# Modeling the Process of Online Q&A Discussions using a Dialogue State Model

Shitian Shen and Jihie Kim

University of Southern California Information Sciences Institute

4676 Admiralty Way, Marina del Rey, CA, U.S.A

shitians@usc.edu, jihie@isi.edu

**Abstract:**

Online discussion board has become increasingly popular in higher ed-ucation. As a step towards analyzing the role that students and instructors play during the discussion process and assessing students' learning from discussions, we model different types of contributions made by instructors and students with a dialogue-state model. By analyzing frequent Q&A discussion patterns, we have developed a graphic model of dialogue states that captures the information role that each message plays, and used the model in analyzing student discussions. We present several viable approaches including CRF, SVM, and decision tree for the state classification. Using the state information, we analyze information exchange patterns and resolvedness of the discussion. Such analyses can give us a new insight on how students interact in online discussions and kind of assistance needed by the students.

**Keywords:** online discussions, dialogue transition, speech act, CRF.

## 1. Introduction

Online discussion boards, an application of social network on education, provides a platform for students and instructors to share their ideas or to discuss their question not only in traditional courses but also in web-based courses. Such tools can help students solve their problems opportunely, as well as improving instructors' work efficiency. As the discussion board usage increases, we want to understand how students interact with instructors and peers, and how they learn through that interaction.

Although research in online chat and discussion analysis has been increasing re-cently [8,12,14], there has been limited research on modeling the process of information exchange in Q&A forums or how resolvedness of discussions can be determined. In order to analyze and model the process of information exchange, we map interactions in discussions into a Q&A dialogue state model. The state for each message illustrates the status and function of the given message in the Q&A process (discussion thread) [5,6]. We identified six distinctive and frequent states in the discussion process: Problem presenting, Problem understanding, Solving, Solution understanding, Solution objecting, and Solution appreciation. In order to classify the dialogue states efficiently, we apply machine classifiers including linear Conditional Random Fields (CRF), a widely used tool for characterizing the sequential data. The features are generated from message content and positional information, including cue word posi-tions, participants' order, which provides additional hints for state labeling.

The results indicate that frequent states can be reasonably identified using ma-chine classifiers. We demonstrate that the state model can be used in finding frequent patterns in the dialogue progress and evaluating the roles the instructor and students play during the Q&A discussion process. Furthermore, we show that state information can help identifying unresolved discussions, which can be reported to the instructor.

## 2. State transition model of Q&A discussions

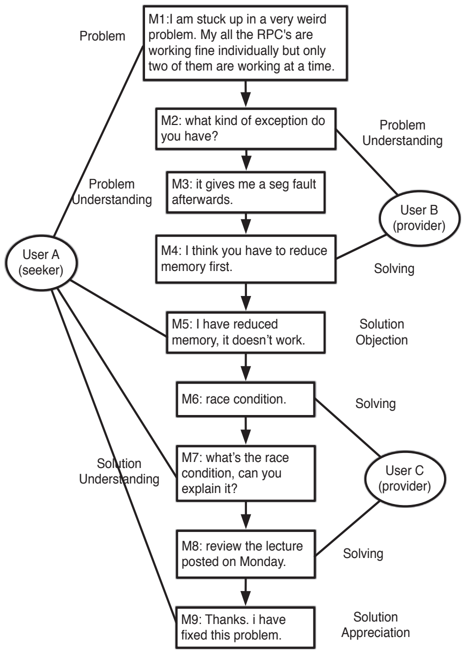


**Fig.1.** An example of discussion thread.

We use discussion corpus from undergradu-ate operating systems courses. The courses contain programming projects, and students use discussions to share problems and get help from the instructor and other students. Figure 1 shows an example discussion thread with a sequence of message. User A, B and C represents the participants. User A initiates the thread by describing the problem and asks for help. User B asks for more details related to the problem and User A provides some information. User B then gives a possible solution and User A complains that it doesn't solve the problem. User C offers another answer, and User A asks a related question. User C provides an additional suggestion. Finally, User A acknowledges the help with thanking.

Through analyses of the discussion corpus, we identified six distinctive and fre-quent states. User roles are relevant to characterizing the states: information seeker and information provider, and often the role of a user stays the same within a short dis-cussion thread [16]. The first state (Problem or P) is presented by a Seeker. In Figure 1, M1 can be regarded as a P state. In the second state (Problem Understanding or PU), the problem is further elaborated and discussed. PU can consist of multiple messages. Another discussant (student or the instructor) may post a question in order to under-stand the problem that the seeker confronts. Such questions are usually followed by an answer by the seeker who posted the problem. For example, M2 and M3 help the participants understand the problem. In the third state (Solving or S), a participant provides a direct solution or a hint. Although we label it as S, the grammatical form for such messages may vary. For example, hints can be provided as a question:"why not try ABC?". After Solving, the seeker (or other participants) can respond with Solution Appreciation (SA), Solution Objection (SO) or Solution Understanding (SU). In SA, seeker can acknowledge the assistance with thanking, like M9.

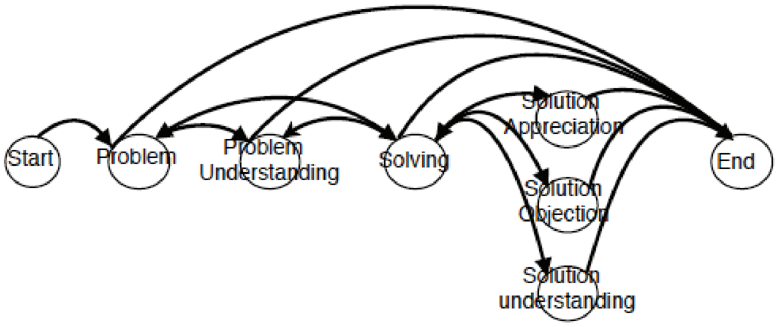


**Fig. 2.** State transition model for Q&A discussions

**Table 1.** A Q&A State Model: Definitions and Examples.

| State | Definition | Example | Count | Kappa |
|---|---|---|---|---|
| Problem (P) | Original problem is proposed by information seeker | I stuck in a very weird problem….. | 251 | 0.98 |
| Problem Understanding (PU) | 1.Providers ask related questions for understanding original question<br>2. Seekers answer the related questions and supply more details related to original issues. | 1.What kind of exception do you have?<br>2. It's seg fault afterwarods | 49 | 0.96 |
| Solving (S) | Information providers supply answer or suggestions for solving original question | You can try to reduce the memory | 447 | 0.99 |
| Solution Appreciation (SA) | Seekers solve the problem and acknowledge the help from providers | It works, Thanks. | 25 | 0.92 |
| Solution Objection (SO) | Seekers find the answer doesn't work and may ask for more help. | It doesn't work, any ideas? | 18 | 0.88 |
| Solution Understanding (SU) | Seekers may be confused about answer and ask questions for understanding. | What's the race condition, can you explain it? | 108 | 0.97 |

Note that not all of the messages containing the 'thank' words can be labeled as SA because some P messages can contain 'thanks' in advance before a solution is provided. In SO, the seeker or another participant objects or criticizes the answer proposed by a provider, as shown in Figure 1. SU may appear when the seeker fails to understand the solution and may ask for more information. M7 is an example. Note that it is hard to identify the difference between PU and SU only based on the content of the message because similar words may be used in both states. However, the context or the dialogue state of the message can help distinguishing the two. In Figure 2, we illustrate transitions among these states. P state can be followed by a PU as well as a S but its transition to a SA, a SO, or a SU is rare.

Table 1 presents a description of each state and examples. The state information is annotated manually and the last column shows the Kappa values for agreement between two annotators. The table also shows the distribution of the states. We can find that almost 50 percent of states belong to S. There is a small number of SOs.

**Table 2.** State transition matrix frequencies

| state | P | PU | S | SA | SO | SU |
|---|---|---|---|---|---|---|
| **P** | - | 14 | 220 | - | - | - |
| **PU** | - | 20 | 19 | - | - | - |
| **S** | 9 | 16 | 101 | 22 | 17 | 92 |
| **SA** | - | - | 4 | 4 | - | 3 |
| **SO** | - | - | 13 | - | - | - |
| **SU** | - | - | 90 | - | - | 10 |

Table 2 shows the frequency of state transitions. We can find that S is a bridge between the first two states and the last three states. The first two states (P and PU) discusses about the problem to be solved, while the last three are the feedback to the solution, and S connect the two parts. S dominates in the corpus. A S often directly follows a P, but there are cases where the Q&A process goes through a PU. Below we examine frequent patterns in the discuss process using the state information.

## 3. Automatic Discussion State Classification

236 threads and 899 posts are used for constructing the state transition model.

<u>Data preprocessing, normalization, and feature generation</u>

Student discussion data is highly noisy due to variances and informal nature of student written messages. The data pre-processing steps convert some of the informal expres-sions. For example, “yep”, “yeah” and “yea” are all substituted by “yes”. “what’s” and “wats” have to be converted to “what is”. The features for state classification are generated from (a) the message content, (b) neighboring messages, and (c) the mes-sage/author locational information:

-F1: n grams features within current message

-F2: position of the current message, such as the first message, the last message

-F3: position of participants, like the first author, the last author

-F4: n grams features within the previous message

-F5: position of the previous message

-F6: position of previous author

**Table 3.** Top 3 features for each state

| P | PU | S | SA | SO | SU |
|---|---|---|---|---|---|
| [get] unigram103_ NotFirst | [get] unigram103_NotFirst | 2ndAuthor | [correct] unigram197_Bottom | 1stAuthor | [fine] unigram330_ NotFirst |
| [somehow+delet] bigram65_ Any | [somehow+delet] bigram65_ Any | [get] unigram103_ NotFirst | [Cat _WH+should] bigram421_ Any | replyTo2ndMessage | [it+okai] PRE_bigram248_Bottom |
| 2ndAuthor | [about] PRE_unigram134_ Any | [somehow+delet] bigram65_ Any | [Cat _Subj_IWE+had] bigram581_Any | [give+Cat_Objective _IWE] bigram154_ NotFirst | [Cat _BE+wrong] PRE_bigram393_ NotFirst |

Given the full features generated from the content and the position, we use In-formation Gain [15] to reduce the features space. We select the 1620 features. The top 3 features for each state are shown in Table 3. Some of the features are n-grams from the current message or the previous message, e.g., a unigram CAT_ISSUE and a bigram not+sure. PRE represents feature from the previous message. Top/Bottom/Any/NotFirst represent position of the cue words in the message.

Linear CRF and other machine learning methods

Linear CRF [9] is a probabilistic model for characterizing the sequential data, referring the feature information, as well as the dependency among neighbors. The probability function are presented in equation (1) as follows,

$$\mathrm{p}\left({}^{Y}\!/_{X}\right) = \frac{1}{Z(x)} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right\} \qquad (1)$$

where Z(x) is an instance-specific normalization function, defined as equation (2),

$$Z(X) = \sum_{y} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right\} \qquad (2)$$

Y is the sequence data, X is the feature vectors with the total number K.   is a parameter vector, and the corresponding feature functions are defined as  .

In our application, each thread, containing several messages, is regarded as the sequence data. Linear CRF can capture the dependence among these messages, and give a most likely state transition with the purpose of characterizing each state for each message in a thread. We use Mallet [7] to create the model. Other machine learning methods such as SVM, decision tree, and logistic regression are widely used in practice. Since differences among states are rather clear and the data space is partitional, decision tree can build the model by separating feature space iteratively. SVM is also used as it is sensitive to the data points near the states' boundaries and has been suc-cessfully used for many problems. Logistic regression is another effective algorithm for categorical variables. Weka [10] was employed.

Resampling

We apply sampling methods due to unbalanced data. We split the six states as majority classes, including P, S and minority classes containing the rest four states. Because SVM, decision tree and logistic regression regard each message independently, resampling method can be applied directly by adding a copy of each minority instance.

As linear CRF rely on the message sequence, we separate threads as majority and minority classes. Majority thread can be defined as threads that have only P and/or S state, while minority threads include at least one message with other states: PU, SA, SO and SU. A combination of downsampling and upsamping methods is utilized for balancing the data and obtaining the better results; we reduce the majority threads by 30% and duplicate minority threads twice. For each classifier, we performed 10-fold cross-validation. In each fold, we separate data randomly, and use 80% for training data and 20% for test. Resampling is done for training data only.

Classification Results

**Table 4.** Classification Results

| | Precision/Recall/F-measure (%) | | | | | |
|---|---|---|---|---|---|---|
| **Model** | **P** | **PU** | **S** | **SA** | **SO** | **SU** |
| **Linear CRF** | 98.1/95.3/**96.7** | 32.0/20.6/**25.0** | 86.4/90.6/88.5 | 43.1/38.8/40.8 | 23.3/12.4/16.2 | 62.2/74.0/**67.5** |
| **SVM** | 100/93.8/**96.7** | 15.8/36.7/22.1 | 88.7/91.1/**90.0** | 42.1/63.0/**53.6** | 24.1/56.7/**31.2** | 53.8/90.6/**67.5** |
| **J48** | 99.6/94.1 | 10.1/28.7/15.8 | 83.0/89.0/85.2 | 22.5/48.8/29.1 | 10.8/23.3/14.3 | 47.6/80.1/59.5 |
| **LR** | 87.2/87.5/87.3 | 12.1/22.7/15.8 | 85.8/87.9/85.2 | 41.0/56.3/29.1 | 22.8/15.0/14.3 | 41.8/59.6/59.5 |

Table 4 shows precision, recall and F-measure scores for different classifiers. Linear CRF, SVM perform better than logistic regression and decision tree. It seems that the relation between states and features are not fully captured through a non-linear function directly. Although SVM and decision tree regard messages individually, both methods make use of dependencies among neighboring messages as some of the features capture previous message content and location information. Because of the small size for state PU, SA and SO, the precision and recall for these three states is low, especially for decision tree, which is sensitive for the features and instances. The precision and recall for state SA is relatively high. A possible reason is that its features include useful cue words including "thanks", "it works" that appear regularly. On the other hand, although we have 108 instances for state SU, the precision and recall for it is not so high. We may need further examples due to its variances. Another reason is that SU often contains a question for the solution, which may use similar key words as in P, thus it's challenging to completely distinguish SU from P.

## 4. Analyzing Q&A Process with State Information

Frequent dialogue patterns

We use the classified information in analyzing frequent state transitions and dialogue patterns. State transitions are represented as a sequence of three states: " previous state -> current state -> next state". We further distinguish contributions by the instructor and students. The end of discussion is labeled as "end". We list the top ten frequent patterns from 236 discussion threads in Table 5.

**Table 5.** The top ten frequent patterns for both instructor and students

| Instructor | | | Student | | |
|---|---|---|---|---|---|
| pattern | count | percent | pattern | count | percent |
| P->S->end | 88 | 13.31% | S->SU->S | 77 | 11.65% |
| P->S->SU | 36 | 5.45% | P->S->S | 33 | 4.99% |
| SU->S->end | 30 | 4.54% | S->S->S | 26 | 3.93% |
| S->S->end | 20 | 3.03% | P->S->end | 24 | 3.63% |
| SU->S->SU | 17 | 2.57% | SU->S->S | 16 | 2.42% |
| S->S->SU | 12 | 1.82% | S->SO->S | 13 | 1.97% |
| P->S->PU | 8 | 1.21% | S->S->end | 13 | 1.97% |
| PU->S->end | 8 | 1.21% | S->S->SU | 12 | 1.82% |
| S->S->S | 7 | 1.06% | S->SU->SU | 9 | 1.36% |
| SU->S->S | 6 | 0.91% | SU->SU->S | 9 | 1.36% |

The trends include:

a. Most SUs are generated by students.

b. The most frequent pattern for instructors is "P->S->end", and its frequency is much higher than the corresponding students' pattern. It indicates that instructor's answers can end many discussions, and may discourage further participation by the student.

c. If the previous state is P, most of the current states, generated either by the instructor or the students, is S. Instructor answers may be followed by SU: "P->S->SU", In contrast, students' S states tend to be followed by another S. That is, additional or different answers are proposed to students' answers more often than to instructors' answers.

d. If the previous state is SU, the instructor tends to post S, and the next state is often SU. Given a SU, students post either S or SU, and it can follow by another S.

e. If the previous state is S, students tend to post US, which is followed by a S. This is the most frequent pattern for students. Students can also post a S in response to S, which can be followed by another S. The second most frequent pattern is "S->S->S".

f. If the previous state is PU, the instructor tends to post a S. Students may post PU in response to a PU, which is followed by S or PU. Generally speaking, students may need more discussion turns to comprehend the problem.

Timing of responses

Table 7 lists frequent state transitions based on time information. "N/A" means that there is no such state transition in the instructor pattern. 'Instructor' columns represent time interval values when the current message is posted by the instructor. Likewise, 'Student' columns show time intervals when the current message is posted by a student. According to the Table 7, we can observe the following.

**Table 7.** Time interval for state transition

| Previous state ->Current state | Instructor | | Student | |
|---|---|---|---|---|
| | Median | Mean | Median | Mean |
| P->S | 4:38:39 | 7:56:37 | 1:55:11 | 5:29:28 |
| P -> PU | 3:36:7 | 6:23:6 | 3:9:58 | 3:37:16 |
| PU -> PU | 1:37:32 | 1:4:21 | 2:16:4 | 8:19:21 |
| PU -> S | 4:27:53 | 8:16:52 | 0:57:45 | 5:44:10 |
| S -> S | 5:25:58 | 8:49:43 | 1:34:26 | 5:41:39 |
| SU -> S | 4:22:19 | 8:10:59 | 1:18:58 | 3:22:22 |
| S -> SA | 1:4:37 | 4:30:39 | 0:45:54 | 2:17:21 |
| S -> SO | N/A | N/A | 1:55:21 | 5:2:18 |
| S -> SU | N/A | N/A | 1:59:2 | 9:1:9 |

1. From P state to S state, usually students spend less time in posting S than the instructor.

2. Student will spend less time to positively acknowledge (correct) answers. In other words, SA is quickly followed by a S. Transitions from S to SA, SO, and SU take a longer time. If the answer doesn't work, students may spend more time to check their problem and work.

3. The most time consuming state transition is when the instructor posts S in response to a S.

4. Usually, students reply messages more promptly than the instructor.

5. Resolved/Unresolved Discussion Classification

A discussion thread is 'resolved' when all the problems proposed by the participants, including initial problems, derived problems are solved successfully. Otherwise, it's unresolved thread. The features used for thread classification are:

-F1: n gram features within the final message in a thread

-F2: position of the final message, such as the first message, the last message

-F3: position of the final author, (the first author, the last author)

-F4: n gram features within the previous message

-F5: position of previous message and previous poster

-F6: state information

Table 8 presents the thread classification result. Comparing the three tables, we can conclude that state information indeed improve the performance of classifiers for thread classification. The state information represents the role of the message and effectively abstract low-level feature content or locational features. The state infor-mation also captures the dependencies among the messages within the whole thread, and can provide additional context information. For example, if the last state is PU, without state information, the message can be labeled as S for the understanding problems, and the classifier may label it as the resolved because it provides a solution. Generally, such abstractions provide better performance in machine classification when training data is not enough [15]. They also assist human analysis. The thread classification can help instructors in distinguishing resolved vs. unresolved discussions. Furthermore, state information helps instructors have insight on the process of discus-sion and facilitate them to understand the current state of the discussion. Such infor-mation supplies suggestions for instructors to decide when or whether to participate in the discussion.

**Table 8.** Precision, Recall and F-measure for thread classification

(a) Without state information

| | Resolved | | | Unresolved | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F-value** | **Precision** | **Recall** | **F-value** |
| **J48** | 0.92 | 0.94 | **0.93** | 0.71 | 0.66 | **0.68** |
| **SVM** | 0.87 | 0.98 | 0.92 | 0.81 | 0.39 | 0.52 |
| **LR** | 0.90 | 0.90 | 0.90 | 0.55 | 0.55 | 0.55 |

(b) With annotated state information

| | Resolved | | | Unresolved | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F-value** | **Precision** | **Recall** | **F-value** |
| **J48** | 0.95 | 0.99 | **0.97** | 0.94 | 0.75 | **0.84** |
| **SVM** | 0.90 | 0.98 | 0.94 | 0.85 | 0.50 | 0.63 |
| **LR** | 0.92 | 0.93 | 0.93 | 0.68 | 0.64 | 0.66 |

(c) With classified state information

| | Resolved | | | Unresolved | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F-value** | **Precision** | **Recall** | **F-value** |
| **J48** | 0.93 | 0.97 | **0.95** | 0.88 | 0.71 | **0.78** |
| **SVM** | 0.88 | 0.97 | 0.93 | 0.84 | 0.51 | 0.64 |
| **LR** | 0.91 | 0.90 | 0.90 | 0.64 | 0.66 | 0.65 |

## 5. Related work

There has been prior work on discussion analysis including use of speech act framework in modeling online discussions [3,4,5]. Some people focus on the roles that students play such as asking problems or answering other's questions [12,13]. Hidden Markov Model provides the framework for modeling the dialogue structure with hidden states [1,2,11]. They are closely related to our work, and we extend the existing framework by closely modeling the dialogue development and information exchange in Q&A discussions. In particular, we explicitly model problem and solution understanding phases as well as question and answer phases, and analyze the information exchange process using the state information. Graph-based approaches have been used in text mining, clustering and other related problems including labeling dialogue with tutors [1]. In order to facilitate the analysis of student discussions, we extend the existing work and represent a discussion thread as a graph model where each state in the model represents a message. There has also been work on machine classification of student online discussions [8,12,14] and results have been used to find meaningful dialogue patterns including features for critical thinking. Our work complements these results by closely examining and classifying Q&A processes.

## 6. Conclusion

We have presented a graph model for analyzing the discussion process and developed approaches for message state classification and thread characterization. The state information is used in analyzing frequent patterns and time intervals, and identifying different roles that instructors and students play in the Q&A process. Thread classifi-cation for resolved vs. unresolved problem is supported by the state information. As a next step, we plan to collect more data in order to obtain the more reliable classification result and explore additional improvement, including topic-based analysis of student problems. We plan to evaluate usefulness of the information with instructors.

## References

[1] Boyer, K. E., Ha, E. Y., Wallis, M. D., Phillips, R., Vouk, M. A., & Lester, J. C. (2009). Discovering tutorial dialogue strategies with hidden Markov models. Proc. AIED 2009.

[2] Chi, M., VanLehn, K., & Litman, D. (2011). An Evaluation of Pedagogical Tutorial Tactics for a Natural Language Tutoring System: A Reinforcement Learning Approach. IJAIED.

[3] Forbes-Riley, K., & Litman, D. (2009). Adapting to student uncertainty improves tutoring dialogues. Proceedings of International Conference on Artificial Intelligence and Education.

[4] Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, C., & Chen, L. (2010). Generating proactive feedback to help students stay on track. Proc. of ITS Conference 2010.

[5] Hirschberg, J. and Litman, D. (1993). Empirical Studies on the Disambiguation of Cue Phrases, Computational Linguistics, 19 (3).

[6] Kim, J., et al., (2006). Mining and assessing discussions on the web through speech act analysis. Workshop on Web Content Mining with Human Language Technologies at ISWC.

[7] McCallum, A. (2002). MALLET: a machine learning for language toolkit. http://mallet.cs.umass.edu

[8] McLaren, B. et al., Using Machine Learning Techniques to Analyze and Support Mediation of Student E-Discussions, Proceedings of AIED 2007.

[9] Sutton, C. (2006). An Introduction to Conditional Random Fields for Relational Learning.

[10] Witten, I.H., Frank, E.(2005). Data Mining: Practical machine learning tools and techniques. 2nd edn.

[11] Seo, S., Kang, J., Drummond, J. and Kim, J. (2011). Using Graphical Models to Classify Dialogue Transition in Online Q&A Discussions , Proceedings of AIED 2011.

[12] Mu, J., Stegmann, K., Mayfield, E., Rose, C., and Fischer, F. (2012). The ACODEA framework: Developing Segmentation and Classification Schemes for Fully Automatic Analysis of Online Discussions. Proc. CSCL 2012.

[13] Ravi, S. and Kim, J. (2007). Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers. Proceeding of AIED 2007.

[14] Rus, V., Graesser, A., Moldovan, C., and Niraula, N. (2012). Automatic Discovery of Speech Act Categories in Educational Games, Proceedings of EDM.

[15] Saeys, Y., Inza, I., and Larranaga. P. (2007). A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics, 23(19):2507-2517.

[16] Yoo, J. and Kim, J. (2012). Predicting Learners Project Performance with Dialogue Features in Online Q&A Discussions. Proceedings of ITS 2012.