Posting Act Tagging Using Transformation-Based Learning

Tianhao Wu, Faisal M. Khan, Todd A. Fisher, Lori A. Shuler and William M. Pottenger

Computer Science and Engineering, Lehigh University

{tiw2, fmk2, taf2, lase, billp}@lehigh.edu

Abstract

In this article we present the application of transformation-based learning (TBL) [1] to the task of assigning tags to postings in online chat conversations. We define a list of posting tags that have proven useful in chat-conversation analysis. We describe the templates used for posting act tagging in the context of template selection. We extend traditional approaches used in part-of-speech tagging and dialogue act tagging by incorporating regular expressions into our templates. We close with a presentation of results that compare favorably with the application of TBL in dialogue act tagging.

1. Introduction

The ephemeral nature of human communication via networks today poses interesting and challenging problems for information technologists. The sheer volume of communication in venues such as email, newsgroups, and chat-rooms precludes manual techniques of information management. Currently, no systematic mechanisms exist for accumulating these artifacts of communication in a form that lends itself to the construction of models of semantics [5]. In essence, dynamic techniques of analysis are needed if textual data of this nature is to be effectively mined.

At Lehigh University we are developing a text mining tool for analysis of chat-room conversations. Project goals concentrate on the development

of functionality to answer questions such as "What topics are being discussed in a chat-room?", "Who is discussing which topics?" and "Who is interacting with whom?" In order to accomplish these objectives, it is necessary to first identify threads in the conversation (i.e., topic threads). One of the first steps in our approach to thread identification is the automatic assignment of tags that characterize postings. These tags identify the type of posting; for example, Greet, Bye, etc. We term this classification task Posting Act Tagging.

Posting act tagging aids in both social and semantic analysis of chat data. For example, the question tag type, which consists of Yes-No-Question and Wh-Question tags, identifies postings that give clues to both the start and topic of a topic thread within a chat conversation. Other postings tagged as a Greet or Bye, for example, may not contribute significantly to the semantics of a particular topic thread. These types of postings, however, may yield information important to the social analysis of the conversation – e.g., "who is talking with whom?" Thus the tag type assigned to a posting aids the model-building process.

Posting act tagging idea is similar to dialogue act tagging. [4] demonstrates that dialogue act tagging can be widely used in dialogue recognition. Chat conversations are similar to spoken dialogues in some ways. In fact, a chat conversation is a kind of dialogue in written form. Therefore, we expect that techniques applied in dialogue recognition may also be useful in chat conversation analysis.

Chat conversations differ, however, in significant ways from dialogues[1]. Chat conversations are usually informal, and multiple topics may be discussed simultaneously. In addition, multiple people are often involved. Participants do not always wait for responses before posting again. Furthermore, abbreviations and emotion icons are frequently used in chat conversations, mixed together with chat-system-generated information.

Based on these and related issues, we have extended dialog act tagging as presented in [3] [4] [6] [7] to classify postings in chat conversations.

In the following sections we present our approach to posting act tagging. We detail the posting tag types that we have used, including some new types specific to chat conversations, in section 2. In section 3 we briefly describe the machine-learning framework that we have employed, transformation-based learning (TBL) [1]. In section 4, we present the application of TBL to posting act tagging. We discuss preliminary experimental results, including a statistical analysis comparing our results with those obtained in dialogue act tagging, in section 5. Finally, we

[1] E.g., the classification work reported in [3] is based on recorded conversations of phone calls to schedule appointments.

discuss conclusions and future work in section 6 and acknowledge those who have contributed to this work in section 7.

2. Posting Act Tags

Table 1 is our posting act tag list and includes 15 tag types. The tag types come from three different sources. The Accept, Bye, Clarify, Greet, and Reject tags are drawn from the VerbMobil project [6]. The Statement, Wh-Question¸ Yes-No-Question, Yes-Answer, No-Answer, Continuer and Other tag types derive from the dialogue act tagging research reported in [7]. As noted above, the final three tag types are specific to chat conversations and were included based on our research: Emotion, Emphasis, and System.

Table 1. List of Tags, Examples and Frequency

Tag	Example	%
Statement	I'll check after class	42.5
Accept	I agree	10.0
System	Tom [JADV@11.22.33.44] has left #sacba1	9.8
Yes-No-Question	Are you still there?	8.0
Other	*********	6.7
Wh-Question	Where are you?	5.6
Greet	Hi, Tom	5.1
Bye	See you later	3.6
Emotion	LOL	3.3
Yes-Answer	Yes, I am.	1.7
Emphasis	I do believe he is right.	1.5
No-Answer	No, I'm not.	0.9
Reject	I don't think so.	0.6
Continuer	And …	0.4
Clarify	Wrong spelling	0.3

Altogether there are over 40 tags employed in dialogue act tagging [8]. In this article, we select a subset of higher frequency dialogue act tags as our posting act tags, and add three chat-specific tags as noted above.

Statement is the most often used tag in dialogue act tagging. It covers more than 36% of the utterances. The Statement tag also has a high frequency in our posting act tagging because we use Statement to cover more than one tag used in dialogue act tagging. Statement can be split into several tags if more detailed tagging information is desired. The tag Other

is used for postings that do not fit readily into the other categories (i.e., are untagged).

Since [6] and [7] give clear definitions of the tags used in their work, we briefly define the System, Emotion, and Emphasis tags that we have added.

System postings are generated by chat-room software. For example, when a person joins or leaves a chat room, the chat-room software usually posts a System message.

People express strong feelings in Emotion postings. These feelings include "surprise", "laughing", "happiness", "sadness", etc. Most chat-room software supports emotion icons, and these icons give clues to participants' emotional states.

Emphasis postings are the postings in which people emphasize something. For instance, people often use "do" just before a verb to put more emphasis on the verb. Another example is the use of "really" to likewise emphasize a verb.

In this section, we introduced posting act tags for chat conversations. Some of tags are derived from dialogue act tagging; others are specific to chat conversations. Table 1 list all tags, example postings for each tag, and tag distribution in our datasets (training and testing). In the following two sections, we describe how TBL is applied in the discovery of rules for classifying postings automatically.

3. Overview of Transformation-Based Error-Driven Learning

Transformation Based Learning (TBL) is an emerging technique with a variety of potential applications within textual data mining. TBL has been utilized for tasks such as part-of-speech tagging, dialogue act tagging, and sentence boundary disambiguation, to name a few. TBL performs admirably in these tasks since they rely on the contextual information within textual corpora.

The core functionality of TBL is a three-step process composed of an initial state annotator, templates, and a scoring function [2]. The initial state annotator begins by labeling unannotated input (e.g., postings) with tags based on simple heuristics. Using a scoring function, the annotated input is then compared to a 'ground truth' consisting of the same text with the correct labels. TBL automatically generates transformation rules that rewrite labels in an attempt to reduce the error in the scoring function.

Potential rewrite rules are automatically generated from preexisting human-expert-generated templates. The input in question is then re-annotated using newly generated rules, and once again compared with the ground truth. The procedure selects the best rule (the one with minimal error) and saves it to a final rule sequence. This cycle repeats until the reduction in error reaches a predetermined minimum threshold.

At heart, TBL is a greedy learning algorithm. Within each learning iteration, a large set of different transformation rules can be generated. The rule with the best performance (least error as measured by the scoring function) is chosen. The final set of rules can be used for classification of new input.

4. Using TBL in Posting Act Tagging

In this section, we discuss the application of the three steps in the TBL learning process discussed in section 3 to posting act tagging with a special emphasis on template selection. We extend traditional approaches used in template selection in part-of-speech tagging and dialogue act tagging by incorporating regular expressions into the templates.

4.1. Initial State Annotator

Since the Statement tag occurs most frequently in our data sets, we simply tag each posting as Statement in the initial annotator of TBL. If the initial state has high accuracy, the learning process will be more efficient because TBL is a greedy algorithm. Therefore, Statement is the best choice for the initial state of each posting in posting act tagging.

4.2. Template Selection

Through manual study of patterns in chat data we developed a number of rule templates. In this section we discuss the antecedents of seven such templates.

1. "A particular string 'W' appears within the current posting, where 'W' is a string with white space preceding and following." Domain-expert-identified regular expressions are used to replace 'W' during learning. This template was chosen since manual inspection of chat data yielded the result that certain words are often crucial in posting tagging. For

example, a posting with the word "Why" often indicates that this is a Wh-Question posting. This is also true for dialogue act tagging [3]. However, just using a single word in this template is not sufficient for posting act tagging. This is due to the fact that chat conversations are complex. One of complexities is that typos and variations of words frequently occur. For instance, "allright", "all right", "alright", "allriggggght" all have similar meaning. It is not feasible to include all variations explicitly in a template. As a result, we employed regular expressions. For example, the regular expression "al+( )?rig+ht" covers all four variations of "all right". In section 5 we present a statistical comparison between explicit representations of words vs. the use of regular expressions to confirm this intuitive result.

2. "A character 'M' appears in the current posting, where 'M' is any punctuation mark." Punctuation marks are valuable in posting tagging. For example, a question mark usually indicates that a posting is indeed a question.
3. "A word with part of speech tag 'T' appears in the current posting, where 'T' is a part of speech tag from the Brown tag set."[2] Part of speech tags often aid in identifying posting tags. For instance, the part of speech tag WRB (when, how, etc.) can be used to identify Wh-Question postings.
4. "The current posting's length (the number of words) is 'L', or the current posting's length is greater than 'L', where 'L' is a heuristically chosen constant." In this case, we observe that some postings' tags are related to their length. For example, Yes-Answer and No-Answer postings are usually shorter while Statement postings are often long.
5. "The author of the preceding or following posting is the same as the author of the current posting." Each participant in the chat environment is termed an author. We noted that authors often separate their sentences into several consecutive postings. Thus, it is likely that a posting is a Continuer if its neighbor postings have the same author. For example, the posting "<Tom> and at least try it" is a Continuer of the posting "<TOM> go to play basketball". These two postings come together and have the same author and topic.
6. "The first character of a posting is 'C'." The first character gives a crucial clue in classifying system postings. For instance, system postings do not have (human) authors, whereas author names (i.e., screen names) are usually delimited using characters such as "(" and ")", "<" and ">", or "[" and "]" in chat conversations. In IRC chat

[2] See chapter 4 in [10] for a listing of the Brown Tags used in part-of-speech tagging.

conversations, “Tom [JADV@11.22.33.44] has joined #sacba1” is an example System posting that is generated by the chat server when a user joins the current conversation. Conversely, a delimiter followed by an author name is often the start of a non-system posting. For example, “<Tom> How is everyone?” Therefore, the first character is useful in discriminating whether a posting is a System posting or not.

7. “The previous or following posting’s tag is ‘T’” is also helpful to determine the current posting’s tag. For example, a Yes-Answer or No-Answer is normally nearby a Yes-No-Question.

We have listed seven antecedents useful as templates in TBL. It is necessary, however, that templates have consequents. In our research it sufficed to have a single consequent for all seven antecedents: “Change the current posting’s tag to ‘B’, where ‘B’ is any tag”. An example of a rule generated using template number one is: “if al+( )?rig+ht is in a posting, then change the posting’s tag to Accept”.

The learning process instantiates rules based on these templates using features present in the input postings. As a result, each template generates numerous rules. Within a given iteration during training, all of the rules so generated are applied to the postings in the training set and the rule with the best performance is chosen. We use a scoring function described in section 4.3 to measure learning performance.

4.3. Scoring Function

As in [1], we use accuracy, which is defined in Equation 1, as the scoring function for TBL. If a posting’s tag assigned during learning is the same as the correct tag in the ground truth, that posting is a true positive (TP). The definition of accuracy for posting act tagging is thus:

$$Accuracy = \frac{\#\ of\ \ TP}{\#\ of\ \ Total\ \ Postings} \qquad (1)$$

In this section, we have described the implementation of the three core steps of the TBL learning process for posting act tagging, with an emphasis on template selection. Our experimental results reported in the next section show that this approach is viable.

5. Experimental Results

In this section, we describe the datasets for training and testing. We use the widely applied technique of cross-validation to evaluate our models. We also analyze each tag's precision and recall for the test datasets. Finally, we do a statistical comparison between explicit representations of words vs. the use of regular expressions in templates.

Table 2. Training and Testing Data

Conversations	Postings	Authors
Conversation 1	384	9
Conversation 2	736	16
Conversation 3	97	7
Conversation 4	184	3
Conversation 5	262	6
Conversation 6	634	16
Conversation 7	368	7
Conversation 8	246	9
Conversation 9	218	3

Our datasets include nine IRC chat conversations containing 3129 postings in all. Each posting in each data set was manually tagged by a human expert, thereby creating our ground truth. The distribution of each tag over all chat conversations was depicted in Table 1 (in section 2). Table 2 portrays the characteristics of the nine data sets.

Table 3. Cross-validation results

Training sets	# of Rules learned	Test set	Test accuracy (%)
2,3,4,5,6,7,8,9	59	1	80.46875
1,3,4,5,6,7,8,9	63	2	76.90217
1,2,4,5,6,7,8,9	59	3	77.31959
1,2,3,5,6,7,8,9	60	4	71.19565
1,2,3,4,6,7,8,9	58	5	76.33588
1,2,3,4,5,7,8,9	56	6	76.81388
1,2,3,4,5,6,8,9	61	7	78.26087
1,2,3,4,5,6,7,9	64	8	80.89431
1,2,3,4,5,6,7,8	60	9	79.81651

In evaluating our approach we employed nine-fold cross-validation. Eight of the nine datasets were combined to form nine training sets, and the remaining dataset was used for testing. Table 3 presents the resulting nine test accuracies. The first column details the nature of each training set. For example, (2,3,4,5,6,7,8,9) means we used chat conversations two through nine to form the training set. The second column is the number of rules learned based on the given training set. The third column reports the dataset used for testing, and the last column is the accuracy that results from the application of the learned rule sequence on the given test dataset.

The best accuracy we achieved on any single test set is 80.89%, which is somewhat less than the best single-test-set accuracy reported in [4] (84.74%) for dialogue act tagging. From Table 3, our average test accuracy is 77.56% with σ=2.92%. Compared to the best average accuracy reported in [3] of 75.12%, our accuracy is slightly better. We conducted a statistical analysis using a one-tailed t-test to compare our results with those reported in [3]. We determined with greater than 95% confidence that our accuracy is significantly greater than that reported in [3]. As a result, we conclude that our approach to posting act tagging compares favorably with related work in the field of dialogue act tagging and is therefore viable.

Table 4. Average precision and recall for each tag

Tag	Precision	Recall
Statement	0.747266	0.925508
Accept	0.714286	0.273312
System	0.99026	0.993485
Yes-No-Question	0.687023	0.72
Other	0.900433	0.985782
Wh-Question	0.564103	0.5
Greet	0.885906	0.830189
Emotion	0.896104	0.663462
Bye	0.957447	0.79646
Yes-Answer	0.506173	0.773585
Emphasis	0.5	0.021739
No-Answer	0	0
Reject	0	0
Continuer	0	0
Clarify	0	0

The average precision and recall for each tag individually is depicted in Table 4. For example, we see that TBL succeeds in discovering System, Other, and Greet tags because all of them have both high precision and

high recall. Yes-No-Question, Statement, Emotion, Bye, and Yes-Answer have more modest but still reasonable precisions and recalls. Accept has high precision with low recall. On the other hand, the classification rules generated by TBL have relatively poor performance on Emphasis, No-Answer, Reject, Continuer, and Clarify tags, with precision and recall close to zero. One reason for this relatively poor performance is the sparseness of representation – these tags are not well represented in the ground truth (i.e., 13 occurrences of Continuer, 20 of Reject, 29 of No-Answer, 8 of Clarify, and 46 of Emphasis).

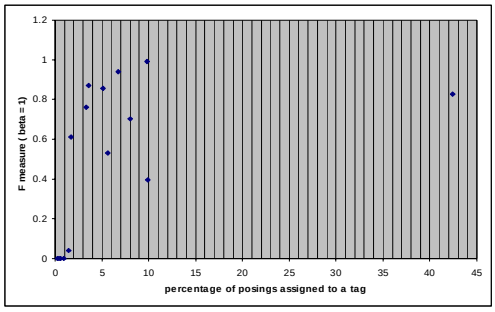

Fig. 1. Percentage of postings per tag vs. F-measure

As noted there are 3129 postings in all, and all postings were used in the nine-fold cross-validation evaluation. As a result, all of these tags have low occurrence frequencies in our datasets (Table 1). Figure 1 shows that the F-measure [11] (a combination of precision and recall) becomes reasonable once a tag occurs in at least 1.7% of the postings. A second reason for the poor performance of the classifier with these particular tags may have to do with the need for more specialized templates to handle these tag types.

To make the comparison between explicit representations of words vs. the use of regular expressions, we employed only template one for both training and testing. Each chat conversation was used in turn, first as a training set to generate a TBL rule sequence. In order to obtain a statistically significant sample, we chose combinations of seven out of the eight remaining datasets as test sets. In this way we generated eight test sets for each training set for a total of 9*8=72 test results. Table 5 depicts example test results using dataset number one as a training set. We applied a one-tailed t-test to the two distributions, one using an explicit representation and the second using regular expressions. Based on this we determined with a confidence of over 94% that regular expressions perform significantly better than an explicit representation. Therefore, we

conclude that regular expressions should be used when constructing templates such as template number one.

Table 5. Explicit representation vs. regular expressions in dataset one

Test set	Accuracy using explicit representation (%)	Accuracy using regular expressions (%)
1	52.1652563	53.3101045
2	53.0052265	54.0505226
3	53.2820281	54.3232232
4	51.8066635	52.9798217
5	55.5449025	58.4368737
6	52.691358	53.5802469
7	52.9576153	53.842571
8	52.8735632	54.0229885

Our experimental results provide evidence that TBL can be usefully applied to the problem of posting act tagging. We achieved reasonable and stable test set performance for all nine of our test datasets, and our results compare favorably with similar results obtained in dialogue act tagging.

6. Conclusion

We have presented a novel application of transformation-based learning to the problem of identifying postings in chat-room conversations. Posting act tagging aids in the formation of models of social and semantic relationships within chat data. Tagging of this nature thus represents an important first step in the construction of models capable of automatically extracting information from chat data and answering questions such as "What topics are being discussed in a chat room?", "Who is discussing which topics?" and "Who is interacting with whom?".

In the work reported in this article a well known natural language processing algorithm, transformation-based learning, has been applied to posting act tagging. We developed seven templates that have proven useful in learning rules for posting act tagging. Furthermore, we have shown that the use of regular expressions in templates improves test set accuracy.

One of the tasks that lie ahead is to deal with multiple-sentence postings that call for more than one tag (on a single posting). Transformation-based learning, however, is not suited for learning problems of this nature, and as a result we are developing a new algorithm, BLogRBL [9], to handle such cases. Another task relates to the difficulty in manually creating generic regular expressions for templates. Little work has been done in the

automatic generation of such regular expressions. At Lehigh University, however, we are engaged in a project with Lockheed-Martin and the Pennsylvania State Police to develop this capability. Finally, it is necessary to identify additional templates for tags that are not well represented in the training data. Our future work will focus on these problems.

Acknowledgements

This work was supported in part by NSF EIA grant number 0196374. The authors gratefully acknowledge the help of family members and friends, and co-authors William M. Pottenger and Tianhao Wu gratefully acknowledge the continuing help of their Lord and Savior, Yeshua the Messiah (Jesus Christ) in their lives.

References

[1] Brill, Eric. A report of recent progress in Transformation-based Error-driven Learning. Proceedings of the ARPA Workshop on Human Language Technology.1994.

[2] Brill, Eric. Transformation-Based Error Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics 21(94): 543-566. 1995.

[3] Ken Samuel, Sandra Carberry and K. Vijay-Shanker. Dialogue Act Tagging with Transformation-Based Learning. Proceedings of COLING/ACL'98, pp. 1150-1156. 1998.

[4] E. Shriberg, R. Bates, P. Taylor, A. Stolcke, D. Jurafsky, K. Ries, N. Cocarro, R. Martin, M. Meteer, and C. Van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech?, Language and Speech, 41:439—487. 1998.

[5] William M. Pottenger, Miranda R. Callahan, Michael A. Padgett. Distributed Information Management. Annual Review of Information Science and Technology (ARIST Volume 35). 2001.

[6] Alexandersson, J., Buschbeck-Wolf, B., Fujinami, T., Maier, E., Reithinger, N., Schmitz, B. & Siegel, M. Dialogue acts in VERBMOBIL-2. Verbmobil Report 204, DFKI, University of Saarbruecken. 1997.

[7] Jurafsky, D., Shriberg, E., Fox, B. & Curl, T. Lexical, Prosodic, and Syntactic Cues for Dialog Acts. Proceedings of ACL/COLING 98 Workshop on Discourse Relations and Discourse Markers, pp. 114-120, Montreal. (1998).

[8] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Meteer, M., and Van Ess-Dykema, C. Dialogue Act Modeling

for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics, 26:3. 2000.
[9] Tianhao Wu, and William M. Pottenger. Error-Driven Boolean-Logic-Rule-Based Learning: A Powerful Extension of Transformation-Based Learning. Lehigh CSC 2002 Technical Reports LU-CSE-02-008. October, 2002.
[10] Christopher D. Manning and Hinrich Schütze. Foundations of Statistical Natural Language Processing, MIT Press, 2000.
[11] Van Rijsbergen. Information Retrieval. Butterworths, London. 1979.