Modeling Student Response Times: Towards Efficient One-on-one Tutoring Dialogues

Luciana Benotti
CONICET/Univ. Nacional de Cordoba
Cordoba, Argentina
benotti@famaf.unc.edu.ar

Jayadev Bhaskaran
ICME
Stanford University, USA
jayadev@stanford.edu

Sigtryggur Kjartansson
Department of Computer Science
Google/Stanford University, USA
sigkj@stanford.edu

David Lang
Graduate School of Education
Stanford University, USA
dnlang86@stanford.edu

Abstract

In this paper we investigate the task of modeling how long it would take a student to respond to a tutor question during a tutoring dialogue. Solving such a task has applications in educational settings such as intelligent tutoring systems, as well as in platforms that help busy human tutors to keep students engaged. Knowing how long it would normally take a student to respond to different types of questions could help tutors optimize their own time while answering multiple dialogues concurrently, as well as deciding when to prompt a student again. We study this problem using data from a service that offers tutor support for math, chemistry and physics through an instant messaging platform. We create a dataset of 240K questions. We explore several strong baselines for this task and compare them with human performance.

1 Introduction

One-on-one tutoring is often considered the gold-standard of educational interventions. Past work suggests that this form of personalized instruction can increase student performance by two standard deviation units (Bloom, 1984). Chatbots, intelligent tutoring systems (ITS), and remote tutoring are often seen as a way of providing this form of personalized instruction at an economical scale (VanLehn, 2011). However, their key limitation is that they are unable to identify when students have disengaged or are struggling with a task.

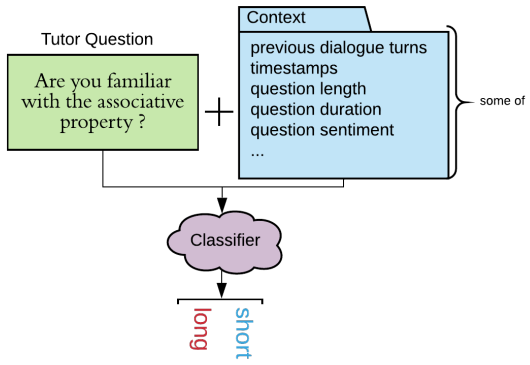

Figure 1: Diagram of our binary task definition. The classifier receives the tutor question text and dialogue contextual features such as the text and timing of previous dialogue turns, the duration and number of words in the question, the entrainment and sentiment between dialogue participants, among others.

Tutors and ITS need to calibrate how frequently and often they message their students. Prompting students too frequently could result in students feeling frustrated and disrupted, while prompting too slowly could result in students becoming disengaged or simply not learning as fast they could have with more prompting. This task is further complicated by the fact that interactions between students and a digital platform involve tasks of varying complexity and duration (such as performing a calculation, explaining a definition, or answering yes or no).

We propose predicting response latency of a tutor's question, as an indirect measure of a student's engagement (Beck, 2005) and question complexity (Strombergsson et al., 2013).

Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, pages 121–131
Brussels, Belgium, Nov 1, 2018. ©2018 Association for Computational Linguistics

The domain that we work with is tutoring session transcripts from an on-demand tutoring company, in which students take photos of their math, chemistry, and physics problems with their mobile or tablet devices. These images are then sent to a tutor in a remote location. Tutors and students then communicate back and forth over text messages until the student is able to solve the problem.

Specifically, the task that we focus on is: given a question from the tutor, predict whether it can be responded immediately or it is a question that requires more thought (see Figure 1). We formulate this task as a binary classification problem (short/long) whose inputs are the tutor's question and several dialogue contextual features. In this paper we make the following contributions:

- We define the task of modeling student response latency in order to make one-on-one tutoring dialogue more efficient.
- We have partnered with an Educational Technology company called Yup to produce one of the largest educational dialogue corpora to date. This initial dataset including over 18K tutoring sessions spanning 7K hours of text-based dialogue, including over 240K questions from tutors.
- We explore several strong classifiers for this task[1] whose performance is statistically significant better than expert human performance.

2 Related Work

Response time has been used as an indicator of student engagement (Beck, 2005) and performance (Xiong et al., 2011). These studies find that question response time is correlated with student's performance and engagement, and thus being able to predict a student's response latency is a useful measure for ITS. However, the task of predicting student response time to open-ended questions from tutors has not been addressed before. There is significant work in related topics such as response time analysis, dialogue automatic processing, sentiment analysis and education. Our problem lies within the intersection of these fields, so now we analyze a cross-section of past work and propose how their approaches, findings and analyses might be applicable to our situation.

[1]The source code is available at https://tinyurl.com/ybe65ctu.

To start with, Graesser and Person (1994) finds that analyzing question text is beneficial when assessing response characteristics, and this forms the basis for our bag-of-words baseline model. Moreover, Strombergsson et al. (2013) argue that the timing of past responses in a dialogue is correlated with the timing of future responses. Based on this study we propose our second baseline model trained only on how long it took students to respond to the previous turns in the dialogue.

Given these two baselines we investigate the following hypotheses, motivated by prior work from the different areas we mentioned:

H.1 One of the most interesting and counter-intuitive results in (Avrahami et al., 2008) is that the longer the message, the faster it was responded to. This is somewhat at odds with (Graesser and Person, 1994) which suggests that short questions elicit short responses. We plan to test the influence of question word count on predicting response time in our dataset.

H.2 We hypothesize that using as a feature the tutor's turn duration, which is the number of seconds elapsed between when the tutor first started talking and the tutor's question, will increase model performance (Strombergsson et al., 2013).

H.3 Moreover, (Avrahami et al., 2008) found that responsiveness to previous messages was highly correlated to the responsiveness of the current message, and therefore, considering messages prior to a question could prove useful when predicting response latency. We hypothesize that the content and the timing of dialogue turns that precede the question will increase the F_1 score of our model.

H.4 Brennan (1996) observes that while lexical variability is high in natural language, it is relatively low within a single conversation. When two people talk about the same topic, they tend to coordinate the words they use. This phenomenon is known as lexical entrainment. Thomason et al. (2013) show that prosodic entrainment has a positive effect on tutor-student dialogue rhythm and success. Nenkova et al. (2008) found that high frequency word entrainment in dialogue is correlated with engagement and task success. We test whether high frequency word entrainment has a significant impact on response

time prediction.

H.5 Previous work suggests that using sentiment information can help determine the level of engagement in MOOC forums (Wen et al., 2014). Wen et al. mine the sentiment polarity of the words used in forum posts in order to monitor students' trending opinions towards a course. They observe a correlation between sentiment ratio measured based on daily forum posts and number of students who drop out the course each day. Inspired by this work we hypothesize that the sentiment polarity of the words used in the tutor question might correlate with the student response time.

H.6 Finally, following previous work (Sutskever et al., 2014) we hypothesize that using sequential information (captured through a simple RNN) will improve the performance of response time prediction.

In Section 4 we explain how we design different experiments in order to test these hypotheses. But first, in the next section we describe our dataset.

3 Data

The dataset we are using consists of more than 7,000 hours of tutorial dialogues between students and tutors through an app-based tutoring platform. In total, there are 18,235 tutorial sessions in our dataset. These sessions are between 6,595 unique students and 117 tutors discussing mathematics, physics, and chemistry problems. A session has 61 turns in average, its median length is 34 turns.

TUTOR : I will be your instructor for this session. How far have you gotten in solving the problem? `short` (15 sec.)
STUDENT : I know b and d are rught
TUTOR : How do you know that? :) Can you show me your work? Can you show me your work? `long` (67 sec.)
STUDENT : Because graphed it and the y intercept was 01. Also it can't be a y intercept if it's not 0.

Figure 2: Sample Tutorial Dialogue. Student response times follow each tutor question.

Figure 2 is an excerpt of a tutoring session. It includes examples of two tutor questions and student responses, as well as the corresponding response time labels. Note that successive utterances have been concatenated in order to unify speakers that split their points into several lines (sometimes even breaking up an utterance into two or more lines) and those that include several utterances into the same turn. In this way we model the dialogue as a turn-based interaction such that two successive turns correspond to different speakers. Observe that in the second turn, the tutor utters three questions in one turn. The first question is open ended, the second question is a yes/no question and then he repeats the second question identically. In this case, when there is more than one question in the same turn, we use the timestamp of the first question. The rationale is that at that time the student could have started to formulate an answer. The follow up questions in this turn are refinements or repetitions to the first one motivated by the delay in the response.

The example dialogue in the figure also includes some typos and grammatical errors which illustrate the quality of the data. One of the features and key takeaways the reader should note is that there is a great deal of repetition in the types of questions that tutors ask. In particular, we identified a large number of duplicate questions that ask if a student is still engaged and understands a tutor's previous statement.

The raw data is preprocessed by:

- Sorting rows by session ID and timestamp.
- Removing incomplete rows.
- De-duplicating consecutive rows.
- Normalizing URLs in utterances.
- Tokenizing utterances using `spaCy` (Honnibal and Johnson, 2015).

As part of the project to model response time to tutor questions, we must first be able to distinguish them from other forms of conversation in tutorial dialogues. Past research suggests that humans can identify questions with high reliability (Graesser and Person, 1994). Given the size of our dataset, hand-coding the entire dataset seemed infeasible. As a proxy, we choose to identify tutor questions as any utterance which included a "?" character at the end of a sentence. This is done for three reasons. First, even if a third-party would contest whether or not a sentence is a question, a "?" symbol denotes a revealed preference on behalf of the speaker that anticipates a response. Second, even if a tutor accidentally mistypes the "?" symbol, a student may interpret it as a prompt to respond. Lastly, questions and elicitations may have

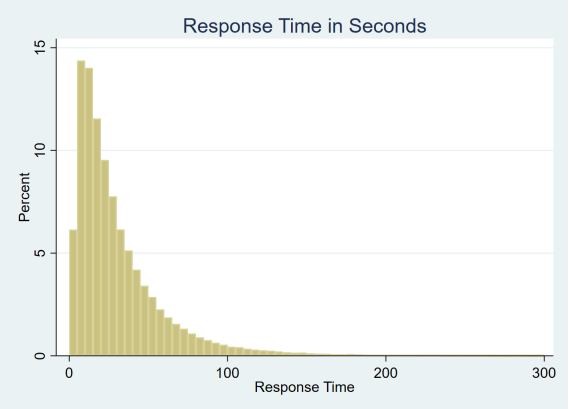

Figure 3: Response Time Histogram

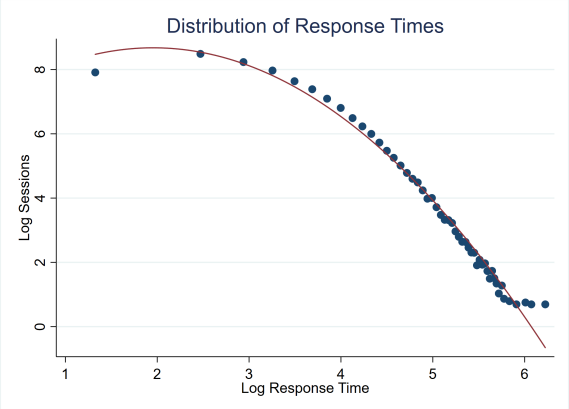

Figure 4: Log-Log Response Time Plots

very similar sentence structures but the "?" has a pedagogically distinct interpretation. Consider the statements "$3 \times 5 = 15$" versus "$3 \times 5 = 15$?" The former is an assertion of the fact and the latter is a form of assessment.

After extracting these candidate questions, we concatenate any surrounding tutor utterances, maintaining the original timestamp of the tutor question. That way if a tutor provides additional context before or after the question, it will be seen as part of the question.

Utilizing this rule, we identify a total of 242,495 questions. We then split sessions into train, dev, and test sets such that the train set comprises approximately 70% of all questions, the dev set comprises 15%, and the test set comprises 15% of questions, with the number of sessions split analogously. Figure 3 shows a histogram of student response times. The vast majority of responses occur within one minute.

The distribution of questions and response times appear to follow an approximate power law distribution (Figure 4). The associated R^2, proportion of variance explained, is 0.95, suggesting that this would be a reasonable approximating distribution.

4 Methodology

In this section, we first describe our approach to formulating the task as a classification problem, and the evaluation methodology that we adopt for measuring performance. Then we delve into the set-up for our experiments. Finally, we describe how we collect human performance for the task.

4.1 Classification Methodology

As we already mentioned, given a question from the tutor, our task is to predict whether the student can respond it right away or she will probably require more time. We cast this task into a binary (short/long) classification task whose inputs are the tutor's question and several dialogue contextual features. Response times are divided intro "short" (20 seconds or less), and "long" (more than 20 seconds). We use this threshold as it is the median response latency in our dataset. Using these thresholds, the classes are roughly divided in a 49/51 split (short/long).

We use weighted macro F_1 scores as our evaluation metric, train on the training set and tune our model parameters based on dev set results.

We propose three simple baselines for the task. We assess the performance of a random guessing baseline (guesses based on prior class distributions) and use this as a lower bound.

As an alternative baseline we use the counts of previous response time labels within session and train classifiers on this three-dimensional feature space. We implement two simple classifiers for our baseline using logistic regression and SVM (linear kernel) from `scikit-learn` (Pedregosa et al., 2011), along with weighted loss functions to account for class imbalance in the dataset. This baseline uses only temporal features, no textual features from the question or the context are used.

For our third baseline, we use only textual features from the tutor questions with a simple unigram bag-of-words model for feature representation, no temporal features are included. We use the same classifiers as above.

We implement these baselines knowing that they are too simple to capture the full complexity of this task. For example, questions such as "Are you still with me?" occur multiple times across the entire dataset, with highly varied response times

that depend on the context and state of the conversation. From our train set, approximately 12% of questions are duplicates and repeated questions frequently have different response times. As we argue in sections 5 and 6, it is necessary to look further into the context combining textual, temporal, as well as other types of information.

Below we describe how we enrich our best baseline with further features motivated by the hypotheses introduced in Section 2.

4.2 Experimentation

Our approach in testing the above hypotheses posed in Section 2 is setting forth experimental augmentations to the baselines introduced above, and evaluating the weighted F_1 scores across all classes in order to assess performance. In other words, we add a feature as a time and evaluate the F_1 score, as reported in Table 1. In this section, each experiment corresponds to the hypothesis with the same number.

For most of our experiments (excluding the RNN), we use both logistic regression and SVM on a bag-of-words model concatenated with respective additional feature(s) (e.g. question word length, question duration, etc.). For all experiments, we conduct a randomized hyper-parameter search for each model and pick the model that performs the best on the dev set.

Experiment 1: *Question Word Count*

Keeping in line with our first hypothesis, we add question word count as a feature along with the default bag-of-words features, to test if this improves the model performance.

Experiment 2: *Question Duration*

We add the temporal duration of each question as a feature within our feature space, and use this to test our second hypothesis.

Experiment 3: *Previous Dialogue Turns*

Modeling a dialogue as a turn-based interaction between the 'student' and the 'platform', we conduct two independent experiments enriching the question text feature space using a turn context window. The first experiment considers only the text of the previous turns, using a bag-of-words model per previous turn (distinguishing those turns that come from the tutor and those that come from the student). This is a simple model with only unigrams used in the bag-of-words model, we will explore more complex models in future work. The second experiment considers only the time in between turns (i.e. the cadence of the dialogue) in addition to the question text. For each of these experiments, we try different window sizes between 1 and 5, and pick the ones that performed the best.

Experiment 4: *Word Entrainment*

For word entrainment, we use the top 25/50/100 most frequent tokens across the corpus, as well as a set of predefined function words. The most frequent tokens may include punctuation symbols as well as function words. Previous work has found that successful dialogue partners align on exactly such tokens (Nenkova et al., 2008; Thomason et al., 2013). We calculate the occurrence of each of the relevant words (for the tutor and the student) over the 10 turns of dialogue prior to the given question, and compare the distributions for the tutor and the student. To compare distributions, we use cosine similarity (which is an intuitive measure of vector similarity) as well as Jensen-Shannon divergence, which has been used in prior work for comparing similarity in the textual domain (Goldberg et al., 2016).

Experiment 5: *Sentiment*

To determine question sentiment, we use the sentiment tagger from Stanford CoreNLP on the question text(Manning et al., 2014). This produces a 5-class sentiment score, ranging from very negative to very positive. For multi-sentence questions, we use the average sentence sentiment.

Experiment 6: *Recurrent Neural Networks*

Following previous work (Sutskever et al., 2014), we believe that using sequential information rather than a bag-of-words model would help improve performance. To test this, we train the simple recurrent neural network (RNN) depicted in Figure 5. As can be seen in the figure, a standard architecture was used with no attention mechanism and there is room for improvement.

The words from the tutor question are tokenized using a vocabulary of size $40,000$, padded to length 128 (99.9^{th} percentile), embedded in a 300-dimensional vector initialized with pre-trained fastText vectors (Joulin et al., 2017) and then fed it into an LSTM with hidden dimension 200. The encoded question is then fed into a

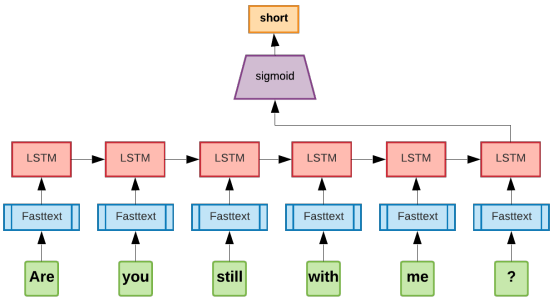

Figure 5: Diagram of Simple RNN on Question-Only Dataset.

densely-connected layer with a sigmoid activation function.

We train this model for a maximum of 20 epochs, optimizing the cross-entropy loss. We keep the model that performs best on the dev set. We achieve the best results after 5 epochs. We use `Keras` (Chollet et al., 2015) to run our deep learning experiments.

4.3 Human Ratings

Since this is a novel task, we additionally conduct an experiment that measures human performance on this task. This helps contextualize the performance of our models, and understand the relative ease/difficulty of this task for trained human experts. We assign three human raters to classify tutor questions. All raters are familiar with the tutoring platform and have been teachers in a classroom environment for several years. We ask the raters to evaluate under two setups. The first setup provides the question, the five turns of the dialogue previous to the question as well as all the turns times. The second context provides only the tutor's question as well as the student's response. The rationale for including the student's response is to understand how much this task depends on a feature that is not available in real-time (at the moment of predicting the response latency, the response is not available yet).

We give each rater 136 instances of 5-turn window questions including times (corresponding to the setup Q+T+L+D+X reported in Table 1) and 150 questions with their answers (Q+A in Table 1). Human agreement on this task is low, giving evidence that this is a difficult task for humans. In the Q+T+L+D+X experiment, Cohen's Kappa is only 0.14. In the question and answer model, Cohen's Kappa is substantially higher with a Kappa of 0.25. Human raters seem to be overwhelmed by too much contextual information, in particular in the Q+T+L+D+X setup. It is hard for people to pick up on the full range of predictive cues, some of which involve subtle behavioral patterns as we argue in the Section 6. Another possibility is that tutors used an availability heuristic in their prediction, low agreement may reflect the fact that tutors' predictions may be overly biased by their recent tutoring sessions (Kahneman and Tversky, 1973). Humans are not good at estimating time and unable to generalize beyond their own experience, computers can outperform them as we argue in the next section.

5 Results

We report best F_1 scores on the dev and test sets in Table 1. We organized the table in 4 parts. In this section we first give an overview of the results describing the rationale for their presentation in four parts, then we describe the results with respect to the hypotheses posed.

5.1 Overview and rationale

The first part of the table includes three simple baselines: a random prediction, an SVM trained using only the count of the labels of the previous questions in the dialogue, and an SVM trained using only the unigrams of the question text.

The second part describes our exploration of the feature space motivated by our hypotheses using SVM and logistic regression. See the table caption for an explanation of the acronyms used in the table. All tests that compares automatic models are paired t-tests because they are on the same datasets. All tests that compare human vs model are unpaired t-tests because the human evaluation was performed on a random subset of the dataset. The difference between the best performing model in this part (in boldface in the table) and the best baseline is statistically significant (.60 vs .62, paired t-test, p=0.05). Also the difference between the best model and the best human performance using these features is statistically significant (.62 vs .55, unpaired t-test, p=0.05).

In the third part, we add the question answer (A) as a single feature over the best performing baseline. Again the difference with the ques-

[2]Window size of 5 gave the best results.
[3]Window size of 5 gave the best results.

Model	F_1	
	Dev	**Test**
Baselines		
Random Classifier	0.50	0.50
Prev. Response Label Counts	0.58	0.57
Question Text (Q)	0.60	0.60
Feature exploration with SVM/LR		
Q + Question Length (L)	0.61	0.61
Q + Question Duration (D)	0.61	0.61
Q + Prev. Turns Texts[2] (X)	0.61	0.60
Q + Prev. Turns Times[3] (T)	0.62	0.61
Q + Word Entrainment (E)	0.60	0.60
Q + Question Sentiment (S)	0.60	0.60
Q+T+L+D+X	**0.63**	**0.62**
Human 1 with Q+T+L+D+X	–	0.50
Human 2 with Q+T+L+D+X	–	0.43
Human 3 with Q+T+L+D+X	–	0.55
Answer addition with SVM/LR		
Q + Answer (A)	**0.67**	**0.67**
Human 1 with Q+A	–	0.53
Human 2 with Q+A	–	0.63
Human 3 with Q+A	–	0.62
Baseline Neural Model		
RNN with Question Text (Q)	**0.62**	**0.62**

Table 1: Results comparing simple baselines, feature exploration using Logistic Regression/SVM, human performance, and a baseline using an RNN. L: Question length in number of words. D: Question duration in seconds (a question may span more than one turn). X: The text of the dialogue turns preceding the question. T: The timestamp of the dialogue turns preceding the question. E: Word entrainment between tutor and student. S: Sentiment analysis on the question.

tion only baseline is statistically significant (.60 vs .67). Furthermore, the difference with the Q+T+L+D+X is statistically significant (.62 vs .67), showing that the answer is useful for this task as argued in the error analysis. As in the earlier part, the difference between this model and the best human performance (for these features) is statistically significant (.63 vs .67).

In the fourth part, we include the results with RNN and compare it with the best baseline which uses the same features: the question text baseline. Also here the difference between the two is statistically significant (.60 vs .62).

We find that for both SVM and logistic regression classifiers the best performance is obtained with L2 penalties. For the SVM, squared hinge loss is found to work better than hinge loss. We find no significant difference in performance between SVM and logistic regression on this dataset.

5.2 Hypotheses analysis

Below we analyze what these results mean for our hypotheses.

Experiment 1: *Question Word Count*

Adding question length as a feature improves performance, validating H.1. Furthermore, longer questions (which usually involve a lot of technical information in the form of formulae/equations, or are an aggregation of repeated questions) tend to result in higher response times. This is contrary to results seen in (Avrahami et al., 2008), but in line with those seen in (Beck, 2005). These results potentially indicate behavioral differences between the two domains - instant messaging (Avrahami et al., 2008), and tutorial dialogue in virtual environments (Beck, 2005); the latter also being the domain of our current work.

Experiment 2: *Question Duration*

We notice similar trends while analyzing question duration as a feature. Using question duration along with bag-of-words features helped boost model performance, verifying H.2. Intuitively, this feature seems to be an indicative measure of question complexity, and longer duration questions result in higher response times.

Experiment 3: *Previous Dialogue Turns*

H.3 is a mixed bag. We start off by adding different spans of previous dialogue turn text. This helps improve performance on the dev set but does not add anything over the baseline when evaluated on the test set, suggesting that these features do not generalize well across conversations. On the contrary, adding previous dialogue times helps improve model performance in both the dev and test sets. In both settings, we find the best results while using 5 turns of previous dialogue.

Experiment 4: *Word Entrainment*

Word entrainment seems to have no effect on model performance. There are no significant differences based on the set of words used to measure entrainment (function words or 25/50/100 most frequent words), as well as the metric of lexical

distance (Jensen-Shannon divergence/cosine similarity). Therefore, we cannot confirm the validity of H.4 in our setting.

Experiment 5: ***Sentiment***

A similar narrative is observed with sentiment (H.5). We note that sentiment analysis is less accurate when sentences get longer, and this might be one of the causes for the relative ineffectiveness of sentiment as a feature. Another possible interpretation is that this text is not aligned well with traditional definitions of sentiment. Many terms in mathematics are neutral but are classified with negative sentiment on a high valence. In future work we plan to explore the use of sentiment analysis on student generated text rather than on tutor questions.

Experiment 6: ***Recurrent Neural Networks***

The results of using deep learning models (RNN) are promising (H.6). The RNN achieves a performance which is statistically significant better than the baseline with the same feature: only the question text. A probable reason is that the baseline uses unigrams, hence it loses the order among the words of the question while the RNN model might benefit from this information. It must be noted that we have not performed extensive hyperparameter tuning, performance might be further improved with more hyperparameter tuning.

6 Analysis

In spite of the fact that the results presented in the previous section are above human performance, we believe that the automatic performance for this task can outperform humans even more. Therefore we perform a detailed qualitative error analysis in this section.

We focus this section on error analysis of one of the best performing models which does not include information about the answer: Q+T+L+D+X. We do not include the answer in this analysis in order to understand why this feature alone makes a significant difference. Also, the answer information would not be available to the model in an application trying to predict student response latency in real time.

There are two kinds of errors for our task. One kind corresponds to the case in which the model overestimates how long it will take the student to respond, and the other to cases in which the model underestimates the latency of the response. We perform a manual error analysis over both types of errors, we describe our findings below.

6.1 Overestimation errors

First, we find that the model overestimates the response time to tutor questions that exhibit some positive politeness strategy (Brown and Levinson, 1987). In many of the overestimated instances analyzed the tutor uses lexical in-group markers. These can include altering the forms of address and using in-group language. For instance, the use of “we" instead of “you", as in the example below, is a kind of in-group marker. Other kinds of in-group language include the use of dialects, jargon or slang, and linguistic contractions. The following is an example of a linguistic contraction, an inclusive pronoun and a smiley, all signs of positive politeness. The label predicted by the model for this example is long and the actual latency is short: “I’ll show you how we can find the other angle of this square :). Is this diagram from your textbook?”. Also, the following positive politeness strategies are found in overestimated instances. A speaker may seek to avoid disagreement by employing a token of agreement or appreciation, or a confirmation question such as in the example “Awesome! We can use a number line to solve the problem. Was that clear?.”

Second, the model also overestimates the response time to tutor questions that include some negative politeness strategy. Whereas positive politeness enhances the hearer’s positive self-image via social appreciation, negative politeness addresses the hearer’s need for freedom from imposition. A useful strategy here is the use of hedges. A hedge is a softening of a statement, induced by employing less-than-certain phrasing such as would, might, or should as illustrated in the example below. Further efforts to avoid encroaching on the hearer’s sense of freedom include impersonalizing the speaker and hearer. Common strategies here include using passive and circumstantial voices as in the following example “It would be best to clarify that the math operation that should be applied to solve this problem is addition. Does that make sense?.”

Third, the model overestimates also when the student turn following the tutor question is not actually a response but a positive acknowledgement (e.g., “Ok, let me see”) or a clarification request (e.g., “the variable is the value?”).

6.2 Underestimation errors

First, we find that the model frequently underestimates the time required for the student response, i.e. the answer is slower than predicted when there is some sort of face threatening act (Brown and Levinson, 1987) that the tutor or student is doing, either by disagreeing (for instance, with the words no, nope, not really) or by some inappropriate behavior. For example: "Question: Do these questions belong to a graded test or quiz?" Answer: "it a quiz, just making sure I'm on the right path." Consistently, face preserving words such as "sorry" are also sometimes present in questions form the tutor that take longer to respond than predicted.

Second, the model also underestimates the response latency of questions that the student avoids answering such as "Um I'm not sure" and "Can u just help me pls I'm in a rush" and "Just give me the answer."

Third, indirect speech acts such as "Do you spot one more solution that does not lie in the domain?" which syntactically require a yes/no answer but pragmatically implicate a request for action, are also underestimated.

Finally, there are also whole sessions where the model underestimates the response time for every question. This may be an indicator than some students are just slower to respond.

In conclusion, the feature space could be improved modeling different politeness strategies (Danescu et al., 2013; Benotti and Blackburn, 2016), including features about whether the most probable response for this kind of question is an answer, an acknowledgement or a clarification request (Benotti and Blackburn, 2017; Rao and Daume, 2018) as well as features about indirect speech acts and implicatures (Benotti and Traum, 2009; Jeong et al., 2009). These three areas are challenging aspects of natural language understanding and interaction modeling but there is encouraging work being done in each of them which we plan to take as starting points to pursue this interesting task further.

7 Conclusion & Future Work

To summarize, this experimental paper comprises several tasks. First, we introduce a new dataset of tutorial dialogue in a mobile tutoring environment with automatically annotated tutor questions and student responses. Secondly, we formally define the task of predicting student response times to tutor questions. Knowing whether a student can respond a given question immediately or it normally requires more thought, would help tutors optimize their own time as well as prompt the student at the right moment. Thirdly, we develop a set of models and explore our hypotheses related to hand-built feature functions and model classes by making experimental augmentations to the baselines. Lastly, we evaluate the performance of trained human experts on the same problem. We conclude that this is a difficult task, even for human beings; while these models are able to outperform humans, further research is required.

We plan to experiment with situational metadata such as tutor and student identity and gender, subject of study and nature of payment system (free trial, pay per minute, pay per month usage). A promising direction for further work is modeling the politeness strategies as well as the other features mentioned in our error analysis. We believe that this enriched feature space can result in a model that outperforms human experts even more.

Acknowledgements

We would like to thank professors Chris Potts and Bill MacCartney from Stanford University for their timely help and guidance at various junctures, and the anonymous reviewers for their suggestions to make the paper better. The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant (#R305B140009) and the Argentinean Research Council through grants PICT-2014-1833, PDTS-CIN-CONICET-172 and an external visiting scholar fund from CONICET. We would also like to Thank Yup.com for providing us with an anonymized version of the data.

References

Daniel Avrahami, Susan Fussell, and Scott Hudson. 2008. Im waiting: Timing and responsiveness in semi-synchronous communication. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW)*. ACM, New York, NY, USA, pages 285–294.

Joseph Beck. 2005. Engagement tracing: Using response times to model student disengagement. In *Proceedings of the 2005 Conference on Artificial Intelligence in Education*. IOS Press, Amsterdam, The Netherlands, pages 88–95.

Luciana Benotti and Patrick Blackburn. 2016. Polite interactions with robots. *Frontiers of Artificial Intelligence and Applications* 290:293–302.

Luciana Benotti and Patrick Blackburn. 2017. Modeling the clarification potential of instructions: Predicting clarification requests and other reactions. *Computer Speech and Language* 45(C):536–551.

Luciana Benotti and David Traum. 2009. A computational account of comparative implicatures for a spoken dialogue agent. In *Proceedings of the Eighth International Conference on Computational Semantics*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 4–17.

Benjamin S Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher* 13(6):4–16.

Susan Brennan. 1996. Lexical entrainment in spontaneous dialog. In *Proceedings of the 1996 International Symposium on Spoken Dialogue (ISSD)*. Acoustical Society of Japan, Philadelphia, PA, pages 41–44.

Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some universals in language usage*. Cambridge University Press.

François Chollet et al. 2015. Keras. https://keras.io.

Cristian Danescu, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. pages 250–259.

Amir Goldberg, Sameer B. Srivastava, V. Govind Manian, William Monroe, and Christopher Potts. 2016. Fitting in or standing out? The tradeoffs of structural and cultural embeddedness. *American Sociological Review* 81(6):1190–1222.

Arthur Graesser and Natalie Person. 1994. Question asking during tutoring. *American educational research journal* 31(1):104–137.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1373–1378.

Minwoo Jeong, Chin-Yew Lin, and Gary Geunbae Lee. 2009. Semi-supervised speech act recognition in emails and forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1250–1259.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pages 427–431.

Daniel Kahneman and Amos Tversky. 1973. On the psychology of prediction. *Psychological review* 80(4):237.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL)*. pages 55–60.

Ani Nenkova, Agustín Gravano, and Julia Hirschberg. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (ACL-HLT)*. pages 169–172.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Sudha Rao and Hal Daume. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Sofia Strombergsson, Anna Hjalmarsson, Jens Edlund, and David House. 2013. Timing responses to questions in dialogue. In *Proceedings of InterSpeech*. pages 2584–2588.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. MIT Press, Cambridge, MA, USA, NIPS'14, pages 3104–3112.

Jesse Thomason, Huy V. Nguyen, and Diane Litman. 2013. Prosodic entrainment and tutoring dialogue success. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 750–753.

Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* 46(4):197–221.

Miaomiao Wen, Diyi Yang, and Carolyn Penstein Rosé. 2014. Sentiment analysis in MOOC discussion forums: What does it tell us? In *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*. pages 130–137.

Xiaolu Xiong, Zachary A. Pardos, and Neil T. 2011. An analysis of response time data for improving student performance prediction. Unpublished.