contrary to this behavior however, the accuracy of CRF learner did not outperform the best baseline accuracy. Furthermore, the accuracy degraded when the history size was increased from 1 to 2.

[2]We could not complete in time the experiments for history of 3 labels with CRFs.

Table 6: Confusion matrix for the best naïve Bayes classifier.

	0	1	2	3	4	5	6	7	8
0	**95**	7	38	2	*120*	8	72	19	10
1	1	**53**	*48*	7	6	2	0	0	0
2	13	17	**423**	15	*82*	1	4	15	3
3	1	1	*25*	**65**	1	1	1	1	2
4	33	4	22	0	**1285**	12	27	*64*	14
5	8	0	8	0	*42*	**45**	14	5	2
6	25	1	7	2	*53*	13	**772**	5	2
7	6	3	17	1	*150*	1	2	**893**	5
8	8	0	10	1	*42*	0	0	6	**25**

Table 7: Confusion matrix for the best Collins' Perceptron Learner.

	0	1	2	3	4	5	6	7	8
0	**156**	6	26	6	*71*	8	66	19	13
1	0	**70**	*42*	1	2	2	0	0	0
2	13	*30*	**449**	25	25	8	7	13	3
3	1	1	*27*	**67**	1	0	0	1	0
4	21	2	27	0	**1213**	30	64	*75*	29
5	4	1	6	0	*31*	**59**	19	2	2
6	19	0	2	1	*25*	6	**823**	4	0
7	7	2	12	0	*76*	0	8	**963**	10
8	9	2	7	0	*32*	1	3	8	**30**

5.4 Observations, Discussion and Error Analysis

The observation that CRFs performed worse than the naïve Bayes baseline is quite counter-intuitive, especially given that the Collins' Perceptron learner seems to improve performance significantly. To analyze the types of errors made by the best naïve Bayes model, the best Collins' Perceptron model and the best CRF model, Tables 6, 7 and 8 show the confusion matrices after combining the results of our leave-one-dialog-out cross-validation experiments using these models. The entries along the diagonal are the correctly classified instances (in **bold**) and the ones in *underlined italics* are the most frequent misclassifications for each label. One trend that can be prominently seen is that in most cases, the most frequent erroneous prediction is 4 (assertion), which is not surprising given that it is the majority class. Both the sequential learning algorithms show a decrease in this tendency as compared to the naïve Bayes classifier, by reducing the number of erroneous predictions in that class. However, even the number of correct predictions of 4 decreases for both of them as compared to naïve Bayes. Except for 4, Collins' Perceptron consistently improves number of correctly classified examples for each class. This however does not hold for the CRF learner. A decrease in number of correctly classified examples is observed for classes 2 (closed response question), 3 (check for understanding), 4 (assertion), and 6 (direction or command). In terms of the per-class F1 measure (which can be calculated from the confusion matrices above), the best improvement was in classifying 0 (other). While this applied to the CRF learner too, its improvement was lesser than that of Collins' Perceptron due to its low precision – it tended to classify many of the turns as belonging to the "other" category, when they were not. This can be observed by comparing the first columns of the three confusion matrices.

6 Conclusions

We have applied 2 sequential learning algorithms, Conditional Random Fields and Collins' Perceptron Learner to the task of dialog act classification. Our baseline accuracy was that

Table 8: Confusion matrix for the best CRF Learner.

	0	1	2	3	4	5	6	7	8
0	**163**	8	21	2	*76*	11	48	21	21
1	1	**67**	*44*	2	2	1	0	0	0
2	33	*42*	**403**	26	34	7	3	20	5
3	7	0	*23*	**64**	0	1	1	2	0
4	70	0	44	1	**1130**	26	57	*99*	34
5	5	1	6	1	*32*	**59**	14	4	2
6	*71*	0	6	1	28	8	**763**	3	0
7	23	1	13	1	*111*	3	5	**911**	10
8	6	0	8	1	*29*	1	3	12	**32**

of the naïve Bayes classifier (chosen among Decision Trees, naïve Bayes classifier and Support Vector Machines). Sequential learning algorithms make use of predicted labels of previous turns for predicting the current label. We experimented with history sizes of 1 and 2 with CRFs and history sizes of 1, 2 and 3 with Collins' Perceptron Learner. The best performing algorithm was Collins' Perceptron Learner with history size of 2 (and also history size of 3, with a negligible decrease in macro-averaged accuracy value), with an improvement of 5.59% over the best baseline.

Acknowledgments

This work was carried out in part using hardware and software provided by the University of Minnesota Supercomputing Institute. We would like to thank Ted Pedersen for facilitating this use.

References

[1] Alex Waibel, Michael Bett, Florian Metze, Klaus Ries, Thomas Schaaf, Tanja Schultz, Hagen Soltau, Hua Yu & Klaus Zechner. Advances in Automatic Meeting Record Creation and Access. *In Proceedings of ICASSP-2001*. (2001)

[2] Dinoj Surendran & Gina-Anne Levow. Dialog Act Tagging with Support Vector Machines and Hidden Markov Models, *In Proceedings of Interspeech 2006*. (2006)

[3] Michel Galley, Kathleen McKeown, Julia Hirschberg, Elizabeth Shriberg. Identifying Agreement and Disagreement in Conversational Speech: Use of Bayesian Networks to Model Pragmatic Dependencies. *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*. (2004)

[4] Jaime Arguello & Carolyn Rosé (2006) Topic Segmentation of Dialogue. *In Proceedings of Workshop on Analyzing Conversations in Text and Speech, HLT-NAACL 2006*. (2006)

[5] John Sinclair & Malcolm Coulthard. Towards an analysis of discourse: The English used by teachers and pupils. *London: Oxford University Press*. (1975)

[6] William Cohen. Minorthird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data. *http://minorthird.sourceforge.net/*. (2004)

[7] John Lafferty, Andrew McCallum & Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *In Proceedings of the International Conference on Machine Learning (ICML-2001)*. (2001)

[8] Michael Collins. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. *In Proceedings of EMNLP 2002*. (2002)