Improving the Quality of Minority Class Identification in Dialog Act Tagging

Adinoyi Omuya
10gen
New York, NY, USA
wisdom@10gen.com

Vinodkumar Prabhakaran
CS, Columbia University
New York, NY, USA
vinod@cs.columbia.edu

Owen Rambow
CCLS, Columbia University
New York, NY, USA
rambow@ccls.columbia.edu

Abstract

We present a method of improving the performance of dialog act tagging in identifying minority classes by using per-class feature optimization and a method of choosing the class based not on confidence, but on a cascade of classifiers. We show that it gives a minority class F-measure error reduction of 22.8%, while also reducing the error for other classes and the overall error by about 10%.

1 Introduction

In this paper, we discuss **dialog act tagging**, the task of assigning a dialog act to an utterance, where a **dialog act** (DA) is a high-level categorization of the pragmatic meaning of the utterance. Our data is email. Our starting point is the tagger described in (Hu et al., 2009), which uses a standard multi-class classifier based on support vector machines (SVMs). While the performance of this system is pretty good as measured by accuracy, it performs badly on the DA REQUEST-ACTION, which is a rare class. Multi-class SVMs are typically implemented as a set of SVMs, one per class, with the overall choice of class being determined by the SVM with the highest confidence ("one-against-all"). Multi-class SVMs are typically packaged as a single system, whose inner workings are ignored by the NLP researcher. In this paper we show that, for our problem of DA classification, we can boost the performance of the rare classes (while maintaining the overall performance) by performing feature optimization separately for each individual classifier. But we also show that we can achieve an all-around error reduction by altering the method by which the multi-class classifier combines the individual SVMs. This new method of combination is a simple cascade: we run the individual classifiers in ascending order of frequency of the classes in the training corpus; the first classifier to classify the data point positively determines the choice of the overall classifier. If no classifier classifies the data point positively, we use the usual confidence-based method. This new method obtains a 22.8% error reduction for the minority class, and around 10% error reduction for the other classes and for the overall classifier.

This paper is structured as follows. We start out by discussing related work (Section 2). We then present our data in Section 3, and in Section 4 we present the experiments with our systems and the results. We report the results of an extrinsic evaluation in Section 5, and conclude.

2 Related Work

Dialog act (DA) annotations and tagging, inspired by the speech act theory of Austin (1975) and Searle (1976), have been used in the NLP community to understand and model dialog. Initial work was done on spoken interactions (see for example (Stolcke et al., 2000)). Recently, studies have explored dialog act tagging in written interactions such as emails (Cohen et al., 2004), forums (Kim et al., 2006; Kim et al., 2010b), instant messaging (Kim et al., 2010a) and Twitter (Zhang et al., 2012). Most DA tagging systems for written interactions use a message/post level tagging scheme, and allow multiple tags for each message/post. In such a tagging scheme, indi-

Proceedings of NAACL-HLT 2013, pages 802–807,
Atlanta, Georgia, 9–14 June 2013.

vidual binary classifiers for each tag are independent of one another. However, recent studies have found merit in segmenting each message into functional units and assigning a single DA to each segment (Hu et al., 2009). Our work falls in this paradigm (we choose a single DA for smaller textual units). We build on the work by (Hu et al., 2009); we improve their dialog act predicting performance on minority classes using per-class feature optimization.

3 Data

In this study, we use the email corpus presented in (Hu et al., 2009), which is manually annotated for DA tags. The corpus contains 122 email threads with a total of 360 messages and 20,740 word tokens. This set of email threads is chosen from a version of the Enron email corpus with some missing messages restored from other emails in which they were quoted (Yeh and Harnly, 2006; Agarwal et al., 2012). Most emails are concerned with exchanging information, scheduling meetings, or solving problems, but there are also purely social emails.

Dialog Act Tag	Count (%)
REQUEST-ACTION (R-A)	35 (2.5%)
REQUEST-INFORMATION (R-I)	151 (10.7%)
CONVENTIONAL (CONV)	357 (25.4%)
INFORM (INF)	853 (60.7%)
Total # of DFUs	1406

Table 1: Annotation statistics

Each message in the thread is segmented into Dialog Functional Units (DFUs). A DFU is a contiguous span within an email message which has a coherent communicative intention. Each DFU is assigned a single DA label which is one of the following: REQUEST-ACTION (R-A), REQUEST-INFORMATION (R-I), CONVENTIONAL (CONV) and INFORM (INF). There are three other DA labels — INFORM-OFFLINE, COMMIT, and NODA for no dialog act — which occurred 5 or fewer times in the corpus. We ignore these DA labels in this paper. The corpus also contains links between the DFUs, but we do not use those annotations in this study. Table 1 presents the distribution of DA labels in our corpus. We now describe each of the DAs we consider in our experiments.

In a REQUEST-ACTION, the writer signals her desire that the reader perform some non-communicative act, i.e., an act that cannot in itself be part of the dialogue. For example, a writer can ask the reader to write a report or make coffee.

In a REQUEST-INFORMATION, the writer signals her desire that the reader perform a specific communicative act, namely that he provide information (either facts or opinion).

In an INFORM, the writer conveys information, or more precisely, the writer signals that her desire that the reader adopt a certain belief. It covers many different types of information that can be conveyed including answers to questions, beliefs (committed or not), attitudes, and elaborations on prior DAs.

A CONVENTIONAL dialog act does not signal any specific communicative intention on the part of the writer, but rather it helps structure and thus facilitate the communication. Examples include greetings, introductions, expressions of gratitude, etc.

4 System

We developed four systems for our experiments: a baseline (BAS) system which is close to the system described in (Hu et al., 2009), and three variants of our novel divide and conquer (DAC) system. Features used in both systems are extracted as explained in Section 4.2. Section 4.3 describes the baseline system, the basic DAC system, and two variations of the DAC system.

4.1 Experimental Framework

In all our experiments, we use linear kernel Support Vector Machines (SVM). However, across the systems, there are differences in how we use them. Our framework was built with the ClearTK toolkit (Ogren et al., 2008) with its wrapper for SVMLight (Joachims, 1999). The ClearTK wrapper internally shifts the prediction threshold based on posterior probabilistic scores calculated using the algorithm of Lin et al. (2007). We report results from 5-fold cross validation performed on the entire corpus.

4.2 Feature Engineering

In developing our system, we classified our features into three categories: lexical, verbal and message-

level. Lexical features consists of n-grams of words, n-grams of POS tags, mixed n-grams of closed class words and POS tags (Prabhakaran et al., 2012), as well as a small set of specialized features — StartPOS/Lemma (POS tag and lemma of the first word), LastPOS/Lemma (POS tag and lemma of the last word), MDCount (number of modal verbs in the DFU) and QuestionMark (is there a question mark in the DFU). We used the POS tags produced by the OpenNLP POS tagger. Verbal features capture the position and identity of the first verb in the DFU. Finally, message-level features capture aspects of the location of the DFU in the message and of the message in the thread (relative position and size). In optimizing each system, we first performed an exhaustive search across all combinations of features within each category. For the lexical n-gram features we varied the n-gram window from 1 to 5. This step gave us the best performing feature combination within each category. In a second step, we found the best combination of categories, using the previously determined features for each category. In this paper, we do not report best performing feature sets for each configuration, due to lack of space.

4.3 Experiments

Baseline (BAS) System This system uses the ClearTK built-in one-versus-all multiclass SVM in prediction. Internally, the multi-class SVM builds a set of binary classifiers, one for each dialog act. For a given test instance, the classifier that obtains the highest probability score determines the overall prediction. We performed feature optimization on the whole multiclass classifier (as described in Section 4.2), i.e., the same set of features was available to all component classifiers. We optimized for system accuracy. Table 2 shows results using this system. In this and all tables, we give the performance of the system on the four DAs, using precision, recall, and F-measure. The DAs are listed in ascending order of frequency in the corpus (least frequent DA first). We also give an overall accuracy evaluation. As we can see, detecting REQUEST-ACTION is much harder than detecting the other DAs.

Basic Divide and Conquer (DAC) System Like the BAS system, the DAC system also builds a binary classifier for each dialog act separately, and the component classifier with highest probability score determines the overall prediction. The crucial difference in the DAC system is that the feature optimization is performed for each component classifier separately. Each component classifier is optimized for F-measure. Table 3 shows results using this system.

	Prec.	**Rec.**	**F-meas.**
R-A	57.9	31.4	40.7
R-I	91.5	78.2	84.3
CONV	92.0	95.8	93.8
INF	91.6	95.1	93.3
Accuracy		91.3	

Table 2: Results for baseline (BAS) system (standard multiclass SVM)

	Prec.	**Recall**	**F-meas.**	**ER**
R-A	66.7	40.0	50.0	15.6
R-I	91.5	78.2	84.3	0.0
CONV	93.9	94.1	94.0	2.6
INF	91.4	96.1	93.7	5.7
Accuracy		91.7		4.9

Table 3: Results for basic DAC system (per-class feature optimization followed by maximum confidence based choice); "ER" refers to error reduction in percent over standard multiclass SVM (Table 2)

Minority Preference (DAC_{MP}) System This system is exactly the same as the basic DAC system except for one crucial difference: overall classification is biased towards a specified minority class. If the minority class binary classifier predicts true, this system chooses the minority class as the predicted class. In cases where the minority class classifier predicts false, it backs off to the basic DAC system after removing the minority class classifier from the confidence tally. Table 4 shows our results using REQUEST-ACTION as the minority class.

Cascading Minority Preference (DAC_{CMP}) System This system is similar to the Minority Preference System; however, instead of a single supplied minority class, the system accepts an ordered list of classes. The classifier then works, in order, through this list; whenever any classifier in the list predicts

	Prec.	Recall	F-meas.	ER
R-A	66.7	45.7	54.2	22.8
R-I	91.5	78.2	84.3	0.0
CONV	93.9	94.1	94.0	2.6
INF	91.6	96.0	93.8	6.5
Accuracy		91.8		5.7

Table 4: Results for minority-preference DAC system — DAC_{MP} (first consult REQUEST-ACTION tagger, then default to choice by maximum confidence); "ER" refers to error reduction in percent over standard multiclass SVM (Table 2)

true, for a given instance, it then assigns this class as the predicted class. The subsequent classifiers in the list are not run. If all classifiers predict false, we back off to the basic DAC system, i.e., the component classifier with highest probability score determines the overall prediction. We ordered the list of classes in the ascending order of their frequencies in the training data. This ordering is driven by the observation that the less frequent classes are also hard to predict correctly. Table 5 shows our results using the ordered list: (REQUEST-ACTION, REQUEST-INFORMATION, CONVENTIONAL, INFORM).

	Prec.	Recall	F-meas.	ER
R-A	66.7	45.7	54.2	22.8
R-I	91.0	80.8	85.6	8.4
CONV	93.7	95.3	94.5	10.1
INF	92.4	95.8	94.0	10.0
Accuracy		92.2		10.6

Table 5: Results for cascading minority-preference DAC system — DAC_{CMP} (consult classifiers in reverse order of frequency of class); "ER" refers to error reduction in percent over standard multiclass SVM (Table 2)

4.4 Discussion

As shown in Table 3, the basic DAC system obtained a 15.6% F-measure error reduction for the minority class REQUEST-ACTION over the BAS system. It also improves performance of two other classes — CONVENTIONAL and INFORM, and obtaines a 4.9% error reduction on overall accuracy. Recall here that the only difference between the DAC system and the BAS system is the per-class feature optimization and therefore this must be the reason for this boost in performance. When we turn to DAC_{MP}, we see that the performance on the minority class REQUEST-ACTION is further enhanced, with an F-measure error reduction of 22.8%; the overall accuracy improves slightly with an error reduction of 5.7%. Finally, DAC_{CMP} further improves the performance. Since the method of choosing the minority class REQUEST-ACTION does not change over DAC_{MP}, the F-measure error reduction remains the same. However, now all three other classes also improve their performance, and we obtain a 10.6% error reduction on overall accuracy over the baseline system.

Following (Guyon et al., 2002), we performed a post-hoc analysis by inspecting the feature weights of the best performing models created for each individual classifier in the DAC system. Table 6 lists some interesting features chosen during feature optimization for the individual SVMs. We selected them from the top 25 features in terms of absolute value of feature weights.

Some features help distinguish different DA categories. For example, the feature *QuestionMark* is the feature with the highest negative weight for INFORM, but has the highest positive weight for REQUEST-INFORMATION. Features like *fyi* and *period (.)* have high positive weights for INFORM and high negative weights for CONVENTIONAL. Some other features are important only for certain classes. For e.g., *please* and *VB_NN* are important for REQUEST-ACTION, but not so for other classes. Overall, the most discriminating features for both INFORM and CONVENTIONAL are mostly word ngrams, while those for REQUEST-ACTION and REQUEST-INFORMATION are mostly POS ngrams. This shows why our approach of per-class feature optimization is important to boost the classification performance.

Another interesting observation is that the least frequent category, REQUEST-ACTION, has the least strong indicators (as measured by feature weights). Presumably this is because there is much less training data for this class. This explains why our cascading classifiers approach giving priority to the least frequent categories worked better than a simple confidence based approach, since the simple approach drowns out the less confident classifiers.

REQUEST-ACTION	REQUEST-INFORMATION	CONVENTIONAL	INFORM
please (0.9)	QuestionMark (6.6)	StartPOS_NNP (2.7)	QuestionMark (-3.0)
VB_NN (0.7)	_BOS_PRP (-1.2)	thanks (2.3)	thanks (-2.2)
you_VB (0.3)	WRB (1.0)	. (-2.0)	. (2.2)
PRP (-0.3)	PRP_VBP (-0.9)	fyi (-2.0)	fyi (1.9)
MD_PRP_VB (0.3)	_BOS_MD (0.8)	, (0.9)	you (-1.0)
will (-0.2)	_BOS_DT (-0.7)	QuestionMark (-0.8)	can_you (-0.9)

Table 6: Post-hoc analysis on the models built by the DAC system: some of the top features with corresponding feature weights in parentheses, for each individual tagger. (POS tags are capitalized; _BOS_ stands for Beginning Of Sentence)

5 Extrinsic Evaluation

In this section, we perform an extrinsic evaluation for the dialog act tagger presented in Section 4 by applying it to the task of identifying Overt Displays of Power (ODP) in emails, proposed by Prabhakaran et al. (2012). The task is to identify utterances where the linguistic form introduces additional constraints on its responses, beyond those introduced by the general dialog act. The dialog act features were found to be useful and the best performing system obtained an F-measure of 65.8 using gold dialog act tags. For our extrinsic evaluation, we retrained the ODP tagger using dialog act tags predicted by our BAS and DAC_{CMP} systems instead of gold dialog acts. ODP tagger uses the same dataset as ours for training. In the cross validation step, we made sure that the test folds for ODP were excluded from training the taggers to obtain DA tags. At each ODP cross validation step, we trained a BAS or DAC_{CMP} tagger using ODP's training folds for that step and used tags produced by that tagger for both training and testing the ODP tagger for that step. Table 7 lists the results obtained.

	Prec.	**Rec.**	**F-meas.**
No-DA	55.7	45.4	50.0
Gold-DA	75.8	58.1	65.8
BAS-DA	60.6	46.5	52.6
DAC_{CMP}-DA	67.2	45.4	54.2

Table 7: Results for ODP system using various sources of DA tags

Using BAS tagged DA, the F-measure of ODP system reduced by 13.2 points to 52.6 from using gold dialog acts (F=65.8). Using DAC_{CMP}, the F-measure improved over BAS by 1.6 points to 54.2. This constitutes an error reduction of 12.1%, taking the system using gold DA tags as the reference. This improvement is noteworthy, given the fact that the overall error reduction obtained by DAC_{CMP} over BAS in the DA tagging was around 10.6%. Also, the DAC_{CMP}-based ODP system obtained an error reduction of about 26.6% over a system that does not use the DA features at all (F=50.0).

6 Conclusion

We presented a method of improving the performance of dialog act tagging in identifying minority classes by using per-class feature optimization and choosing the class based on a cascade of classifiers. We showed that it gives a minority class F-measure error reduction of 22.8% while also reducing the error on other classes and the overall error by around 10%. We also presented an extrinsic evaluation of this technique on detecting Overt Displays of Power in dialog, where we achieve an error reduction of 12.1% over using the standard multiclass SVM to generate dialog act tags.

Acknowledgements

This work is supported, in part, by the Johns Hopkins Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor. While working on this project, the first author Adinoyi Omuya was affiliated with the Center for Computational Learning Systems at Columbia University. We thank several anonymous reviewers for their constructive feedback.

References

Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. 2012. A Comprehensive Gold Standard for the Enron Organizational Hierarchy. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–165, Jeju Island, Korea, July. Association for Computational Linguistics.

J. L. Austin. 1975. *How to Do Things with Words*. Harvard University Press, Cambridge, Mass.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to Classify Email into "Speech Acts" . In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 309–316, Barcelona, Spain, July. Association for Computational Linguistics.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Mach. Learn.*, 46:389–422, March.

Jun Hu, Rebecca Passonneau, and Owen Rambow. 2009. Contrasting the Interaction Structure of an Email and a Telephone Corpus: A Machine Learning Approach to Annotation of Dialogue Function Units. In *Proceedings of the SIGDIAL 2009 Conference*, London, UK, September. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making Large-Scale SVM Learning Practical. In Bernhard Schölkopf, Christopher J.C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, USA. MIT Press.

J. Kim, G. Chern, D. Feng, E. Shaw, and E. Hovy. 2006. Mining and Assessing Discussions on the Web Through Speech Act Analysis. In *Proceedings of the Workshop on Web Content Mining with Human Language Technologies at the 5th International Semantic Web Conference*.

S.N. Kim, L. Cavedon, and T. Baldwin. 2010a. Classifying Dialogue Acts in One-on-one Live Chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871. Association for Computational Linguistics.

S.N. Kim, L. Wang, and T. Baldwin. 2010b. Tagging and Linking Web Forum Posts. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 192–202. Association for Computational Linguistics.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2007. A Note on Platt's Probabilistic Outputs for Support Vector Machines. *Mach. Learn.*, 68:267–276, October.

Philip V. Ogren, Philipp G. Wetzler, and Steven Bethard. 2008. ClearTK: A UIMA toolkit for statistical natural language processing. In *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC)*.

Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. 2012. Predicting Overt Display of Power in Written Dialogs. In *Human Language Technologies: The 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Montreal, Canada, June. Association for Computational Linguistics.

J.R. Searle. 1976. A Classification of Illocutionary Acts. *Language in society*, 5(01):1–23.

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

J.Y. Yeh and A. Harnly. 2006. Email Thread Reassembly Using Similarity Matching. In *Third Conference on Email and Anti-Spam (CEAS)*, pages 27–28.

R. Zhang, D. Gao, and W. Li. 2012. Towards Scalable Speech Act Recognition in Twitter: Tackling Insufficient Training Data. *EACL 2012*, page 18.