TEXT BASED DIALOG ACT CLASSIFICATION FOR MULTIPARTY MEETINGS

Matthias Zimmermann[1], *Dilek Hakkani-Tür*[1], *Elizabeth Shriberg*[1,2], *Andreas Stolcke*[1,2]

[1]International Computer Science Institute, [2]SRI International, USA
{zimmerma,dilek,ees,stolcke}@icsi.berkeley.edu

ABSTRACT

This paper compares the performance of various machine learning approaches and their combination for dialog act (DA) classification of meetings data. For this task, boosting and three other text based approaches previously described in the literature are used. To further improve the classification performance, various combination schemes take into account the results of the individual classifiers. All classification methods are evaluated on the ICSI Meeting Corpus based on both reference transcripts and the output of a speech-to-text (STT) system. The results indicate that both the proposed boosting based approach and a method relying on maximum entropy substantially outperform the use of mini language models and a simple scheme relying on cue phrases. The best performance was achieved by combining methods with a multilayer perceptron.

1. INTRODUCTION

Dialog acts (DAs) represent the functional building blocks of conversations [1] and the classification of dialog acts corresponds to assigning DA types to the individual utterances. How these DA types are defined depends on the experimental setup. A variety of DA tagging schemes have been proposed in the literature. While earlier schemes were designed for specific applications in mind (e.g. MAPTASK [2], or VERBMOBIL [3]) more recent approaches try to impose as few constraints on the conversations as possible (e.g. DAMSL [4] or [5]). The five DA types (derived from [5]) that are distinguished in this work are statements, questions, backchannels, floorgrabbers, and disruptions. Backchannels are mostly single word utterances such as *uhhuh* or *yeah* that are used to indicate that the speaker should go on talking. Floorgrabbers subsume all utterances that are linked to the floor management in multiparty conversations. Frequently, floorgrabbers such as *well* or *i think* are used during an ongoing conversation to indicate that a participant would like to start talking. A speaker can also use floorgrabbers (e.g. *so*) while talking to indicate that she/he is not finished talking yet. Finally, disruptions categorize all utterances that can not be completed as intended by the speaker including self-interruptions.

The goal of this paper is to compare the performance of boosting based DA classification to various text based techniques described in the literature. For this, we use a tightly controlled experimental setup that only allows the methods to access isolated utterances (i.e. the classification is based on words within a given utterance and does not make use of other knowledge sources such as the sequence of utterances or prosody).

Previous work mainly investigated single methods or variations of a single method for the classification of dialog acts. Most prominently, methods relying on word n-gram language models have been investigated in various experimental setups [6, 3, 7, 8]. Semantic classification trees have been proposed in [9], transformation based learning was investigated in [10], and artificial neural networks were used in [11]. More recently, dynamic Bayesian networks have been proposed as well [12]. To our knowledge boosting-based classification of dialog acts has not been investigated so far, and no direct comparison of the performance for the methods investigated in this work is available.

The remainder of this paper is organized as follows. First, the four different DA classification methods are described in the following section. Experiments and results are described in Section 3, while conclusions and possible future work are provided in Section 4.

2. METHODOLOGY

All four DA classification methods described below attempt to predict the most likely DA type d^* for a given utterance $W = (w_1, w_2, \ldots w_n)$. In the next subsections the most widely used technique, based on DA-specific language models is covered first. Section 2.2 then summarizes a method

We would like to thank Özgür Çetin and Luke Gottlieb for their contributions in the preparation of the experimental setup. This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication), by DARPA Contract NBCHD030010 through the SRI CALO project (approved for public release, distribution unlimited), NSF Awards IIS-0121396 and IRI-9619921, and the Swiss National Science Foundation through the research network IM2.

relying on cue phrases [13]. A maximum entropy based approach is described in Section 2.3, and the proposed boosting based method is introduced in Section 2.4.

2.1. Mini Language Models

Dialog act-specific mini n-gram language models (Mini LMs) proposed by [6] have been widely used in previous work [3, 8]. For each DA type d, an individual word n-gram LM is trained on all utterances from an annotated corpus that are tagged with the desired DA type d. This training procedure allows the Mini LMs to capture the DA specific word usage and produce DA specific likelihoods $p(W|d)$.

To classify an unknown utterance W the estimates must then be multiplied by the prior probability $p(d)$ leading to the decision rule given below.

$$d^* = \underset{d}{argmax}\, p(W|d)\, p(d) \quad (1)$$

Although this method represents a principled approach that relies on the well known domain of n-gram language modeling it has the drawback of not being trained in a discriminative way.

2.2. Cue Phrases

The second technique investigated here relies on the concept of cue phrases that correspond to word n-grams up to a specified order. The scheme has been proposed in [13] and is particularly simple to implement. During training the list of cue phrases is constructed in the following way. Initially cue phrase candidates include all word n-grams of a given corpus for $n = 1$ up to $n = 4$. For each such cue phrase C its *predictivity* $p(d|C)$ is computed that measures to which extend the presence of this cue phrase indicates the specific DA type d. For each cue phrase candidate its maximal predictivity that corresponds to the most likely DA type for the presence of this cue phrase is determined. Two thresholds are then used to obtain the final cue phrases. The first threshold requires a cue phrase candidate to be observed at least a given amount of times in the training corpus, and the second threshold only retains cue phrases that exceed a fixed minimal predictivity.

For an unknown utterance W all known cue phrases are then extracted and the DA type corresponding to the cue phrase that is associated with the highest predictivity is used to output the result d^* In our implementation we used the system including position specific cues by explicit modeling of the start and the end of utterances; see [13] for further details. A potential drawback of this method lies in its decision rule that does not generalize well to produce a score for each available DA type.

2.3. Maximum Entropy

One of the main drawbacks of the methods described above lies in their training that does not explicitly optimize the discrimination between correct and incorrect DA types for a given utterance. To take advantage of discriminative training a DA classification technique based on maximum entropy modeling was proposed in [14]. Furthermore, the maximum entropy framework allows directly estimating posterior probabilities $p(d|F)$ for a DA type d and a binary feature vector F (see [15] for an excellent introduction into maximum entropy modeling).

The DA type d^* of an unknown utterance is determined by the DA type d that maximizes the posterior probability $p(d|F)$. In our case the feature vector F is extracted from the utterance W according to [14]. As features the first two words (after removing filler words), the last two words, the initial and the final word bigram, as well as the length of the utterance is used. In contrast to [14] we do not include the first word of the following DA, as our experimental setup only considers isolated utterances.

2.4. Boosting

The fourth method, boosting, is also discriminative and is derived from a text categorization task. Boosting aims to combine "weak" base classifiers to come up with a "strong" classifier. The learning algorithm is iterative, and in each iteration, a weak classifier is learned so as to minimize the training error, and a different distribution or weighting over the training examples is used to give more emphasis to examples that are often misclassified by the preceding weak classifiers. For this approach we use the *BoosTexter* algorithm described in [16], with word n-gram features, as well as features like segment length. To make it comparable with the maximum entropy approach, we have also trained the BoosTexter classifier with the same set of features.

3. EXPERIMENTS AND RESULTS

This section first describes the experimental setup. The optimization of the individual classification methods on the development sets are described in Section 3.2 and the evaluation of the methods on the test sets is provided in Section 3.3, which also investigates various combination schemes.

3.1. Experimental Setup

For all experiments reported here the experimental setup is closely linked with the one described in [14]. Of the 75 available meetings in the ICSI MRDA corpus, two meetings of a different nature are excluded (Btr001 and Btr002). From the remaining meetings we use 51 for training, 11 for development, and 11 for evaluation. The available DA types

Condition	DAs	Words	Chance	WER
Reference	113,191	740,837	45.0%	-
STT Manual	103,443	683,434	42.0%	35.4%
STT Auto	85,501	653,879	35.5%	38.2%

Table 1. Description of the three experimental conditions. The table lists the number of DAs, words, the chance classification error rate (Chance), and corresponding word error rates (WER).

DA Type	Ref	STT Manual	STT Auto
Statements	8,918	8,642	7,740
Questions	1,164	1,108	1,004
Backchannels	1,960	1,437	220
Floor-grabbers	1,924	1,768	1,306
Disruptions	2,237	1,937	1,734

Table 2. Test set frequencies of the DA types under reference conditions (Ref), STT conditions based on manual segments (STT Manual), and STT conditions using automatic segments (STT Auto).

are mapped to the following five mutually exclusive types: backchannels (B), disruptions (D), floor grabbers (F), questions (Q), and statements (S). Our setup differs in a number of aspects from [14]. In contrast to [14] we use the normalized words coming from forced alignments under reference conditions instead of the unnormalized words from the meeting transcriptions. For the speech-to-text (STT) conditions we also make use of a better, more recently developed recognizer [17]. Instead of a 39% word error rate (WER) the new recognizer achieves a 35.4% WER based on the close talking microphones. Furthermore, we define two separate STT conditions. The first one corresponds to the STT conditions of [14] and relies on a manual segmentation of the audio input stream (STT Manual). As a more realistic setup we also include an experimental setup that relies on automatic segmentation of the audio (STT Auto) leading to a higher WER of 38.2%[1]. The ground truth for both STT conditions is generated by aligning the words of the reference setup (that has been annotated for the DAs) to both of the STT conditions, with the constraint that two aligned words may not occur further apart than a fixed time (1 second). Through these alignments the STT words then inherit the DA boundaries and types from the reference conditions. See Table 1 for some statistics of the experimental conditions described in the text above.

Table 2 reports the number of DAs for each DA type for the different test sets used in our experiments. From these DA type statistics the chance error rates as reported in Table 1 is computed by the relative frequency of the statements. It is interesting to observe that, under STT Auto conditions, a large amount of single word DAs that occur in isolation are missed by the automatic segmenter. This effect is particularly dramatic for backchannels for which almost 90% of the DAs are missing compared to the reference conditions. As a result, the 5-way classification problem is almost reduced to a 4-way classification problem resulting in a significantly lower chance error rate.

[1] The difference in WER of the STT Auto conditions compared to the STT Manual conditions is mostly generated by a higher number of deletions.

3.2. System Optimization

For the Mini LM approach described in [6] the DA-specific LMs were trained using interpolated Kneser-Ney smoothing [18]. The order of the n-gram models was optimized up to $n = 4$ on the development set. Significantly different results (using a sign test) were obtained for the step from unigram LMs (error rate 36.7%) to bigram LMs (error rate 27.5%). Trigram and fourgram Mini LMs performed slightly worse but not significantly different from the bigram LMs. The choice of trigrams in [8] can be explained by the fact that the Switchboard corpus contains almost twice as many utterances as the ICSI MRDA corpus. We also compared the effect on training under STT conditions versus training under reference conditions. As training of the LMs under STT conditions slightly decreased the performance of this approach the DA-specific LMs were trained under reference conditions for all experimental setups.

For the Cue Phrase method described in [13] we measured the error rates for different maximum lengths of the cue phrases. In correspondence with [13] the best results were achieved when cue phrases up to 4-grams were used (error rate: 27.3%). In contrast to the Mini LM approach the higher order n-grams significantly helped to improve the performance over the use of bigrams only, for which an error rate of 30.4% was measured. As in the case of the Mini LM, training of the cue phrase based method worked best under reference conditions even when the models are applied under STT conditions.

In the case of the maximum entropy based method described in [14], the effect of the number of words to include at the beginning and at the end of each dialog act was investigated as well as the influence of the removal of initial filler words. As proposed in [14], best results are obtained when initial filler words are removed, although the difference in performance is not statistically significant. The number of the initial words and final words to include as features is not very critical for the performance of this method. When only the first and the last word is kept, an error rate of 22.9% is achieved under reference conditions compared to 22.6% for keeping the first two words (plus the initial word bigram) and the final two words (and the final word bigram). Adding the first three words and last

System	Ref	STT Manual	STT Auto
Mini LM [6]	26.7%	29.8%	27.3%
Cue Phrases [13]	26.6%	29.6%	28.5%
MaxEnt [14]	22.5%	26.5%	23.8%
BoosTexter	21.7%	26.9%	24.2%
Simple Voting	23.8%	27.3%	24.4%
Linear Combination	21.7%	26.3%	23.6%
MLP	21.3%	26.2%	23.3%

Table 3. Comparison of the classification error rates of the different systems under reference conditions (Ref) , STT conditions based on manual segments (STT Manual), and STT conditions using automatic segments (STT Auto). The results for the combination schemes are at the bottom.

three words even leads to a small degradation of the performance (22.7%). In contrast to the previous methods, a significant gain was found when the maximum entropy based models were trained under STT conditions for use under STT conditions. The model trained under reference conditions achieved a 27.8% error rate under STT Manual conditions and a 27.3% error rate under STT Auto conditions. For training under STT Manual conditions the error rates are reduced to 27.0% (STT Manual) and 24.7% (STT Auto).

With Boosting, we have not performed any optimization and ran the classifier for 1,000 iterations for each experimental condition. Using all unigrams, bigrams, and trigrams of an utterance as features results in a classification error rate of 22.1%. When the length of the utterance is included as a feature, the classification error is reduced to 21.7%. A similar error rate, 21.9% is achieved when the BoosTexter is trained on the same features as the maximum entropy based method. These results indicate that the words at the beginning and the end of an utterance carry most of the information that can be exploited by the classification scheme. As in the case of the maximum entropy based method, training of the models under STT conditions is beneficial.

3.3. Evaluation and System Combination

After the individual optimization of each DA classification method, the best performing configuration was used for evaluation on the test sets under the three available conditions. The resulting test set error rates are reported in Table 3. In correspondence with [13] we find that the performance of the approach based on cue phrases compares well with the mini LM based approach, in spite of the difference in both corpus and DA type definitions. Furthermore, it can be observed that the two approaches based on mini LM, and cue phrases perform significantly worse than the maximum entropy based approach and the classification scheme using boosting under all investigated conditions. Under reference conditions boosting outperforms the maximum entropy based approach. For the two STT conditions, boosting performance is slightly inferior to the maximum entropy method when all word n-grams are used as features. When we train BoosTexter on the feature set of the maximum entropy method (only keep DA initial and DA final words as features) the performance becomes the same as in the case of maximum entropy.

Condition	Words	Label	Predicted
Ref.	*it's the shadow*	S	S
STT	*it's it's uh*	S	D
Ref.	*where was heidelberg ...*	Q	Q
STT	*worse heidelberg ...*	Q	S

Fig. 1. Two typical classification errors under STT conditions forced by misrecognized words. Under reference conditions (Ref.) these DAs are correctly classified. S=Statement, D=Disruption, and Q=Question.

In a first experiment a simple voting scheme was implemented that returns the DA type most frequently predicted by the different classifiers, where in case of ties, the most frequent class is chosen. According to the results in Table 3, voting did worse than either the maximum entropy approach (MaxEnt) or the boosting based method (BoosTexter). As a more sophisticated combination method, a linear interpolation of the posterior probabilities of the DA classification methods was investigated using expectation maximization for the optimized of the interpolation weights[2]. For this combination method the probabilities of both the classifier using mini LMs and the boosting based method needed to be normalized to make sure that the results would sum up to 1 while the maximum entropy based approach directly produces posterior probabilities for all possible DA types[3]. The linear interpolation performed better than the simple voting scheme and under the STT conditions linear interpolation outperformed the individual classifiers (at a 90% level of significance). Only the multilayer Perceptron (MLP) based combination of the posterior probabilities from the Mini LM, the MaxEnt and the BoosTexter was able to significantly outperform (at the 99% level) the best individual classifiers under all conditions. For this combination method a simple feed-forward network with a single hidden layer including ten hidden neurons was trained on the development sets.

In an error analysis we combined the output of the four classifiers for each DA in the test sets to determine the or-

[2] For each experimental condition the interpolation weights were optimized on the corresponding development set.

[3] The cue phrase based approach was not considered for this experiment as this algorithm does return a probability for only the most likely DA type.

acle error rates (error rate that none of the four methods predicted the correct DA type). For the four classifiers the oracle error rate under reference conditions is 14.8%, under STT Manual conditions 18.3%, and 17.4% under STT Auto conditions. From these relatively low error rates a significant amount (35% under reference conditions and 27% under STT conditions) of these errors is forced by the experimental setup that does not include the context of DAs and does not consider prosody. As many frequent single-word DAs (*yeah*, *right*, *ok*, *uhhuh*, and *huh* can occur with different DA types. A further major source of errors under STT conditions is caused by misrecognized words that lead to readable utterances (see Fig. 1). For such cases the predicted DA types seem to be correct even for the human reader and the only chance to correctly classify such utterances is by a better speech recognition engine.

4. CONCLUSION AND OUTLOOK

We have investigated the performance of three text based techniques described in the literature for the classification of dialog acts in multiparty meetings. In addition, we proposed the use of a boosting-based method. From the results of the experiments we found that the boosting based method performs favorably compared to the other approaches studied in this paper. Specifically, our results indicate that both the boosting based approach and the method relying on maximum entropy significantly outperform the use of mini language models and the scheme relying on cue phrases. The best performance was achieved by a combination method that involved a multilayer perceptron.

In future work we will investigate support vector machines for classification similar to [19, 20] and the integration of syntactic information such as automatically derived POS tags and prosodic features. Alternatively, the combination of dialog act classification methods described in this paper should be put into a more realistic setup that considers joint segmentation and classification as described in [21].

5. REFERENCES

[1] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice Hall, 2000.

[2] A. Anderson et al., "The HCRC map task corpus," *Language and Speech*, vol. 34, no. 4, pp. 351–366, 1991.

[3] N. Reithinger and M. Klesen, "Dialog act classification using language models," in *Proc. ICASSP*, Rhodes, Greece, 1997, vol. 3, pp. 2235–2238.

[4] M. Core and J. Allen, "Coding dialogues with the DAMSL annotation scheme," in *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Cambridge, USA, 1997, pp. 28–35.

[5] E. Shriberg et al., "The ICSI meeting recorder dialog act (MRDA) corpus," in *Proc. SIGDIAL*, Cambridge, USA, 2004, pp. 97–100.

[6] M. Nagata and T. Morimoto, "First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance," *Speech Communication*, vol. 15, pp. 193–203, 1994.

[7] V. Warnke, R. Kompe, H. Niemann, and E. Nöth, "Integrated dialog act segmentation and classification using prosodic features and language models," in *Proc. 5th Europ. Conf. on Speech, Communication, and Technology*, Rhodes, Greece, 1997, vol. 1, pp. 207–210.

[8] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, no. 3, pp. 339–371, 2000.

[9] M. Mast et al, "Dialog act classification with the help of prosody," in *Proc. ICSLP*, Philadelphia, USA, 1996, vol. 3, pp. 1732–1735.

[10] K. Samuel, S. Carberry, and K. Vijay-Shanker, "Dialogue act tagging with transformation-based learning," in *Proc. 17th Int. Conference on Computational Linguisitics*, Montreal, Canada, 1998, vol. 2, pp. 1150–1156.

[11] K. Ries, "HMM and neural network based speech act detection," in *Proc. ICASSP*, Phoenix, USA, 1999, vol. 1, pp. 497–500.

[12] G. Ji and J. Bilmes, "Dialog act tagging using graphical models," in *Proc. ICASSP*, Philadelphia, USA, 2005, vol. 1, pp. 33–36.

[13] N. Webb, Mark Hepple, and Y. Wilks, "Dialog act classification based on intra-utterance features," CS-05-01, Dept. of Comp. Science, University of Sheffield, UK, 2005.

[14] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proc. ICASSP*, Philadelphia, USA, 2005, vol. 1, pp. 1061–1064.

[15] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[16] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[17] A. Stolcke, X. Anguera, K. Boakye, Ö. Çetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng, "Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system," in *Machine Learning for Multimodal Interaction: 2nd International Workshop, MLMI 2005*, S. Renals and S. Bengio, Eds., pp. 463–475. LNCS 3869, Springer, 2006.

[18] J. T. Goodman, "A bit of progress in language modeling," MSR-TR-2001-72, Machine Learning and Applied Statistics Group, Microsoft, Redmond, USA, 2001.

[19] C. Leslie, E. Eskin, and W. Stafford Noble, "The spectrum kernel: A string kernel for SVM protein classification," in *Proc. Pacific Symposium on Biocomputing*, 2002, pp. 564–575.

[20] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[21] M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke, "A* based joint segmentation and classification of dialog acts in multi-party meetings," in *Proc. 9th ASRU*, San Juan, Puerto Rico, 2005, pp. 215–219.