Predicting Decisions in Language Based Persuasion Games

Reut Apel[1], Ido Erev[1], Roi Reichart[1], and Moshe Tennenholtz[1]

[1]Industrial Engineering and Management, Technion—Israel Institute of Technology
reutapel@campus.technion.ac.il, {erev,roiri,moshet}@technion.ac.il

Abstract

Sender-receiver interactions, and specifically persuasion games, are widely researched in economic modeling and artificial intelligence, and serve as a solid foundation for powerful applications. However, in the classic persuasion games setting, the messages sent from the expert to the decision-maker are abstract or well-structured application-specific signals rather than natural (human) language messages, although natural language is a very common communication signal in real-world persuasion setups. This paper addresses the use of natural language in persuasion games, exploring its impact on the decisions made by the players and aiming to construct effective models for the prediction of these decisions.

For this purpose, we conduct an online repeated interaction experiment. At each trial of the interaction, an informed expert aims to sell an uninformed decision-maker a vacation in a hotel, by sending her a review that describes the hotel. While the expert is exposed to several scored reviews, the decision-maker observes only the single review sent by the expert, and her payoff in case she chooses to take the hotel is a random draw from the review score distribution available to the expert only. The expert's payoff, in turn, depends on the number of times the decision-maker chooses the hotel.

We consider a number of modeling approaches for this setup, differing from each other in the model type (deep neural network (DNN) vs. linear classifier), the type of features used by the model (textual, behavioral or both) and the source of the textual features (DNN-based vs. hand-crafted). Our results demonstrate that given a prefix of the interaction sequence, our models can predict the future decisions of the decision-maker, particularly when a sequential modeling approach and hand-crafted textual features are applied.[1]

1 Introduction

The modeling and analysis of sender-receiver interactions are central to both economic modeling and Artificial Intelligence (AI). Indeed, the 2001 Nobel prize in economics was presented to Akerlof, Spence, and Stiglitz, for their pioneering research lines, showing how the signaling of information can alter strategic interactions (see Spence, 1973). Of particular interest is the study of cheap talk (Crawford & Sobel, 1982), and Bayesian persuasion (Kamenica & Gentzkow, 2011, following Aumann, Maschler, & Stearns, 1995), dealing with different levels of commitment power of the sender. The settings of personalized advertising and targeted recommendation systems (Shapiro & Varian, 2008; Emek, Feldman, Gamzu, PaesLeme, & Tennenholtz, 2014), where different services

[1]Our code and data will be uploaded within a few days from the submission date, to the following URL: https://github.com/reutapel/Predicting-Decisions-in-Language-Based-Persuasion-Games

are offered to potential customers, are forms of strategic sender-receiver interactions in the spirit of these stylized economic models (Arieli & Babichenko, 2019). In AI, the whole agenda of chat-bots design is targeted at sender-receiver interactions, emphasizing the use of language (Jurafsky & Martin, 2018).

Most economic models of sender-receiver interactions are game-theoretic ones. In such a setting, both sender and receiver are strategic players with their own private utilities. They, however, possess asymmetric information: Typically, the sender has more information than the receiver about the state of nature. Of particular interest is the persuasion game setting, where the sender's objective is to persuade the receiver to select some course of action among a set of possible actions. The receiver, in turn, has different payoffs for the different actions. While the receiver's payoff depends on the state of nature, he has a higher level of uncertainty about the state of nature than the sender has. The study of this fundamental setting has received a significant amount of attention in recent years and serves as a solid foundation for powerful applications (Emek et al., 2014, Bahar, Smorodinsky, & Tennenholtz, 2016). In what follows, we refer to the sender as an *expert* and to the receiver as a *decision-maker*.

In this work, we adopt the above framework of persuasion games and consider a setting of repeated expert-decision-maker games. Hence, while there is no notion of commitment by the expert in our setting, there are definitely possible reputation effects (Kreps, Milgrom, Roberts, & Wilson, 1982). Bridging this foundational setting with language-oriented AI, our work introduces for the first time the use of natural language into these stylized persuasion games. Namely, while in the classical setting, the expert's messages to the decision-maker are abstract or well-structured application-specific signals (as nicely implemented also in AI settings: Azaria, Rabinovich, Kraus, Goldman, & Gal, 2012, Azaria, Rabinovich, Kraus, & Goldman, 2011), in our setting these messages are expressed in natural language.

To be more precise, consider a setting where an expert has to select among a safe constant-payoff action and a risky action with the payoff determined by some probability distribution unknown to the decision-maker. The expert aim is for the decision-maker to select the risky action. He can do it by communicating messages, where each message is associated with a different payoff in support of the distribution. The interaction is repeated, where at each interaction, a different distribution is selected, and different messages are associated with the different payoffs. The messages and their relationship to payoffs are grounded in some real-world events, e.g., messages associated with corresponding numeric grades of hotel reviews. The main questions we ask: Given some game's history in the first K trials of interaction, can we predict behavior in the game's subsequent trials? What is the best way to come up with such a prediction?

The approach we have taken to tackle the above challenge is as follows. We created a data set that was collected using an online experiment. In the experiment, two participants are randomly and anonymously paired, and each of them was randomly selected to one of two roles: Decision-maker and expert. They then play a ten trial game together, where at each trial the expert is asked to select one of seven hotel reviews presented to her alongside their related scores. The chosen review was then presented to the decision-maker without its numerical score, and the decision-maker was asked to choose between the risky hotel and the safe stay home options. The expert benefits from hotel choice by the decision-maker, while the decision-maker's payoff in the hotel choice case was determined by the score distribution presented only to the expert.

Given our data, we are interested in predicting the decision-makers' decisions, which determine the experts' payoffs. Notably, we define the following research questions: (1) Given the history of the first pr trials, can we predict the decision made by the decision-maker in each ensuing trial? (2) Given the history of the first pr trials, can we predict the subsequent trials' hotel choice rate? (3) Which modeling strategy would be best for our tasks: a non-structured classifier, a sequence model, or an attention-based approach? (4) Which textual features would serve our prediction model the most? Should we focus on Deep Neural Networks (DNNs) based features? Or can we also gain from hand-crafted features? And (5) Which aspects of the data are crucial for our prediction? Should we only consider the impact of the textual messages or also consider the decision-maker's behavior throughout the game?

To answer our questions, we explore different modeling strategies along three lines: (a) sequential vs. non-sequential models (which also touches on DNN based models vs. more traditional linear models); (b) text-based features learned by DNNs vs. hand-crafted features, and (c) text-based features only, as well as their combination with behavior-based features.

We found the answers to the above questions, both encouraging and illuminating. It turns out that such action prediction in language-based persuasion games can be done effectively. Moreover, the best way to do so, is by using a mixture of a feature-based approach and a sequence neural approach. Namely, rather than learning features from the text in a plain neural approach or applying a more classical feature-based approach, we show that DNNs using relevant features allow us to obtain high-quality predictions, outperforming the baselines and the other approaches. Another intriguing observation is that sequence models outperform the non-sequence models.

The rest of the paper is organized as follows. Section 2 discusses previous work. While we are the first to address the task at hand to the best of our knowledge, we survey previous work on action prediction in machine learning, natural language processing for text-based prediction, and argumentation in multi-agent systems. Section 3 defines our task, including the game definition, our prediction tasks, and how we represent our data. Section 4 describes our data, including the data collection procedure, and provides a qualitative analysis. Section 5 describes our modeling approach, including the algorithmic details of our sequential and non-sequential models, as well as the behavior and textual features. Section 6 provides the details of our experiments, including the baseline models to which we compare our approach and the evaluation measures. Finally, Sections 7, 8 and 9 discuss our results, an ablation analysis, and the derived conclusions.

2 Related Work

We recognize three lines of work that are related to our research. This section discusses the previous works in these lines, highlights the differences between them, and summarizes our novel contributions in light of those works.

2.1 Action Prediction in Machine Learning

Previous work successfully employed Machine Learning (ML) techniques in service of action prediction in an ensemble of games. The first work we are aware of to use ML techniques for action prediction in one-shot strategic-form games is Altman, Bercovici-Boden, and Tennenholtz (2006). This work focuses on the learning of the choices made by individuals in a given game, based on population behavior in the game ensemble and the choices of the particular individual of interest in the other games. Interestingly, this approach defeats in that context leading experimental economics procedures based on cognitive models (Camerer, Ho, & Chong, 2004; Costa-Gomes, Crawford, &

Broseta, 2001). Hartford, Wright, and Leyton-Brown (2016a) has demonstrated that DNN models trained on an ensemble of games can outperform models based on cognitive hierarchies. Plonsky, Erev, Hazan, and Tennenholtz (2017) have shown how psychological features can be integrated with ML techniques in order to predict human choices. The address games against nature (i.e., the choice among gambles), which are common in the psychology literature.

Overall, we identify two groups of previous work, differing in the settings they address. The first group consists of works that try to predict individuals' behavior (see, e.g., Altman et al., 2006, and the references therein). They represent individuals by their play in several labeled games, where all the individuals have previously played the same games . They then predict the behavior of that individual in a new, unseen game. The works in the other group are not concerned with predictions about the behavior of specific individuals, but, instead, every data point is a choice problem, e.g., a selection between two lotteries encoded by probabilities and rewards, and its label is the population statistics (e.g., Hartford, Wright, & Leyton-Brown, 2016b; Plonsky et al., 2017).

In the setting we address in this paper, we aim to predict the outcome in a new game, given information about the behavior of other players in similar (although not necessarily identical) games. Since this game is a multi-stage game, we aim to predict both the average suffix rewards and the reward in each sub-game in the suffix of the game, given the observed behavioral prefix of that game. Beyond this difference in the game-theoretic setup, our emphasis is on introducing organic linguistic texts' strategic use in the stylized game-theoretic interaction.

2.2 Natural Language Processing for Text-based Prediction

Text-based prediction tasks are ubiquitous in the Natural Language Processing (NLP) and ML literature. The most basic tasks address the predictions of properties of the text itself, including its author, topic, and sentiment (Joachims, 1999; Pang, Lee, & Vaithyanathan, 2002; Steyvers & Griffiths, 2007; Pang & Lee, 2008; Gill, Nowson, & Oberlander, 2009). Text-based prediction have also been applied to variables that are not directly encoded in the text. One example is the prediction of what people are likely to think about a given text, e.g., predicting the citation count of a scientific paper (Yogatama, Heilman, O'Connor, Dyer, Routledge, & Smith, 2011; Sim, Routledge, & Smith, 2015). Another, more ambitious, attempt is drawing predictions about real-world events based on texts that discuss related information (Smith, 2010). Examples of this line of work include future movie revenues based on its textual reviews (Joshi, Das, Gimpel, & Smith, 2010), predicting risk from financial reports (Kogan, Levin, Routledge, Sagi, & Smith, 2009), and predicting election outcomes from related tweets (O'Connor, Balasubramanyan, Routledge, Smith, et al., 2010).

Another strand of the literature on text-based prediction related to our efforts is predicting the future actions of the authors of given texts. For example, Niculae, Kumar, Boyd-Graber, and Danescu-Niculescu-Mizil (2015) tried to predict actions in an online strategy game based on the language produced by the players as part of the inter-player communication required in the game. In Ben-Porat, Hirsch, Kuchi, Elad, Reichart, and Tennenholtz (2020), the authors predict an individual's action in a one-shot game based on the free text he/she provides while being unaware of the game to be played. In another study, Oved, Feder, and Reichart (2020) tried to predict NBA players' in-game actions based on their open-ended interviews. However, one key difference between these tasks and our task is that in our study we aim to predict the future actions of the decision-maker who reads and uses the text although he did not produce it. Therefore, the only information we have about this decision-makers is her previous decisions. Another key difference that poses a greater challenge in our case is that we aim to predict a decision sequence, while these previous tasks did not have a sequential element.

Recently, several works studied the connection between natural language and persuasion (Persing & Ng, 2017; Carlile, Gurrapadi, Ke, & Ng, 2018). Wang, Shi, Kim, Oh, Yang, Zhang, and Yu (2019) collected a persuasion dialogue data set and predicted the persuasion strategies used in the corpus. Chatterjee, Park, Shim, Sagae, and Morency (2014) predicted speakers' persuasiveness on housing videos of product reviews using verbal descriptors and para-verbal markers of hesitation. Yang, Chen, Yang, Jurafsky, and Hovy (2019) focused on advocacy requests and proposed a neural network that quantifies persuasiveness and identifies persuasive strategies. In another work, Shaikh, Chen, Saad-Falcon, Chau, and Yang (2020) examined how strategy orderings contribute to persuasiveness in a loan requests data set. In contrast to these works, our work focuses on a repeated persuasion game setting, in which the expert strategy is long term, and her choice in a specific trial affects both the outcome in this trial and her reputation for the rest of the game. Another difference is that in this work we focus on the decision-makers choices, in contrast to these previous works which focus on persuasion strategies.

2.3 Argumentation in Multi-agent Systems

The multi-agent systems community has also conducted research related to our work, dealing with argumentation (Walton, 2009) and negotiation (Kraus & Arkin, 2001). Particularly, improving prediction of persuasive arguments (Azaria et al., 2012; Azaria et al., 2011; Tan, Niculae, Danescu-Niculescu-Mizil, & Lee, 2016; Rosenfeld & Kraus, 2016) has yielded significant progress in argumentation research. Moreover, research into automated negotiations has trained automated agents to exploit human choice prediction (Peled, Gal, & Kraus, 2012; Rosenfeld & Kraus, 2018).

Our approach is complementary since its focus is on the task of persuasion through the use of organic linguistic texts; This is carried out in multi-stage persuasion games, extending the economics literature. We study a fundamental aspect of persuasion: Can we predict an expert (persuader) reward (i.e., the decision-maker's decisions) who aims to convince a less informed decision-maker to adopt risky alternatives while using linguistic texts? Our prediction is based only on the behavior in the prefix of the interaction between the expert and the decision-maker, the texts the decision-maker observes, and information about other experts and decision-makers' plays in different situations.

3 Task

In this section we present the main ideas of this work, the challenges, and our research questions. We aim to predict decisions within a setting of repeated expert-decision-maker persuasion games. This setting raises some challenges.

The expert in our setting observes in each trial seven reviews and their scores, and she knows the decision-maker's payment will be one of these scores. Hence, she has to decide what message (review) to send the decision-maker in order to maximize her own total payoff. This situation raises questions like: What would be a good strategy here? Should I communicate the expected payoff or should I present another statistic? The repeated aspect adds complexity, as the expert's choice in a specific trial affects not only the decision-maker's decision in this trial but also the expert's reputation for the rest of the game.

Using verbal communication introduces additional challenges. Verbal phrases add information across numerical estimates, but can also increase confusion because people interpret verbal terms in different ways (Beyth-Marom, 1982; Budescu, Weinberg, & Wallsten, 1988). For the same reason, verbal communication can also increase dishonesty, and hence may increase the experts' tendency to select inflated evaluations.

Many questions can be asked regarding this setting. This paper focuses on the decision-makers' choices, since these decisions determine the outcome for both the expert and the decision-maker (although for the decision-maker, this is not the final payoff). Particularly, we focus on the following questions:

1. Given the history of the first pr trials, and the texts shown to the decision-maker in the subsequent trials, can we predict the decision made by the decision-maker in each ensuing trial?

2. Given the history of the first pr trials, and the texts shown to the decision-maker in the subsequent trials, can we predict the subsequent trials' hotel choice rate?

3. Which modeling strategy would be best for our tasks: A non-structured classifier, a sequence model or an attention-based approach?

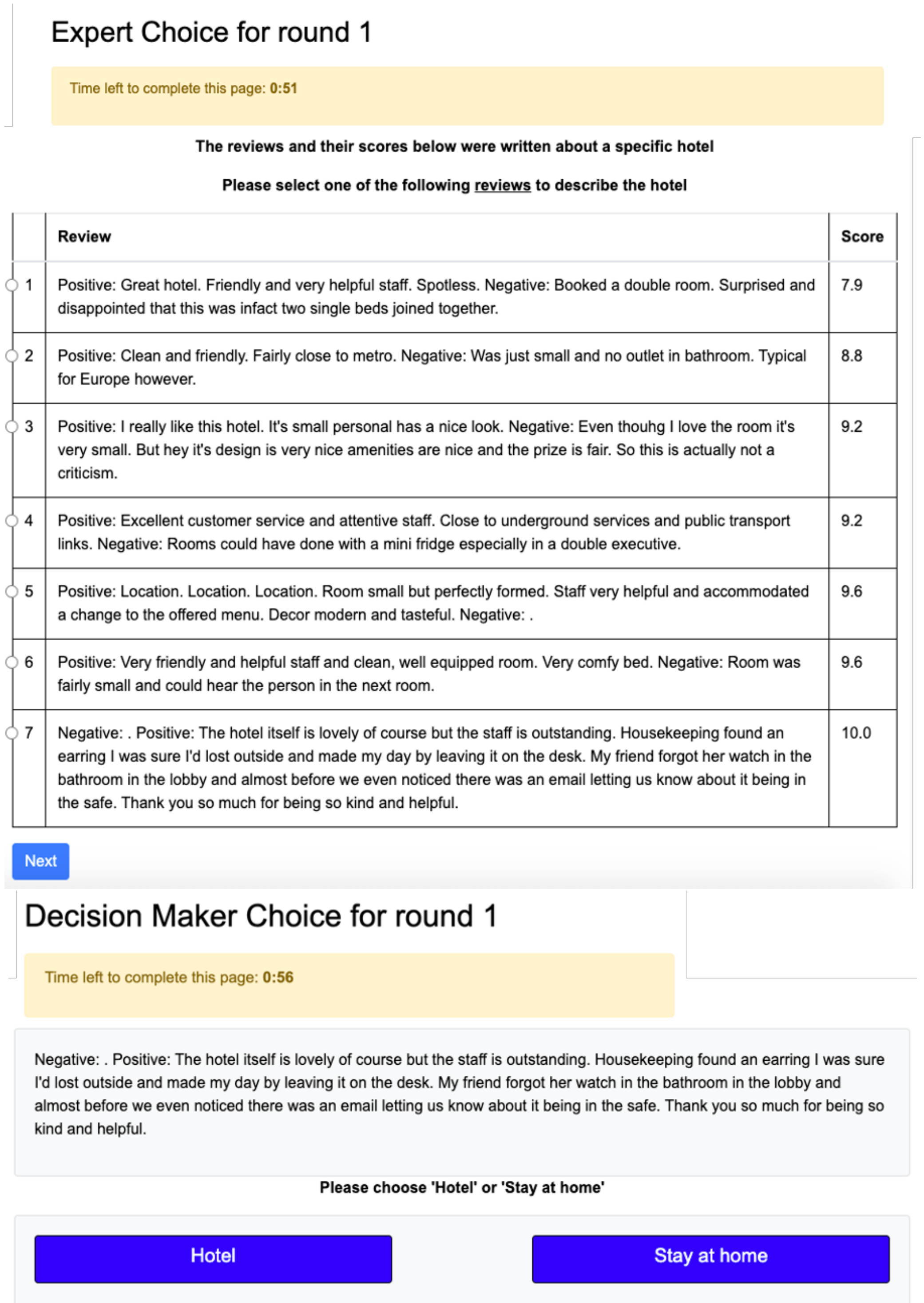
4. Which textual features would serve our prediction model the most? Should we focus on DNN-based features? Or can we also gain from hand-crafted features?

5. Which aspects of the data are crucial for our prediction? Should we only consider the impact of the textual messages or also consider the decision-maker's behavior throughout the game?

The Game In order to implement our setup, we designed a repeated persuasion game between two players, an expert and a decision-maker, using the experimental paradigm presented in Figure 1. The game consists of ten trials, played one after the other. In each trial, the expert tries to sell the decision-maker a different hotel, by sending her textual information about the hotel. Based on this information, the decision-maker is asked to choose between 'hotel' (i.e., the risky action that provides a gain or a loss) and 'stay at home' (i.e., the safe action with a certain payoff of 0). Then, one of the seven scores is randomly selected and determined the decision-maker payoff. At the end of each trial, both participants receive the same feedback that contains the decision-maker's choice, the random score, and their payoffs.

Notations We now formally describe our choice prediction setup. Let $\mathcal{HR}$ be a set of hotels' reviews, and let $\mathcal{HS} \subset \mathbb{R}$ be a set of hotels' scores (written in the well-known Booking.com website). Note that the scores in $\mathcal{HS}$ are between 2.5 and 10, and each review in $\mathcal{HR}$ was originally written with a related score from $\mathcal{HS}$. Let $\mathcal{A} = \{hotel, stay\_home\}$ be the set of action choices made by the decision-makers in each trial of the experiment (which serves to define our labels). The decision-makers make these choices as a response to the textual information that the experts choose to reveal to them.

Prediction Task We are interested in predicting the decisions made by the decision-makers. Specifically, given the information about the first pr trials (hereinafter 'prefix') and partial information about the sf (hereinafter 'suffix') following trials (where $sf = 10 - pr$ and $pr \in \{0, 2, ..., 9\}$), we are interested in predicting the decisions in the sf subsequent trials.



	Review	Score
1	Positive: Great hotel. Friendly and very helpful staff. Spotless. Negative: Booked a double room. Surprised and disappointed that this was infact two single beds joined together.	7.9
2	Positive: Clean and friendly. Fairly close to metro. Negative: Was just small and no outlet in bathroom. Typical for Europe however.	8.8
3	Positive: I really like this hotel. It's small personal has a nice look. Negative: Even thouhg I love the room it's very small. But hey it's design is very nice amenities are nice and the prize is fair. So this is actually not a criticism.	9.2
4	Positive: Excellent customer service and attentive staff. Close to underground services and public transport links. Negative: Rooms could have done with a mini fridge especially in a double executive.	9.2
5	Positive: Location. Location. Location. Room small but perfectly formed. Staff very helpful and accommodated a change to the offered menu. Decor modern and tasteful. Negative: .	9.6
6	Positive: Very friendly and helpful staff and clean, well equipped room. Very comfy bed. Negative: Room was fairly small and could hear the person in the next room.	9.6
7	Negative: . Positive: The hotel itself is lovely of course but the staff is outstanding. Housekeeping found an earring I was sure I'd lost outside and made my day by leaving it on the desk. My friend forgot her watch in the bathroom in the lobby and almost before we even noticed there was an email letting us know about it being in the safe. Thank you so much for being so kind and helpful.	10.0

Figure 1: Screen-shots of the tasks presented to the decision-maker and to the expert.

More concretely, in order to represent a specific interaction with prefix size pr, we define the vector $v_{pr} = (hr_1, hr_2, ..., hr_{10}, a_1, ..., a_{pr}, rs_1, ..., rs_{pr})$ where $hr_t \in \mathcal{HR}$ is the textual information shown to the decision-maker in the t-th trial and $a_t \in \mathcal{A}$ is the decision-maker's choice in trial i. $rs_t \in \mathcal{HS}$ is the score randomly chosen for the hotel out of the set of its review scores in the t-th trial, and determines the decision-maker's payoff in this trial in case of a hotel choice. Hereinafter, we refer to rs as the "random score". Given v_{pr} we are interested in learning the following functions:

1. $F_{trial}(v_{pr}) \in \mathcal{A}^{sf}$: the decision at each trial in the sf subsequent trials. Formally, our trial label in the t-th trial is: $y_{TR_t} = \begin{cases} 1 \text{ if } a_t = hotel \\ 0 \ otherwise \end{cases}$.
2. $F_{ChoiceRate}(v_{pr}) \in \mathbb{R}$: the hotel choice rate in the sf subsequent trials. Formally, our choice rate label is: $y_{CR} = \frac{|\{a_t : a_t = hotel, \ \forall \ i = pr+1, ..., 10\}|}{sf}$.

In this paper we aim to learn the above functions given the history of the first pr trials and the texts shown to the decision-maker in the subsequent trials. In contrast to an online learning setup, in which after each prediction of a decision in the sequence the correct decision is revealed and the learner suffers a loss, here we adopt a batch learning setup. Hence, we do not assume the learner gets neither the correct decision nor the score randomly chosen for the hotel after predicting each suffix trial's decision.

Representation To solve our prediction tasks, we map the vector v_{pr} to the actual inputs of our models, using the behavioral feature space, denoted by $\mathcal{B}$, and the textual feature space, denoted by $\mathcal{T}$ and standing for one of the feature sources we consider in this paper (see Section 5.1.2). More concretely, we consider two different text representation functions: $F_{DNN} : \mathcal{HR} \rightarrow \mathcal{T}_{DNN}$ and $F_{HC} : \mathcal{HR} \rightarrow \mathcal{T}_{HC}$, such that a text $hr \in \mathcal{HR}$ is represented by $F_{DNN}(hr) \in \mathcal{T}_{DNN}$ or by $F_{HC}(hr) \in \mathcal{T}_{HC}$. In our setup, $F_{DNN}(hr)$ corresponds to DNN text representation models, while $F_{HC}(hr)$ corresponds to hand-crafted features. We also consider a representation function $F_{\mathcal{B}} : \mathcal{A} \times \mathcal{HS} \rightarrow \mathcal{B}$, that maps the decision and the random feedback score into our $\mathcal{B}$ feature space, such that each decision and random score are represented by $F_{\mathcal{B}}(a, rs) \in \mathcal{B}$. Details about the texts and the feature spaces are provided in Section 5.1.

4 Data

In this section, we describe our experimental design and data collection process. First, we describe how we collected the participants' actions during our repeated persuasion games. Then, we provide an initial qualitative and quantitative analysis of the collected data.

4.1 Data Collection

Participants were recruited using Amazon Mechanical Turk (https://www.mturk.com/), and the experiment was programmed using oTree (Chen, Schonger, & Wickens, 2016). We paid the participants \$2.5 for their participation and a \$1 bonus based on their performance as described below. The average fee was \$3.07, and the experiment lasted about 15 minutes.

Participants were randomly and anonymously paired, and each of them was randomly selected to be in one of two roles: Decision-maker and expert. The pairs and roles were placed at the beginning of the experiment and remained fixed until the experiment ended.

Each experiment consists of ten trials, played one after the other. In each trial, the expert tries to sell the decision-maker a vacation in a different hotel following the experimental paradigm presented in Figure 1. The expert observes seven reviews written by previous visitors of the hotel, along with their scores: The reviews and their scores were presented in ascending order of score. The expert's task is to select one of the reviews, and this review is revealed to the decision-maker as an estimation of the hotel's quality. Based on the information sent by the expert, the decision-maker is asked to choose between 'hotel' (which provides a gain or a loss) and 'stay at home' (0 payoffs with certainty). The participants had limited time to make their choices. Specifically, the experts had two minutes, and the decision-makers had one minute to make their choice in each trial.

After both participants made their choices, one of the seven scores was randomly sampled from a uniform distribution over the scores. The payoff for the decision-maker, in points, from taking the hotel was this random score minus a constant cost of 8 points. Formally, the decision-maker's payoff in the t-th trial is:

$$DM_{payoff}(a_t) = \begin{cases} tr_t - 8 \text{ if } a_t = hotel \\ 0 \; otherwise \end{cases}$$

This cost reflects a zero expected payoff for a decision-maker who would choose the hotel option in all the ten trials. The payoff for the expert was one point if the decision-maker chose the hotel and 0 otherwise. Formally, the expert's payoff in the t-th trial is:

$$Ex_{payoff}(a_t) = \mathbb{1}_{a_t=hotel}$$

At the end of each trial, both participants received the same feedback that contained the decision maker's choice, the random feedback score, and their payoffs.

As an attention check, participants had to write a specific word to answer the question: "Do you have any comments on this HIT?" before clicking "I agree" at the end of the instructions. Participants who failed this attention check were excluded from the experiment, and their partners were paid for their time. The pairs of participants that passed the attention check were requested to provide personal information, such as age and gender.

At the end of the experiment, the decision-makers were asked to take another attention check. In essence, four reviews were presented to each decision-maker. While two of them were selected by the expert and presented to her during the experiment, the other two were not presented during the entire experiment. The decision-maker was asked to mark the two reviews she had seen before. Decision-makers who had more than one mistake, failed the attention check and were excluded from our analysis.

The probability of obtaining the $1 performance-based bonus was determined by the number of points the participant (expert or decision-maker) accumulated during the experiment. Specifically, we calculated the relative proportion of points that each participant earned from the maximum points she could accumulate during the experiment. If the proportion was higher than a number uniformly sampled from the [0,1] range, the participant received the bonus. We used this bonus mechanism to motivate participants to maximize their payment and, at the same time, maintain a random aspect of their final payment.

We created two separate data sets: Train-validation and test data sets. Both data sets were created using the same process, but in each of them, we used a different set of hotels (see Section 4.2 for hotel description). In the train-validation data, we had 3548 participants, but 34% did not pass the first attention test, and 8% of the remaining participants left the experiment before taking the test. Next, 26% of the remaining participants passed the attention test, but their partner did not pass it or did not take it, and hence they could not continue the experiment. This left us with 1116 participants. We created 558 pairs, but in four cases at least one participant did not meet the deadline in more than 90% of the trials, and in 63 cases at least one participant decided to leave the experiment, and hence we filtered out these pairs. Finally, 16% of the decision-makers that finished the experiment did not pass the second attention test or decided not to take it. These pairs were also filtered out. We thus ended up with 408 pairs (4080 trials) in the train-validation data set.

In the test data, we had 1504 participants, but 40% did not pass the first attention test. A total of 13% of the remaining participants left the experiment before taking the test. Next, 29% of the remaining participants passed the attention test, but their partner did not pass it or did not take it, and hence they could not continue the experiment, which left us with 258 participants. We created 129 pairs, but in seven cases at least one participant decided to leave the experiment, and hence we filtered out these pairs. Finally, 14% of the decision-makers that finished the experiment did not pass the second attention test or decided not to take it. These pairs were also filtered out. We thus ended up with 101 pairs (1010 trials) in the test data set.

4.2 Quantitative Data Analysis

In this section we provide an initial quantitative analysis of the data we have collected. We present the participants' properties, some of the reviews and their scores, and statistics of the decision-makers' decisions. Among the 816 participants that were included in the train-validation data, 408 were female and 408 male. The average, median, and standard deviation of participants' age were 35.5, 32, and 11.34 years, respectively. Among the 202 participants that were included in the test data, 96 were female and 106 male. The average, median, and standard deviation of participants' age were 32.8, 31, and 10.8 years, respectively.

The hotels and their Reviews We use a data set with more than 500,000 reviews written in the well-known Booking.com website. To create the hotels' list for the train-validation set, we randomly select ten hotels with at least seven reviews and each review with at least 100 characters. We then randomly choose seven reviews for each hotel. To create the hotels' list for the test set, we first randomly select ten hotels (excluding the hotels in our train-validation data). We then randomly choose seven reviews for each hotel, such that each hotel is matched to one of the train-validation hotels such that both hotels have close enough score distributions (mean and median differences of up to 0.4, and maximum score difference of up to 0.8). The matching between the hotels in the train-validation and test sets is one-to-one, and this way the difference between the average over the average hotel review scores in both data sets was lower than 0.05.

Below we present four representative reviews from the train-validation data set. All the experts in each data set in our setting observed the same hotels and the same reviews for each hotel. However, the test set hotels and reviews are different than those of the train-validation set (see above). At the beginning of the experiment, we randomly choose for each expert the hotel presentation order during the experiment, and the order of presentation for the positive and negative parts of each review. Table 1 presents the hotels' score distributions in our train-validation and test sets. It shows that the mean score was 8.01 in the train-validation data and 8.06 in the test data, and six of the ten hotels in both sets had a mean score larger than 8. As mentioned in Section 4, the decision maker's payoff in points for taking the hotel at each trial is the random score minus a constant cost of 8 points. These properties of the stimuli imply that the decision maker's expected payoff from always choosing the hotel option was close to zero, and if the decision-makers choose optimally (take all the hotels with a mean score above 8), the expert's average payoff is 0.6.

The reviews differ from each other in many ways. Example properties include the lengths of their negative and positive parts, the topics mentioned in each part, the review's structure, etc. To illustrate this diversity and provide a better exposition of the textual features described in Section 5.2, we provide four representative reviews in Table 2. This table also provides the score associated with each review and the score distribution of the corresponding hotel.

Train-Validation hotels				
#hotel	**Average score**	**Median score**	**Scores' STD**	**Hotel's scores distribution**
1	4.17	3.8	1.39	2.5, 3.3, 3.3, 3.8, 4.2, 5.8, 6.3
2	6.66	7.9	2.45	3.3, 3.3, 6.3, 7.9, 8.3, 8.3, 9.2
3	7.44	7.5	1.63	5, 5.8, 7.1, 7.5, 8.3, 8.8, 9.6
4	7.97	8.3	1.91	5.4, 5.8, 7.1, 8.3, 9.2, 10, 10
5	8.11	8.8	1.62	5.8,6.3, 7.5, 8.8, 8.8, 9.6, 10
6	8.33	8.3	1.79	5.4, 7.1, 7.5, 8.3, 10, 10, 10
7	8.94	9.2	1.34	6.3, 8.3, 8.8, 9.2, 10, 10, 10
8	9.19	9.2	0.68	7.9, 8.8, 9.2, 9.2, 9.6, 9.6, 10
9	9.54	9.6	0.28	9.2, 9.2, 9.6, 9.6, 9.6, 9.6, 10
10	9.77	9.6	0.21	9.6, 9.6, 9.6, 9.6, 10, 10, 10
Average Score		**8.01**		

Test hotels				
#hotel	**Average score**	**Median score**	**Scores' STD**	**Hotel's scores distribution**
1	4.56	4.2	0.97	3.8, 3.8, 3.8, 4.2, 4.6, 5.4, 6.3
2	6.69	6.7	2.09	4.2, 4.2, 6.3, 6.7, 7.1, 8.3, 10
3	7.74	8.3	1.78	5, 5.8, 7.5, 8.3, 8.8, 8.8, 10
4	7.91	7.9	1.49	5.8, 6.7, 7.5, 7.9, 7.9, 9.6, 10
5	8.04	8.3	1.24	5.8, 7.5, 7.5, 8.3, 8.8, 8.8, 9.6
6	8.33	8.3	1.82	5, 7.1, 8.3, 8.3, 9.6, 10, 10
7	8.89	8.8	1.34	6.3, 8.3, 8.8, 8.8, 10, 10, 10
8	9.04	9.2	0.99	7.9, 7.9, 8.3, 9.2, 10, 10, 10
9	9.65	9.6	0.42	8.8, 9.6, 9.6, 9.6, 10, 10, 10
10	9.66	9.6	0.43	8.8, 9.6, 9.6, 9.6, 10, 10, 10
Average Score		**8.06**		

Table 1: Hotels' score distributions in our train-validation and test sets.

Behavior Statistics We now turn to discuss the statistics of the participants' behavior in the train-validation data set. This is important not only for the analysis of how participants behave in the game, but also in order to understand the data and the features we extract from the texts.

	Review	Score	hotel's scores distribution
1	**Negative:** The swimming pool on the top of the roof is very small. In high season there is little possibility that you will be able to use it. The worst thing was that during my stay the crew started to paint all the walls on my floor's corridor. The paint smell really awful. Although the stuff from the Reception desk was ok the women bartender who worked on morning shift wasn't very nice maybe she felt a little bit sleepy. In my opinion the cost was too high compared to the offer. **Positive:** The location is awesome. You can go across the street and grab a subway. The Sagrada Familia is about 15 20 minutes by foot.	5.8	5.8, 6.3, 7.5, 8.8, 8.8, 9.6, 10
2	**Positive**: The whole experience of our trip to Barcelona and the hotel was perfect. I can not speak highly enough of everyone who made our stay so special. Our room was lovely and clean with a fantastic shower and huge comfy bed. We spent time in the spa and on the roof terrace which has spectacular views over the city very close to the metro so getting about was easy I will return here I hope sometime in the future. **Negative:** I really cannot think of anything at the moment.	10	5.8, 6.3, 7.5, 8.8, 8.8, 9.6, 10
3	**Negative:** 1. we didn't received what we asked a room with a bath and a double bed 2. no WIFI only in the lobby 3. room was to hot airco didn't worked properly 4. really old fashion and this hotel urgently needs to be refreshed 5. simple breakfast. RESUMED this hotel does not deserve 4 stars at all and can not be recommended at all. We don't understand that booking.com included it in its list. **Positive:** the location.	3.3	2.5, 3.3, 3.3, 3.8, 4.2, 5.8, 6.3
4	**Negative:**. **Positive:** Location. Location. Location. Room small but perfectly formed. Staff very helpful and accommodated a change to the offered menu. Decor modern and tasteful.	9.6	7.9, 8.8, 9.2, 9.2, 9.6, 9.6, 10

Table 2: Example of four reviews, their scores, and the score distributions of all the reviews assigned to the same hotel. All the reviews are part of the train-validation data.

Figure 2 presents the percentage of decision-makers that chose the hotel option. The left histogram presents this percentage as a function of the trial number. It shows that this value decreases as the number of attempts increases, but the slope is moderate. The decrease with time can imply either a better understanding of the instructions and a better inference of the hotels' quality from the content of the reviews, or a lower trust in the expert as the experiment progresses. The right histogram shows this percentage as a function of the scores associated with the reviews that were presented to the decision-makers during the experiment. This histogram demonstrates that the decision-makers tended to choose the hotel option as the review score increases. This result indicates that although the decision-makers only observed the reviews and not the scores, they could infer the quality of the hotels from the content of the reviews. Both histograms also demonstrate that there is not much of a difference between male and female participants, and hence we did not use the gender as a feature in our models.

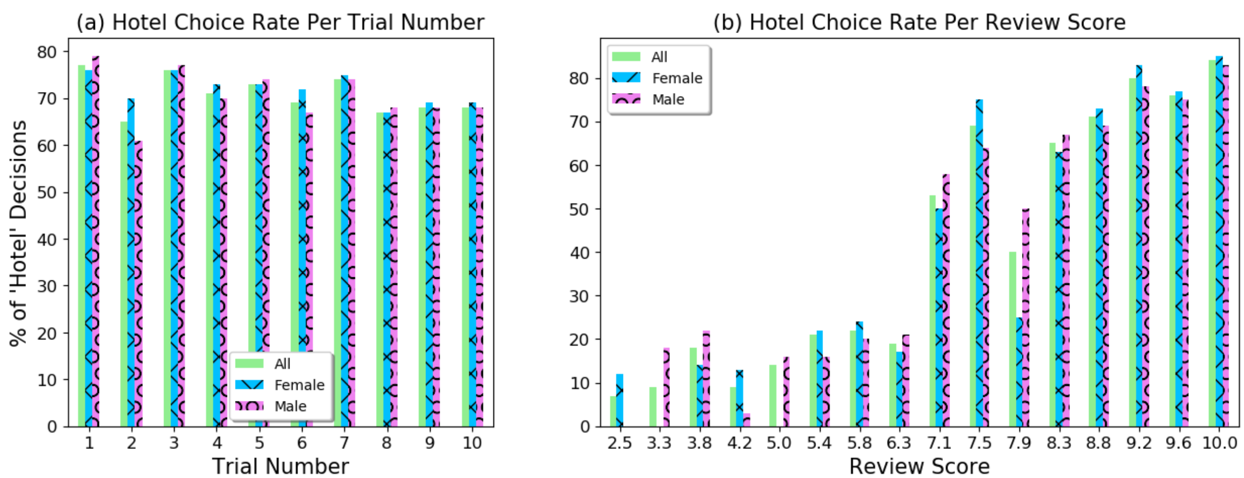


Figure 2: Hotel choice rate histogram. Figure (a) shows the percentage of decision-makers who chose the hotel option as a function of the trial number. Figure (b) shows the percentage of decision-makers who chose the hotel option as a function of the scores associated with the reviews that were presented to them.

As noted above, we designed our experimental paradigm such that every decision-maker that would choose the hotel option in all the ten trials would have a zero expected payoff. An optimal decision-maker would only choose the hotel option, when the mean score is above 8, i.e., in 60% of the trials. For each number of trials, K, Figure 3 presents the percentage of decision-makers who chose the hotel K times. The histogram demonstrates that 95.3% of the decision-makers chose the hotel option in at least half of the trials. Particularly, the average, median, and standard deviation of the total number of hotel choices were 7.18, 7, and 1.52, respectively. These results show that the decision-makers tend to choose the hotel option as expected since, in seven of the ten hotels, the median and the average score were above or very close to 8. This behavior is in line with the experimental phenomenon known as 'probability matching' (Vulkan, 2000). The results further indicate that baseline strategies that assign the average or the median hotel choice rate for every participant are very effective when it comes to predicting the overall choice rate. However, we also aim to perform this prediction correctly for populations that differ from the average participant in this aspect.

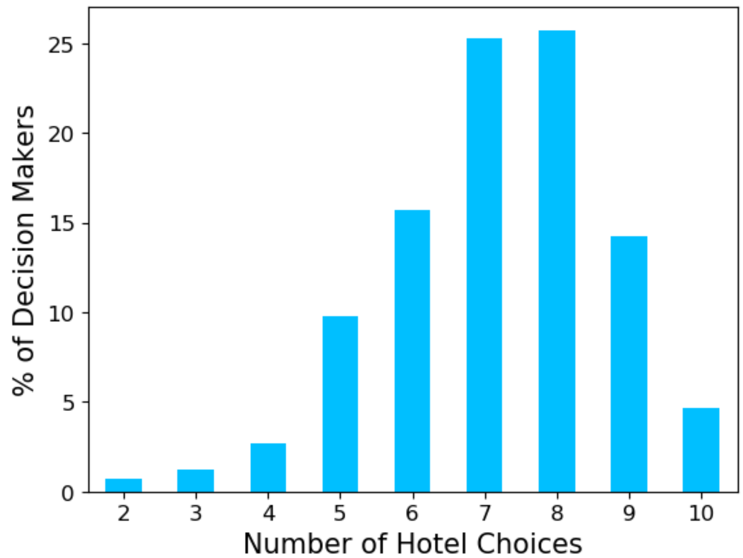


Figure 3: The tendency to choose the hotel option. For each number of trials, K, (x axis) the histogram presents the percentage of decision makers who chose the hotel K times.

This analysis can open a direction to use Bayesian modeling that will take into consideration prior knowledge regarding the decision-makers' tendency to choose the hotel option. While we do not design such a model in this paper, we employ this knowledge in our baselines and aim to design models that outperform a baseline that predicts the average or the median hotel choice rate.

Figure 4 presents the decision-makers' decisions as a function of their previous choices, as well as the score that was randomly selected at the end of each trial and determines the decision maker's payoff. It shows these choice rates in the cases where the previous decision was hotel and in the cases where it was 'stay at home'. The figure indicates that the decision in the previous trial and the feedback the decision-makers got, directly influence her decision in the subsequent trial. Focusing on previous trial random scores that are higher than 8 (i.e., higher than the cost of choosing the hotel option), we can infer that if the decision-makers chose the hotel option and earned, they are more inclined to choose the hotel option in the next trial, compared to the case where they chose the 'stay at home' option and could earn. In addition, focusing on cases where the random score selected in the previous trial was lower than 8, we can infer that the decision-makers would like to compensate for their losses in cases where they chose the hotel and lost. Generally, a previous hotel decision indicates a higher likelihood of a subsequent hotel decision again, and a higher previous random score indicates a lower probability of a hotel decision in the next trial.

In summary, our analysis shows that the decision-makers could infer the hotels' quality from the reviews' content. In addition, it indicates that the decision-makers' decision and the feedback they observe after each trial influence their decision in the subsequent trial. Finally, there is no significant difference between female and male decision-makers. These results led us to use the text the decision-makers observe at each trial, their decision and the feedback they observe in the prefix trials as features in our models. The sequential effect also calls for the application of sequential models.

In Section 5, we expand our analysis to include the decision-makers' behavior as a response to

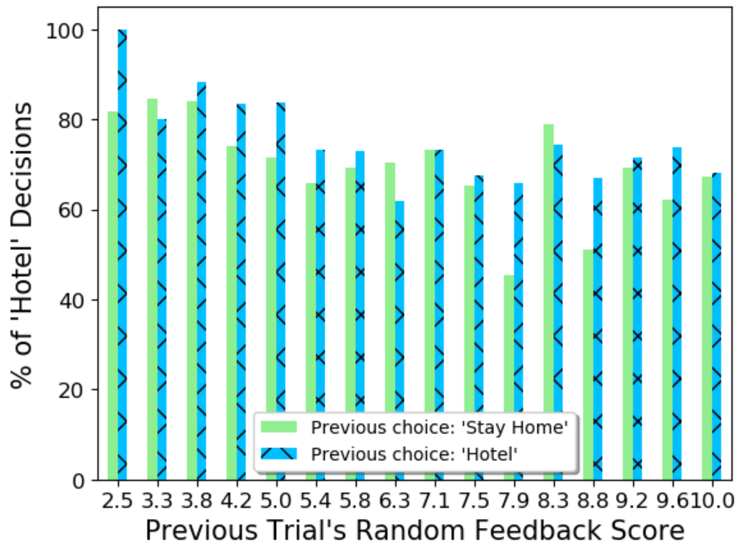


Figure 4: Decision makers' decisions as a function of previous trial decision and the random feedback score. The figure presents the percentage of decision-makers who chose the hotel option both in cases where the previous decision is 'hotel' and in cases where it is 'stay at home'.

the features of the text they observe. This analysis will be presented following a description of our human generation textual feature space.

5 Our Approach

We will next describe our approach. In order to answer our research questions described in Section 3, our approach is based on two steps: In the first step, we map each vector $v_{pr} = (hr_1, hr_2, ..., hr_{10}, a_1, ..., a_{pr}, rs_1, ..., rs_{pr})$ to a feature vector using our textual feature sets ($\mathcal{T}$) and our behavioral feature set ($\mathcal{B}$). In the second step, we design non-structured classifiers, sequence models and attention-based models that learn the two functions we discussed in Section 3: $F_{ChoiceRate}$ and F_{trial}. In order to predict the hotel choice rate, we consider three modeling strategies: (a) direct prediction of the choice rate; (b) prediction of the decision at each trial, i.e., designing models that learn the F_{trial} function, from which the choice rate can be derived; and (c) learning of these two functions jointly. The decision sequence is considered in both the representation and the modeling steps of our approach.

Data Representation In the first step, we first map each review $hr \in \mathcal{HR}$ to the features that represent it in our two text feature spaces (i.e., $\mathcal{T}_{HC}$ and $\mathcal{T}_{DNN}$) using our two functions (F_{HC} and F_{DNN}). In addition, we map each pair $(a, rs) \in \mathcal{A} \times \mathcal{HS}$ of a decision and a random score to our behavioral feature space, $\mathcal{B}$. We then use these functions to map each vector $v_{pr} = (hr_1, hr_2, ..., hr_{10}, a_1, ..., a_{pr}, rs_1, ..., rs_{pr})$ into the feature vectors that will serve as our models' inputs. The features are explained in detail in Section 5.1, and the mapping of v_{pr} into our feature space is described in Section 5.2.

Models In the second step, we want to learn the two functions we defined in Section 3: $F_{ChoiceRate}$ and F_{trial}. We explore different modeling strategies, particularly sequential models that learn to predict the decision in each trial of the suffix, sequential and non-sequential models that learn to directly predict the hotel choice rate in the suffix, and joint models that jointly learn to predict both the choice rate and the decision in each trial. We now elaborate on these two steps.

5.1 Feature Sets

In this section, we describe our three feature sets: the $\mathcal{B}$ feature set that represents the decision-maker's behavior in the prefix, as well as $\mathcal{T}_{HC}$ and $\mathcal{T}_{DNN}$ that represent the texts. The final representation of each vector v_{pr} is the concatenation of the textual features and the behavioral features and is different for each model. The specific input for each model is described in Section 5.2.

5.1.1 Behavioral Features ($\mathcal{B}$)

Our task deals with a communication setup. We are trying to understand if we can predict future decision-maker behavior from the information that (a) was available to her and (b) can be observed by an external spectator. Specifically, what was presented to the decision-maker, what decisions did she make previously and what feedback did she observe after her decisions. Therefore, we do not encode information that was not available to the decision-maker, such as the score distribution from which her payoff is drawn or the score of the review that was selected by the expert.

Here, we describe our behavioral feature space, $\mathcal{B}$. Specifically, we map each pair $(a, rs) \in \mathcal{A} \times \mathcal{HS}$ of a decision and a random score that determines the decision-maker's payoff to the following eight features:

1. Decision: a binary feature with the decision-maker's choice at the current trial, i.e., $a \in \mathcal{A}$.

2. Random score: three binary features that indicate whether the random score (rs) is lower than 3, between 3 to 5, or higher than 8. Note that this random score determines the decision-maker's payoff, in case of a hotel choice.

3. Chose and lose (cl): a binary feature that indicates whether the decision-maker chose the hotel option and lost. Formally, $cl = \begin{cases} 1 \text{ if } a = hotel\ and\ rs < 8 \\ 0\ otherwise \end{cases}$.

4. Did not choose and could lose (nccl): a binary feature that indicates whether the decision-maker did not choose the hotel option and could have lost, had she chosen it. Formally, $nccl = \begin{cases} 1 \text{ if } a = stay\_home\ and\ rs < 8 \\ 0\ otherwise \end{cases}$.

5. Chose and earned (ce): a binary feature that indicates whether the decision-maker chose the hotel option and earned points. Formally, $ce = \begin{cases} 1 \text{ if } a = hotel\ and\ rs \geq 8 \\ 0\ otherwise \end{cases}$.

6. Did not choose and could earn (ncce): a binary feature that indicates whether the decision-maker did not choose the hotel option and could have earned had she chosen it. Formally, $ncce = \begin{cases} 1 \text{ if } a = stay\_home \; and \; rs \geq 8 \\ 0 \; otherwise \end{cases}$.

Since these features provide information regarding the decision and the feedback, and since we perform batch rather than online learning, we use them to describe only the prefix trials. In Section 5.2, we describe the way we encode these features into each of our models. Below we refer to the value of the j'th behavioral feature as $\mathcal{B}_j$.

5.1.2 Textual Features

We have so far dealt with the representation of the behavioral information, and we will now move to describe the features that represent the texts observed by the decision makers. Previous works have already modeled sequential decision making (e.g., Kolumbus & Noti, 2019) but have not modeled text as the basis of these decisions, and hence this is a contribution of this paper.

We focus on two sets of textual features: $\mathcal{T}_{DNN}$: Features inferred by pre-trained deep contextualized embedding models, and $\mathcal{T}_{HC}$: Hand-crafted features. Research into textual representations has recently made significant progress with the introduction of pre-trained deep contextualized embedding models (Peters, Neumann, Iyyer, Gardner, Clark, Lee, & Zettlemoyer, 2018; Radford, Narasimhan, Salimans, & Sutskever, 2018). In particular, we chose the BERT pre-trained model (Devlin, Chang, Lee, & Toutanova, 2018) as our text encoder since it is dominant in NLP research. We would also like to explore the value of hand-crafted, high-level semantic features for our task, because models like BERT, that are based on language modeling related objectives, may not capture the high-level semantic information encoded in these features. We first describe our two approaches for textual representation, and in Section 5.2 we discuss how these features are integrated into our models. Below we refer to the value of j'th textual feature as $\mathcal{T}_j$.

BERT-based representation ($\mathcal{T}_{DNN}$) In this approach, we utilize the pre-trained BERT model as a source of text representation. BERT is a contextualized language representation model that is based on a multi-layer bidirectional Transformer architecture and a masked language model objective. We used the uncased pre-trained BERT-Base model (L = 12 layers, H = 768 hidden vector size, A = 12 attention heads, P = 110M parameters), trained on the BookCorpus (800M words) (Zhu et al. 2015) and Wikipedia (2,500M words), publicly available via source code provided by the Google Research's GitHub repository.[2] We utilized the model's source code from the "HuggingFace's PyTorch Pretrained BERT" GitHub repository.[3] BERT can handle a sequence of up to 512 tokens, and since we use relatively short texts (our longest review contains only 144 tokens), BERT fits our needs. For each review we produce an embedding vector, by extracting the vector associated with the special [CLS] token from the last hidden layer of BERT.

Hand Crafted Features ($\mathcal{T}_{HC}$) We define 42 binary hand-crafted textual features, aiming to capture high-level semantic information that may not be captured by DNNs like BERT. The features

[2] `https://github.com/google-research/bert`
[3] `https://github.com/huggingface/transformers`

are described in Table 4, while Table 5 presents the feature representation of the hotel reviews from Section 4.2.

Some of the features make use of sentiment words. In order to adjust such sentiment words to our goal, we decided to extract these words from the reviews, instead of using a publicly available sentiment words list. In this service, three graduate students read the train-validation hotels' reviews, extracted positive and negative sentiment words, and divided them into three groups, according to their positive and negative intensity. These lists were then merged into a unified list. The groups of positive and negative sentiment words are listed in Table 3.

	Group #1	**Group #2**	**Group #3**
Positive Part	comfort, ok, close, like, sleek, spacious, available, fair, welcoming, helpful, new, lots, extra, neat, tastefully, cleanliness, friendly, good, variety, quiet, approachable, refurbished, surprising, free, honest, attentive, comfy, comfortable, newly, clean, nicely well, easy, big plenty, safe, nice	very, large, surprise, great, really, love, renovated, huge, unrivalled, pristine, enjoyed, impressive, extensive, warm, brilliant, exceeded professional, useful, efficiently, highly everything	stunning, superb, extremely, impressed, fabulous, perfectly, perfect, awesome, outstanding, wow, incredible, top class, beyond fabulous, best, phenomenal, special, spectacular, fab, magnificent, fantastic, largest, spotless, excellent, amazing
Negative Part	little, small, overpriced, could be better, dirty, supposed, tacky, not friendly, basic, outdated, bit of, noisy, expensive, simple, limited, smell, old, tight, missing, compact, inconspicuous, busy, isolated, average, pricier, noise, thin, not obsolete, sleepy, empty, uncomfortable, regular	very, everything, not good enough, too, bad, really, poor, sufficiently claustrophobic	incredibly, shockingly, overcharged, awful, disgusting, worst, completely, flooding, extremely, surprised, dangerous, exceptional, urgently, terrible, disaster, disappointed

Table 3: Sentiment word groups used in the hand-crafted features.

High Level	**#**	**Feature Name**	**Feature Description**
	1	Facilities	The positive part provides details about the hotel's facilities
	2	Price	The positive part provides details about the hotel's price

Positive Part Topics	3	Design	The positive part provides details about the hotel's design
	4	Location	The positive part provides details about the hotel's location
	5	Room	The positive part provides details about the room
	6	Staff	The positive part provides details about the hotel's staff
	7	Food	The positive part provides details about the food in the hotel
	8	Transportation	The positive part provides details about the transportation options in the hotel's area
	9	Sanitary Facilities	The positive part provides details about the sanitary facilities in the room
	10	View	The positive part provides details about the view from the hotel
Positive Part Properties	11	Empty	The positive part is empty
	12	Nothing positive	The positive part explicitly states that there is nothing positive about the hotel
	13	Summary Sentence	The positive part provides a positive summary sentence, e.g., "We would stay there again"
	14	Words from the first positive group	The positive part contains words from the first positive group
	15	Words from the second positive group	The positive part contains words from the second positive group
	16	Words from the third positive group	The positive part contains words from the third positive group
	17	Short positive part	The number of characters in the positive part is lower than 100
	18	Medium positive part	The number of characters in the positive part is between 100 and 199
	19	Long positive part	The number of characters in the positive part is higher than 200
Negative Part Topics	20	Price	The negative part provides details about the hotel's price
	21	Staff	The negative part provides details about the hotel's staff
	22	Sanitary Facilities	The negative part provides details about the sanitary facilities in the room

	23	Room	The negative part provides details about the room
	24	Food	The negative part provides details about the food in the hotel
	25	Location	The negative part provides details about the hotel's location
	26	Facilities	The negative part provides details about the hotel's facilities
	27	Air	The negative part provides details about the hotel's air-conditioning facilities
Negative Part Properties	28	Empty	The negative part is empty
	29	Nothing negative	The negative part explicitly states that there is nothing negative about the hotel
	30	Summary Sentence	The negative part provides a negative summary sentence, e.g., "I do not know how it is a 4 stars hotel"
	31	Words from the first negative group	The negative part provides words from the first negative group
	32	Words from the second negative group	The negative part provides words from the second negative group
	33	Words from the third negative group	The negative part provides words from the third negative group
	34	Short negative part	The number of characters in the negative part is lower than 100
	35	Medium negative part	The number of characters in the negative part is between 100 and 199
	36	Long negative part	The number of characters in the negative part is higher than 200
Overall Review Properties	37	Detailed Review	The review provides many details about the hotel
	38	Review structured as a list	The review is arranged as a list of the hotel's positive and negative properties
	39	Positive part shown first	The positive part is shown before the negative part
	40	Low proportion between positive and negative parts' lengths	The proportion between the number of characters in the positive and the negative parts is lower than 0.7
	41	Medium proportion between positive and negative parts' lengths	The proportion between the number of characters in the positive and the negative parts is between 0.7 and 4

	42	High proportion between positive and negative parts' lengths	The proportion between the number of characters in the positive and the negative parts is higher than 4

Table 4: Descriptions of the hand-crafted features.

High Level	#	**Feature Name**	**Text #1**	**Text #2**	**Text #3**	**Text #4**
Positive Part Topics	1	Facilities	0	1	0	0
	2	Price	0	0	0	0
	3	Design	0	0	0	1
	4	Location	1	0	1	1
	5	Room	0	1	0	1
	6	Staff	0	0	0	1
	7	Food	0	0	0	0
	8	Transportation	0	1	0	0
	9	Sanitary Facilities	0	1	0	0
	10	View	0	1	0	0
Positive Part Properties	11	Empty	0	0	0	0
	12	Nothing positive	0	0	0	0
	13	Summary Sentence	0	1	0	0
	14	Words from the first positive group	0	1	0	1
	15	Words from the second positive group	0	1	0	1
	16	Words from the third positive group	1	1	0	1
	17	Short positive part	0	1	0	0
	18	Medium positive part	0	0	1	0
	19	Long positive part	1	0	0	1
Negative Part Topics	20	Price	0	0	0	0
	21	Staff	1	0	0	0
	22	Sanitary Facilities	0	0	0	0
	23	Room	0	0	1	0
	24	Food	0	0	1	0
	25	Location	0	0	0	0
	26	Facilities	1	0	0	0
	27	Air	0	0	1	0
	28	Empty	0	0	0	1
	29	Nothing negative	0	1	0	0
	30	Summary Sentence	0	0	1	0
	31	Words from the first negative group	1	1	1	0

Negative Part Properties	32	Words from the second negative group	1	1	1	0
	33	Words from the third negative group	1	0	1	0
	34	Short negative part	0	0	0	0
	35	Medium negative part	0	1	0	1
	36	Long negative part	1	0	1	0
Overall Review Properties	37	Detailed Review	1	1	0	0
	38	Review structured as a list	0	0	1	0
	39	Positive part shown first	0	1	0	0
	40	Low proportion between positive and negative parts' lengths	0	1	0	1
	41	Medium proportion between positive and negative parts' lengths	1	0	1	0
	42	High proportion between positive and negative parts' lengths	0	0	0	0

Table 5: Hand-crafted feature values for the texts of Section 4.2.

Figure 5 analyses the quality of our hand-crafted features. It shows the fraction of decision-makers in the train-validation data set that select the hotel option, in cases where the reviews they saw at each trial include or do not include each feature. For example, the numbers for feature #11 indicate that adding a positive part to a review increases the choice rate of the hotel option by 72%. As another example, adding a positive bottom line (as in text #2, see Table 5), increases the hotel choice rate by 25% percent, as indicated by the numbers for feature #13. Comparing the choice rates of features #14, #15 and #16 reveals that using words that more strongly emphasize positive aspects of the hotel increases the acceptance rate. A similar comparison of features #31 − #33 reveals their negative impact on the hotel choice probability.

5.2 Models

Now that we have discussed the properties of our task and data, we are ready to present the way we predict the outcome of the human behavior in our task, i.e. learning the functions $F_{ChoiceRate}(v_{pr})$ and $F_{trial}(v_{pr})$ from Section 3. We propose models that learn each of these functions separately (denoted with -CR, for Choice Rate, and -TR, for TRial, respectively), as well as models that learn them jointly (denoted with -TRCR). We are particularly focused on two modeling aspects: Comparing sequential models to non-sequential ones, and comparing DNN-based models to models that do not apply this approach. We next provide a high level description of our models, and then proceed with more specific details.

$F_{ChoiceRate}()$ **Models** . We implement three models for $F_{ChoiceRate}(v_{pr})$. Two of our models are DNN-based: One employs a Long Short-Term Memory (LSTM) recurrent architecture (Hochreiter

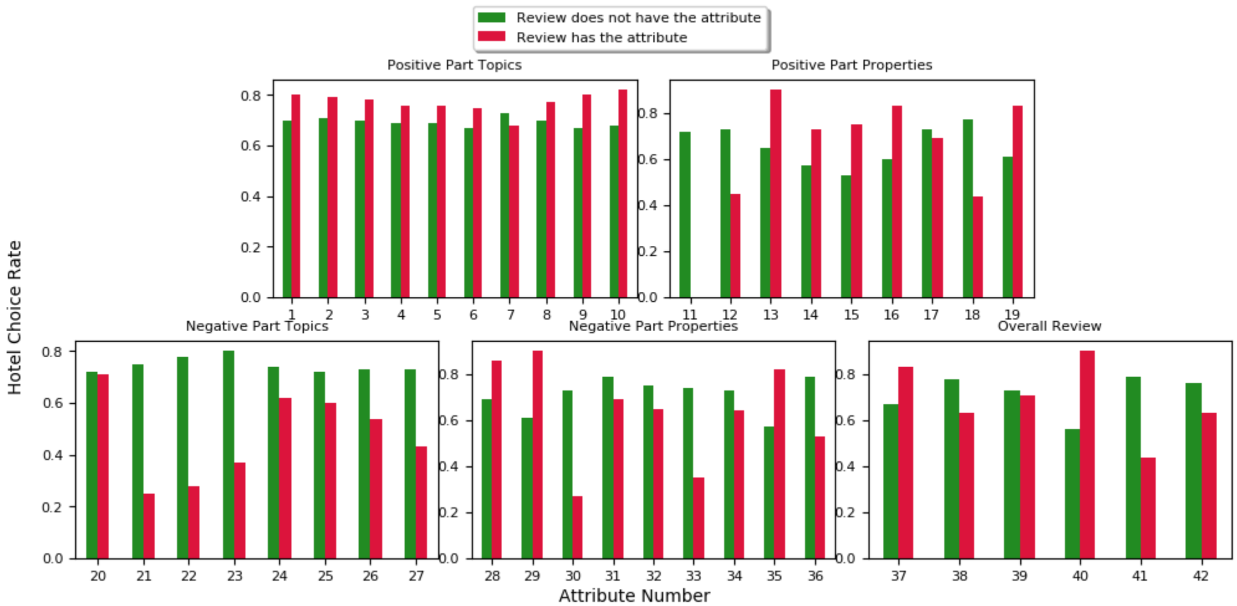


Figure 5: Hotel choice rates as a function of the hand-crafted feature values. The analysis is based on the train-validation dataset, and each histogram presents the analysis of a specific feature subset.

& Schmidhuber, 1997) for sequence processing, and the other employs the Transformer architecture (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, & Polosukhin, 2017) for sequence-to-sequence learning with self-attention. The third model is a Support Vector Machine (SVM) (Cortes & Vapnik, 1995), which will let us evaluate the power of a non-DNN and non-sequential modeling approach.

$F_{trial}()$ Models . We implement two models for $F_{trial}(v_{pr})$. Note that these models address a more general task than the $F_{ChoiceRate}()$ models, since the hotel choice rate can be derived from their per-trial predictions. As for $F_{ChoiceRate}()$, one of our $F_{trial}()$ models is based on an LSTM architecture and one on the transformer architecture.

Joint modeling of $F_{ChoiceRate}()$ and $F_{trial}()$. Multi-task Learning is an approach in which multiple learning tasks are solved jointly, by sharing related information (Reichart, Tomanek, Hahn, & Rappoport, 2008; Ruder, 2017). As the choice rate and per-trial prediction tasks are tightly connected, we hope that solving them jointly would produce better results on each. Multi-task learning has been applied to a variety of NLP tasks, and DNNs are particularly suitable for its implementation (e.g. Søgaard & Goldberg, 2016; Rotman, Vulić, & Reichart, 2018; Malca & Reichart, 2018). We therefore implemented a model that jointly learns the $F_{ChoiceRate}()$ and $F_{trial}()$ functions. As for the $F_{ChoiceRate}()$ and $F_{trial}()$, one of our joint models is based on an LSTM architecture and one on the transformer architecture. We next describe each of our models in detail.

5.2.1 Support Vector Machines for $F_{ChoiceRate}()$ (SVM-CR)

Here we describe our SVM regression model that predicts the hotel choice rate in a given trial suffix. Since SVMs are not sequential models, this model considers the past sequence of texts and behaviors only through its features. We considered various possible representations and, based on development data experiments, we decided to represent the input state v_{pr} by the weighted average of the prefix trials' textual features ($PW\mathcal{T}$), the weighted average of the prefix trials' behavioral features ($PW\mathcal{B}$), and the average text features of all the trials in the suffix ($SW\mathcal{T}$). The weighted average of the prefix trials is defined such that earlier trials get lower weights.

Formally, the weighted average of the j-th prefix behavioral feature, $\mathcal{B}_j$, and the weighted average of the j-th prefix textual feature, $\mathcal{T}_j$, for a prefix size pr, are:

$$PW\mathcal{B}_j = \frac{1}{pr}\sum_{t=1}^{pr} 0.8^{pr+1-t} \cdot \mathcal{B}_{jt}$$

$$PW\mathcal{T}_j = \frac{1}{pr}\sum_{t=1}^{pr} 0.9^{pr+1-t} \cdot \mathcal{T}_{jt}$$

where $\mathcal{B}_{jt}$ is the value of the j-th behavioral feature in the t-th trial , $\mathcal{T}_{jt}$ is the value of the j-th textual feature in the t-th trial , and 0.8 and 0.9 are hyper-parameters tuned on our development data.

For example, the vector v_4 (i.e., a vector of an example with a prefix of size 4) is mapped into the concatenation of the following features:

- The prefix trials' weighted j-th behavioral feature: $PW\mathcal{B}_j = \frac{1}{4}\sum_{t=1}^{4} 0.8^{5-t} \cdot \mathcal{B}_{jt}, \forall \mathcal{B}_{jt} \in \mathcal{B}$
- The prefix trials' weighted j-th textual feature: $PW\mathcal{T}_j = \frac{1}{4}\sum_{t=1}^{4} 0.9^{5-t} \cdot \mathcal{T}_{jt}, \forall \mathcal{T}_{jt} \in \mathcal{T}$
- The suffix trials' weighted j-th textual feature: $SW\mathcal{T}_j = \frac{1}{6}\sum_{t=5}^{10} \mathcal{T}_{jt}, \forall \mathcal{T}_{jt} \in \mathcal{T}$

5.2.2 Deep Neural Network Modeling

DNNs have proven effective for many text classification tasks (Kim, 2014; Ziser & Reichart, 2018). In this part of the paper, we provide a high level description as well as more specific details of our DNN models.

In our DNN models, each trial in the prefix is represented using its behavioral features (as described in Section 5.1). These features are concatenated to the trial's textual features (either its $\mathcal{T}_{DNN}$ or its $\mathcal{T}_{HC}$ features, or their concatenation, as described in Section 5.1). In contrast, since the suffix trials' behavioral features are not known, each trial in the suffix is represented only with its textual features.

The LSTM Models These models belong to the family of Recurrent Neural Networks (RNNs), which can process variable length sequences. We hypothesize that since our data involve multiple trials, and based on our analysis described in Section 4.2 where we show a sequential effect in the decision-making process, a sequential model could capture signals that non-sequential models cannot.

LSTM is an RNN variant designed to handle long-distance dependencies, while avoiding the vanishing gradients problem. It has shown very useful in sequence modeling tasks in NLP, such as language modeling (Sundermeyer, Schlüter, & Ney, 2012), speech recognition (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2016) and machine translation (Wu, Schuster, Chen, Le, Norouzi, Macherey, Krikun, Cao, Gao, Macherey, et al., 2016). We describe our LSTM models below, focusing on their labels, input vectors and architectures.

We have considered various LSTM-based models and multiple approaches for mapping the input v_{pr} as these models' input. Each input v_{pr} is represented with a sequence of feature vectors, such that each feature vector represents one trial (either a prefix or a suffix trial; the feature vectors of each prefix and suffix trial are described above). We next describe the best model version based on our development data results, an illustration of the architecture is provided in Figure 6.

LSTM-CR. This is the LSTM model that predicts the hotel choice rate in the suffix. Figure 6 (right) provides a description of this architecture. The LSTM is sequentially fed with the prefix and suffix trials' representations, one trial at a time. The suffix trials' hidden vectors are fed into a dot product attention layer, followed by two linear layers with a dropout layer and a ReLU activation function, in order to predict the hotel choice rate in the suffix trials. The model applies the mean squared error (MSE) loss as implemented in the PyTorch.nn module:[4]

$$MSE = \frac{1}{batch} \sum_{i=1}^{batch} (\hat{y}_{CR_i} - y_{CR_i})^2$$

where *batch* is the size of the training batch (in the stochastic optimization process), and $\hat{y}_{CR_i}$ and y_{CR_i} are the predicted and the gold hotel choice rates in the i-th example of the batch, respectively.

LSTM-TR. This is the LSTM model that predicts the decision in each suffix trial. The LSTM-TR architecture is described in the left side of Figure 6. The result of this model can also be averaged in order to get the hotel choice rate in the suffix trials.

The LSTM is sequentially fed with the prefix and suffix trials' representations, one trial at a time. Each hidden state of the suffix trials is fed into a dropout layer followed by a linear layer with a ReLU activation function, in order to predict the label for each suffix trial. The loss function of this model is the sequence cross-entropy (SCE), as implemented in the AllenNLP software package:[5]

$$SCE = \frac{1}{batch} \sum_{i=1}^{batch} \frac{\sum\limits_{j=pr+1}^{10} -(y_{TR_{ti}} \cdot log(p_{ti}) + (1 - y_{TR_{ti}}) \cdot log(1 - p_{ijs}))}{sf}$$

where *batch* is the size of the training batch (in the stochastic optimization process), *pr* is the prefix size, sf is the suffix size, p_{ti} is the predicted probability that the t-th trial of the i-th example of the batch is hotel, a_{ti} is the decision in the t-th trial of the i-th example of the batch and $y_{TR_{ti}} \in \{0, 1\}$ is the choice of the t-th example of the batch in the i-th trial, such that $y_{TR_{ti}} = \begin{cases} 1 \text{ if } a_{ti} = hotel \\ 0 \; otherwise \end{cases}$.

LSTM-TRCR. This model jointly learns to predict the decisions made by the decision maker in

[4] https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html
[5] https://github.com/allenai/allennlp/blob/master/allennlp/nn/util.py

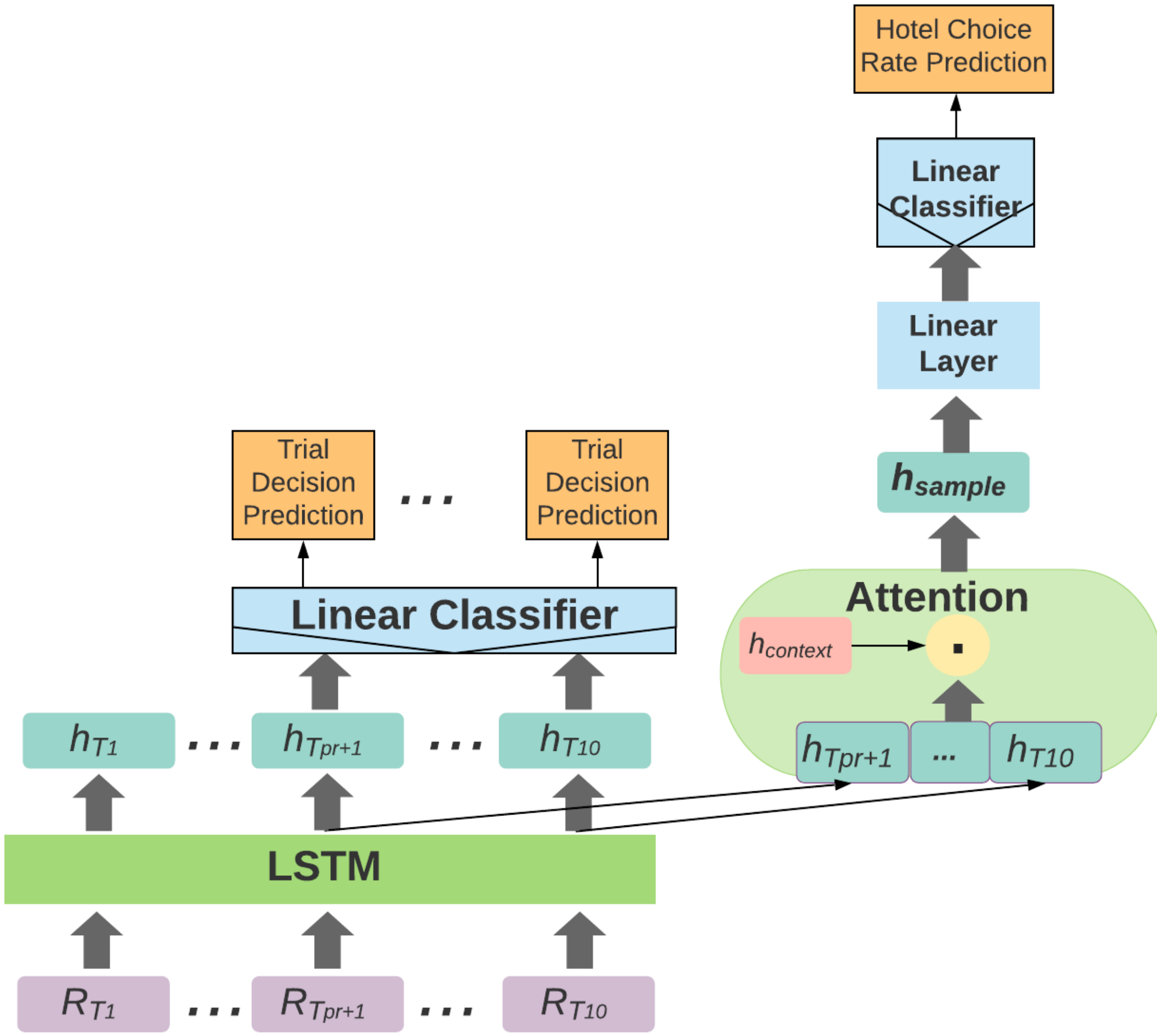


Figure 6: The LSTM-based models. pr denotes the example prefix size and R_{T_i} denotes the representation vector of the i-th trial. The left part is the LSTM-TR model, the right part is the LSTM-CR model, and the complete figure presents the joint LSTM-TRCR model.

each trial, and the hotel choice rate. The LSTM-TRCR architecture, a combination of the above LSTM-TR and the LSTM-CR, is described in Figure 6.

Since the choice rate and the trial labels of each example are strongly related, such that the hotel choice rate label is an average of the trial labels, we augment the above losses with a loss term that is aimed to minimize the squared distance between the predicted choice rate and the average of the individual trial predictions. For this purpose, we calculated the averaged trial predictions using the argmax value of a softmax layer that is fed with the sequence of trial prediction. Formally, given the above notation, and defining $\hat{y}_{TR_{ti}} \in [0, 1]$ to be the prediction of the t-th trial of the i-th example, we define the mean squared trial-choice rate error (MSTRCRE):

$$MSTRCRE = \frac{1}{batch}\sum_{i=1}^{batch}(\hat{y}_{CR_i} - \frac{1}{sf}\sum_{j=pr+1}^{10}\hat{y}_{TR_{ti}})^2$$

Finally, we define the trial-choice rate loss (TRCRL) as the weighted average of three losses, MSE, SCE and MSTRCRE:

$$TRCRL = \alpha \cdot MSE + \beta \cdot SCE + \gamma \cdot MSTRCRE$$

where α, β and γ are hyper-parameters.

The Transformer Models . Another neural network model that has proven to be especially effective for many natural language processing tasks is the Transformer (Vaswani et al., 2017). The Transformer has shown very useful in various NLP tasks, including machine translation (Vaswani et al., 2017, Shaw, Uszkoreit, & Vaswani, 2018) and speech recognition (Dong, Xu, & Xu, 2018), among many others. The Transformer is a sequence-to-sequence model that consists of an encoder and a decoder. In our case, the encoder maps the prefix trials' input sequence to a sequence of continuous representations. Given these representations and an input sequence of the suffix trials, the decoder then generates an output sequence of the suffix trials' representations fed to our model's next layers to generate the output predictions.

Below we describe our Transformer models: Their labels, input vectors and architectures.

By implementing Transformer-based models, we aim to model each input v_{pr} as two sequences: A sequence of the prefix trials and a sequence of the suffix trials, so that to model our task as a translation of the prefix trials to the decisions in the suffix trials. The representations of both the prefix and suffix trials are described above. Since the model's input consists of two sequences, we did not feed it with examples with $pr = 0$. See Figure 7 for an illustration of the model architecture.

Transformer-CR. This is the Transformer model that predicts the hotel choice rate in the suffix, and its architecture is described in the right side of Figure 7. The Transformer is fed with two sequences as described above, and its output is a sequence of sf hidden vectors. These hidden vectors are fed into a dot product attention layer, followed by two linear layers with a dropout layer and a ReLU activation function, in order to predict the hotel choice rate in the suffix trials. The loss function of this model is the MSE loss described above.

Transformer-TR. The Transformer model that predicts the decision in each suffix trial, and its architecture is described in the left side of Figure 7. The output of this model can also be averaged to get the hotel choice rate in the suffix trials. The Transformer is fed with two sequences as described above, and its output is a sequence of sf hidden vectors. Each hidden vector is fed into a linear layer with a dropout layer and a ReLU activation function in order to predict the label for each suffix trial. The loss function of this model is the SCE loss described above.

Transformer-TRCR. This is the Transformer model that jointly predicts the per-trial decision and the overall hotel choice rate. This model is a combination of the two models described above: The Transformer-TR and the Transformer-CR, and its architecture is described in Figure 7. The loss function of this model is the TRCRL loss described above.

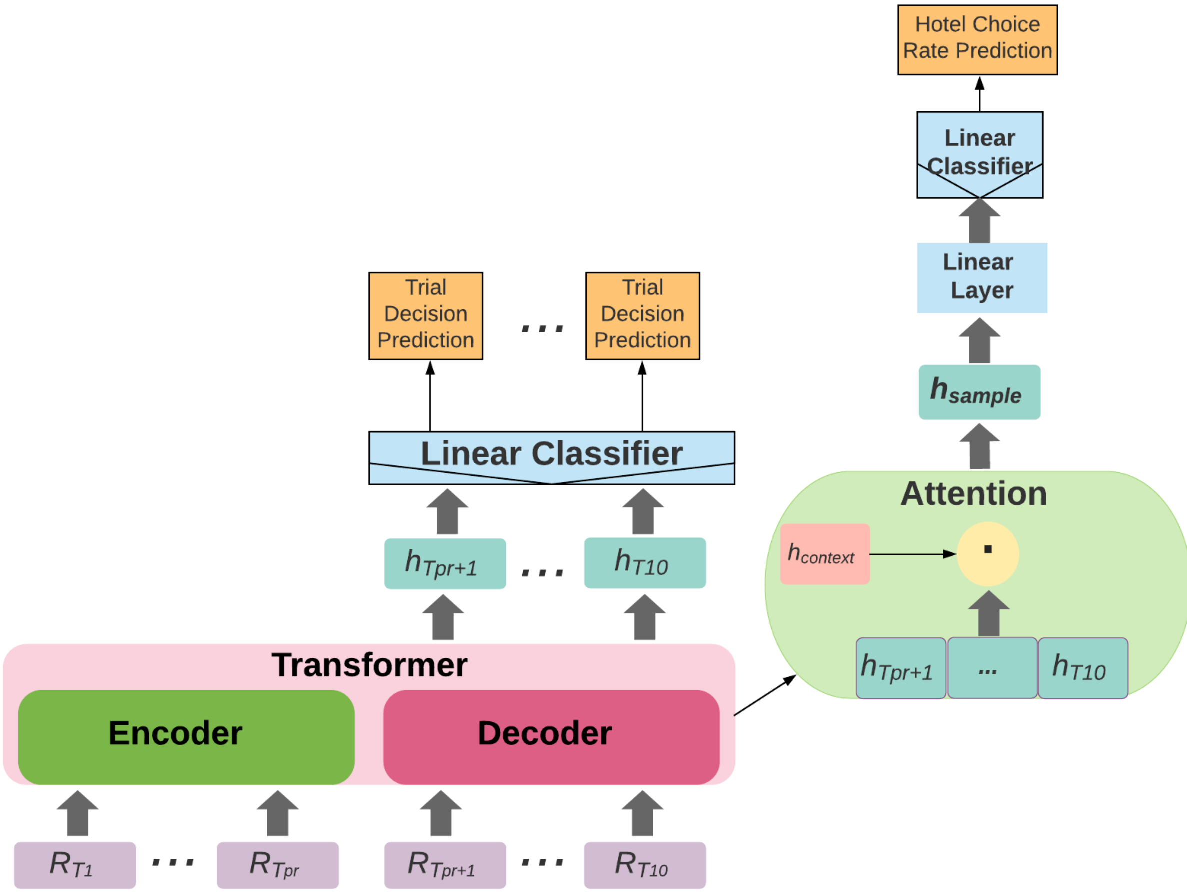


Figure 7: The Transformer-based models. pr denotes the prefix size of the sample, R_{T_i} denotes the representation vector of trial i, and $h_{context}$ is randomly initialized and learned jointly with the attention weights during the training process. The left part is the Transformer-TR model, the right part is the Transformer-CR model, and the entire figure stands for the joint Transformer-TRCR model.

6 Experiments

Recall from Section 4.1 that we have collected a train-validation set of 408 examples and a separate test set of 101 examples. Each example in each set consists of a ten-trial game, and the sets differ in their hotel (and hence also review) sets. We break each example in each of the sets into 10 different examples, such that the first $pr \in \{0, 1, \ldots, 9\}$ trials serve as a prefix and the remaining $sf = 10 - pr$ trials serve as a suffix. This yields a total of 4080 training examples and 1010 test examples.

6.1 Models and Baselines

We consider the following models (described in further detail in Section 5.2): SVM-CR, LSTM-CR, LSTM-TR, LSTM-TRCR, Transformer-CR, Transformer-TR and Transformer-TRCR, where -CR, -TR and -TRCR denote model variants for hotel choice rate, per-trial decisions and joint TR and CR predictions, respectively.

Research Questions Recall our five research questions from Section 3. Our experiments are designed to compare different modeling strategies in learning one of two functions: (1) $F_{trial}()$ and (2) $F_{ChoiceRate}()$. In addition, they are designed to (3) compare different modeling strategies: A non-structured classifier, a sequence model and an attention-based approach. Our experiments are also designed to (4) compare different types of text-based features, and (5) compare between models with text-only and both text and behavior features, in order to explore whether these features have a complementary effect.

Model Variants and Comparisons We perform two sets of comparisons between variants of our models. The comparisons are designed so that to allow us to answer questions #1-#3 about the prediction capacity of our models and their optimal structure, while directly addressing our questions about the optimal textual representation (question #4) and the complementary effect of textual and behavioral features (question #5).

In the first set of comparisons, related to question #4, we consider three variants of each of the models, such that all models use the entire set of features (both textual and behavioral features), but they differ from each other in their textual features: A variant that employs only the $\mathcal{T}_{DNN}$ (BERT-based) features, a variant that uses the $\mathcal{T}_{HC}$ (human crafted) features, and a variant that employs both sets. In the second set of comparisons, related to question #5, we consider two variants of our models: A variant with textual features only (for both the prefix and the suffix trials), and a variant with both the textual and the behavioral features for all trials, similarly to the first set of comparisons. Clearly, these comparisons will also provide us with answers to questions #1-#3.

Note that since we use batch learning models, we use the behavioral features to represent only the prefix trials as they are not known for the suffix trials. Also, since our main focus of this paper is on the use of texts in a persuasion games setup, we do not consider a variant of our model with behavioral features only.

Baselines Similar to our models, the baselines we consider predict either the hotel choice rate or the decision at each trial (and derive the hotel choice rate from the per-trial prediction).

As choice rate baselines, we consider the *Average* (AVG) and the *Median* (MED) baselines. These baselines assign to each example in the test set the average or the median hotel choice rate label, respectively, as computed over the training set examples:

$$AVG = \sum_{i=1}^{|\mathcal{T}|} \frac{y_{CR_i}}{|\mathcal{T}|}$$

where y_{CR_i} is the hotel choice rate label of the i'th example in the training set $\mathcal{T}$. Let $ChoiceRates = \{y_{CR_i} | i \in \mathcal{T}\}$ be the set of the hotel choice rate labels of all the training set examples, the MED baseline assigns the following label to test-set example:

$$MED = \frac{1}{2}(ChoiceRates_{\lfloor(|\mathcal{T}|-1)/2\rfloor} + ChoiceRates_{\lceil(|\mathcal{T}|+1)/2\rceil})$$

We also consider per-trial baselines. The first is the strong *Majority Vote Classifier* (MVC) that assigns to each trial of each test set example the majority decision across all training set trials. Formally, let $y_{TR_{ti}} \in \{0,1\}$ be the trial label of the t-th trial, of the i-th training set example with prefix of size pr_i and suffix of size $sf_i = 10 - pr_i$, and let

$$avg\_TR\_label = \frac{\sum\limits_{i=1}^{|\mathcal{T}|}\sum\limits_{t=pr_i+1}^{10} y_{TR_{ti}}}{\sum\limits_{i=1}^{|\mathcal{T}|} sf_i}.$$

be the average of these labels. The MVC baseline then assigns to each trial in the sequence of each test set example the following prediction:

$$MVC = \begin{cases} 1 \text{ if } avg\_TR\_label \geq 0.5 \\ 0 \ otherwise \end{cases} .$$

Note that this baseline assigns the same label to all the trials of all the test set examples. While one might also like to consider a per-trial majority vote baseline, we noticed that the majority votes of all trial numbers (from 1 to 10) were identical (that is, take the hotel) which makes this baseline identical to our MVC.

In addition, we consider an *Expected Weighted Guess* (EWG) baseline. For this baseline we compute the expected quality of the predictions of a stochastic per-trial classifier, which assigns every test-set trial with one of the two possible labels according to the unigram probability of that label in the training set (when computing this probability we do not distinguish between trial numbers). Note that this is a theoretical classifier that we use in order to put our results in context. Since this classifier is not deterministic, it cannot be applied in practice. We evaluated this theoretical classifier by drawing 5000 assignments for the test-set trials and averaging the resulting values of our evaluation measures.

6.2 Evaluation Measures

We consider three evaluation measures in order to analyze the multiple aspects of our task.

Trial-level measures To evaluate the performance of the various models in predicting the decision at each trial, $F_{trial}()$, we consider the Accuracy-Per-Trial measure, defined as the fraction of test-set trials that are correctly labeled by the algorithm. More specifically, let $y_{TR_{ti}}$ be the trial label of the t-th trial, of the i-th training set example with prefix of size pr_i and suffix of size $sf_i = 10 - pr_j$, and let $\hat{y}_{TR_{ti}}$ be the predicted trial label. The Accuracy-Per-Trial measure is:

$$Accuracy - Per - Trial = \frac{\sum\limits_{i=1}^{|\mathcal{TS}|}\sum\limits_{t=pr_i+1}^{10} \mathbb{1}_{y_{TR_{ti}}=\hat{y}_{TR_{ti}}}}{\sum\limits_{i=1}^{|\mathcal{TS}|} sf_i}$$

We also compute the Macro Average F1-score: We compute the F1-score of each label and report the average of the resulting values. More specifically, let $y_{TR_{ti}}$ be the trial label of the t-th trial, of the i-th training set example with prefix of size pr_i, and let $\hat{y}_{TR_{ti}}$ be its predicted trial label. For each class $i \in ('hotel', 'stay\_home')$ we compute:

$$Precision_i = \frac{\sum_{i=1}^{|\mathcal{TS}|} \sum_{t=pr_i+1}^{10} \mathbb{1}_{y_{TR_{ti}}=\hat{y}_{TR_{ti}}=i}}{\sum_{i=1}^{|\mathcal{TS}|} \sum_{t=pr+1}^{10} \mathbb{1}_{\hat{y}_{TR_{ti}}=i}}, Recall_i = \frac{\sum_{i=1}^{|\mathcal{TS}|} \sum_{t=pr_i+1}^{10} \mathbb{1}_{y_{TR_{ti}}=\hat{y}_{TR_{ti}}=i}}{\sum_{i=1}^{|\mathcal{TS}|} \sum_{t=pr_i+1}^{10} \mathbb{1}_{y_{TR_{ti}}=i}}$$

$$F1_i = \frac{2 \cdot Recall_i \cdot Prediction_i}{Recall_i + Prediction_i}.$$

In other words, $Precision_i$ is the fraction of the examples belonging to class i from those classified as class i, while $Recall_i$ is the fraction of the examples from class i that are classified as class i. The $F1_i$ score is the harmonic average of the precision and recall of the i-th class. The Macro Average F1-score is:

$$Macro - F1 = \frac{1}{2} \sum\nolimits_{i \in ('hotel', 'stay\_home')} F1_i$$

i.e., it is the average of the class-based F1-scores, with equally weighted classes.

Choice rate measures To evaluate the performance of the various models in predicting the choice rate, $F_{ChoiceRate}()$, we employ the Root Mean Square Error (RMSE) measure:

$$RMSE = \frac{1}{|\mathcal{TS}|} \sqrt{\sum_{i=1}^{|\mathcal{TS}|} (\hat{y}_{CR_i} - y_{CR_i})^2}$$

where y_{CR_i} is the choice rate label of the i-th example in the test set $\mathcal{TS}$, and $\hat{y}_{CR_i}$ is the predicted choice rate of that example.

Since most of the participants in our experiment behave similarly to the average participant (see details in Section 4.2), the RSME measure would not indicate that a model fails in capturing behaviors that deviate from the average behaviour. We hence perform a bin analysis, mapping the choice rates into four bins:

- Bin 1: $choice\ rate < 0.25$
- Bin 2: $0.25 \leq choice\ rate < 0.5$
- Bin 3: $0.5 \leq choice\ rate < 0.75$
- Bin 4: $choice\ rate \geq 0.75$

We then compute the Macro Average F1-score over these bins. More specifically, let y_{bin_i} be the bin label of the i-th example in the test set $\mathcal{TS}$, and let $\hat{y}_{bin_i}$ be the predicted bin of that example. For each bin $j \in (1, 2, 3, 4)$ we compute:

$$Precision_j = \frac{\sum_{i=1}^{|\mathcal{TS}|} \mathbb{1}_{\hat{y}_{bin_j}=y_{bin_j}=j}}{\sum_{i=1}^{|\mathcal{TS}|} \mathbb{1}_{\hat{y}_{bin_j}=j}}, Recall_j = \frac{\sum_{i=1}^{|\mathcal{TS}|} \mathbb{1}_{\hat{y}_{bin_i}=y_{bin_i}=j}}{\sum_{j=1}^{|\mathcal{TS}|} \mathbb{1}_{y_{bin_i}=j}}$$

$$F1_j = \frac{2 \cdot Recall_i \cdot Prediction_i}{Recall_i + Prediction_i}.$$

The Macro Average F1-score is then:

$$Bin - Macro - F1 = \frac{1}{4} \sum_{j=1}^{4} F1_i$$

Note that we evaluate the choice rate models using the RMSE and the bin analysis measures only. In contrast, we evaluate the per-trial and the joint models using all our evaluation measures, because the choice rate can be derived from the predictions of the per-trial models.

6.3 Cross Validation

We employ a six-fold cross-validation protocol in order to tune the hyper-parameters of each model. For this purpose, we split the 408 (expert, decision-maker) pairs of the train-validation set into six subsets, such that each subset consists of 68 pairs. As described above, each decision sequence is translated into ten examples, each with a different prefix size, resulting in 680 examples in each subset. In each fold, we select one subset for development and the remaining five subsets serve for training. Each model is then trained on the training set, and its hyper-parameters are tuned on the development set so that to minimize its RMSE loss. The tuned models are then applied to the held-out test set. Finally, for each model we report the average results across the six folds.

6.4 Hyper-parameter Tuning

We next describe the hyper-parameters of each of the models.

SVM We use the standard Support Vector Regression (SVR) model of the sklearn package to predict the hotel choice rate.[6] We use the default values for all the model hyper-parameters and tune the type of kernel function (rbf, linear, polynomial) as well as the polynomial degree of the kernel function (3, 5, 8).

DNNs For all DNNs, we use ReLU as the activation function for all internal layers, and we tune the dropout parameter (0.0, 0.1, 0.2, 0.3), such that the same dropout parameter was used in the LSTM and Transformer models, as well as in the linear layers placed on top of these models. Training is carried out for 100 epochs with early stopping, and a batch size of 10 in the LSTM-based models and 9 in the Transformer-based models. Each batch consisted of all the examples of one decision-maker. We use a different batch size for each model, since we did not feed the Transformer with examples with prefix of size 0, as mentioned in Section 5.2, and we still want to have examples of only one decision-maker in each batch. We use the ADAM optimization algorithm (Kingma & Ba, 2015) with its default parameters as implemented in Pytorch: learning rate=$1e^{-03}$, fuzz factor $\epsilon = 1e^{-08}$, and learning rate *decay* over each update=0.0. We developed the DNNs with the AllenNLP software package (Gardner, Grus, Neumann, Tafjord, Dasigi, Liu, Peters, Schmitz, & Zettlemoyer, 2018)[7] over Pytorch (Paszke, Gross, Chintala, Chanan, Yang, DeVito, Lin, Desmaison, Antiga, & Lerer, 2017).

[6]`https://github.com/scikit-learn/scikit-learn/tree/master/sklearn/svm`
[7]https://allennlp.org/

For our LSTM-based models we tune the LSTM hidden layer size (50, 80, 100, 200) and the number of LSTM layers (1, 2, 3). For our Transformer-based models, we tune the number of encoder and decoder layers (3, 4, 5, 6). In addition, we tune the dimension of the linear layers that follow the multi head attention layer of the encoder and decoder (0.5, 1, and 2 times of the input dimension), such that these parameters are the same for the encoder and the decoder. Finally, for the joint models, we tune the weight of the MSE, SCE and MSTPE losses ($(\alpha, \beta, \gamma) \in ((1,1,1),(2,2,1),(1,1,2))$).

7 Results

As mentioned in Section 6, we perform two sets of comparisons in order to address the five research questions posed in Section 3, and in this section we will present their results. Both sets aim to answer the questions regarding our ability to predict the hotel choice rate and the per-trial decisions (questions #1 and #2, respectively), while comparing different modeling strategies (question #3). The first set of comparisons focuses on the question that deals with text representations (question #4), while the second set focuses on the complementary value of textual and behavioral representations (question #5).

7.1 Per-Trial Prediction Results

Table 6 presents the per-trial accuracy and macro average F1-score results for the various baselines and models, when using both the textual and the behavioral features. The table is divided to four sub-tables: The top table reports the results of our models when using the hand-crafted textual features, the next table reports the results of our models when using the BERT-based textual features, the next table reports the results of our models when using a concatenation of both BERT and our hand-crafted textual features, and the bottom table reports the results of the baselines.

The results show that the joint trial/choice-rate LSTM-TRCR model with our hand-crafted features is superior to the other models and baselines. The results also show that the Transformer-based models using BERT-based textual features, either alone or together with the hand-crafted features, do not perform well.

In fact, they produce the same output as the MVC baseline. Note, however, that when the Transformer models are used with our hand-crafted features only, their performance substantially improves (e.g. the Transformer-TRCR model with our hand-crafted features is the second best model in terms of the macro average F1-score). These results provide a positive answer to our first research question by showing that our models, when using the hand-crafted textual features, perform better than the baselines according to both evaluation measures, especially in terms of the macro average F1-score.

We next show that the main reason for the superiority of our models is that they can predict, at least with a decent accuracy, the cases where the decision-maker deviates from its majority behavior (which is choosing the hotel). Table 7 presents the F1-scores of each of the classes: 'hotel' and 'stay at home'. Results are presented for the baselines and two of our best models: LSTM-TR and LSTM-TRCR with our hand-crafted features. It demonstrates that the LSTM-TRCR model performs almost as well as the baselines, when considering the F1-score of the 'hotel' class. It also shows that both models succeed in predicting some of the cases in which the decision-maker chooses to stay home, while the baselines perform poorly on this class (for the MVC baseline the score is 0 by definition, since this is not the majority decision in the training data).

Model	Per-Trial Accuracy	Macro Average F1-score
Hand Crafted Features ($\mathcal{T}_{HC}$)		
LSTM-TR	76.5	70
LSTM-TRCR	**76.9**	**70.5**
Transformer-TR	75.3	60.1
Transformer-TRCR	76.2	65.5
BERT-based Features ($\mathcal{T}_{DNN}$)		
LSTM-TR	61.3	51.8
LSTM-TRCR	62.6	51.5
Transformer-TR	73.3	42.3
Transformer-TRCR	73.3	42.3
BERT-based Features + Hand Crafted Features		
LSTM-TR	70.2	60
LSTM-TRCR	71.8	61.2
Transformer-TR	73.3	42.3
Transformer-TRCR	73.3	42.3
Baselines		
MVC	73.3	42.3
EWG	59.8	50

Table 6: Per-trial performance with both textual and behavioral features.

Model	F1-score 'Hotel'	F1-score 'Stay Home'
LSTM-TR	83.8	56.1
LSTM-TRCR	84.2	**56.7**
MVC	**84.6**	0
EWG	72.1	27.8

Table 7: Per-Class F1-scores for the per-trial analysis. LSTM-TR and LSTM-TRCR use the hand-crafted textual features and the behavioral features.

7.2 Hotel Choice Rate Results

Table 8 presents our hotel choice rate results in terms of RMSE and bin macro average F1-score, when our models use both the textual and the behavioral features. The table is divided into four sub-tables, similarly to Table 6.

When considering the RMSE measure, the median (MED) and average (AVG) baselines, as well as the LSTM-CR model with our hand-crafted textual features, are superior. Also, the Transformer-CR and Transformer-TRCR models do not lag substantially behind. As discussed in Section 4.2, the behavior of most of our participants is similar to the average behavior in the entire group of participants. Hence, it is not surprising that the MED and AVG baselines excel under RMSE evaluation.

In contrast, when considering the bin macro average F1-score, LSTM-TR with our hand-crafted textual features outperforms all other models and baselines, by a large margin (e.g. 48.3 compared to only 13.2 for the AVG baseline). Generally, all our models but one substantially outperform the MVC, AVG and MED baselines on this measure, when using the hand-crafted features only, and the same holds for the LSTM models when using the BERT features (with or without the hand-crafted features). These results provide a positive answer to our second research question by showing that our models can indeed learn to perform hotel choice rate prediction quite well.

Model	RMSE	Bin Macro Average F1-score
Hand Crafted Features ($\mathcal{T}_{HC}$)		
SVM-CR	29.8	26.6
LSTM-TR	26.6	**48.3**
LSTM-CR	**24.6**	34.5
LSTM-TRCR	29.7	31.7
Transformer-TR	33.1	33.9
Transformer-CR	25.2	16.7
Transformer-TRCR	25.3	31.6
BERT-based Features ($\mathcal{T}_{DNN}$)		
SVM-CR	27.2	20.8
LSTM-TR	34	32.8
LSTM-CR	28.4	27.7
LSTM-TRCR	33.8	23.9
Transformer-TR	37.2	16.3
Transformer-CR	25.5	12.3
Transformer-TRCR	25.4	12.1
BERT-based Features + Hand Crafted Features		
SVM-CR	27.4	21.9
LSTM-TR	30.4	39.2
LSTM-CR	25.6	27.5
LSTM-TRCR	28.3	26.2
Transformer-TR	37.2	16.3
Transformer-CR	25.5	12.8
Transformer-TRCR	25.4	12.3
Baselines		
MVC	36.5	17.7
EWG	34.7	32.4
AVG	**24.6**	13.2
MED	**24.6**	16.2

Table 8: Hotel choice rate prediction performance of the baselines and the models with the textual and behavioral features. Note that for RMSE lower values are better, while for the bin macro average F1-score higher values are better.

As in the per-trial prediction, the main reason for the superiority of our models is their ability to predict deviations from the majority behavior. Table 9 presents the F1-scores of each of the bins defined in Section 6.2, both for the baselines and for the models with our hand-crafted textual features. It demonstrates that LSTM-TR outperforms all other models and baselines on two of the bins that relate to the non-majority behavior, while on the third non-majority bin Transformer-TR performs best and LSTM-TR is second best. The AVG baseline, in contrast, performs best on the majority behavior bin.[8]

Model	**F1-score Bin 1**	**F1-score Bin 2**	**F1-score Bin 3**	**F1-score Bin 4**
SVM-CR	21	18	44.6	23
LSTM-TR	**50.9**	**33.1**	47.9	61.4
LSTM-CR	26.4	15.3	42.8	53.5
LSTM-TRCR	26	15.6	26.7	58.4
Transformer-TR	24.5	21.1	24.5	**65.5**
Transformer-CR	0	1.8	39.2	25.8
Transformer-TRCR	16.2	11.2	43	55.8
AVG	0	0	**52.8**	0
MED	0	0	17.6	47.1

Table 9: Hotel choice rate F1-scores for the four bins. The table presents the performance of the baselines and the models with the hand-crafted textual features and the behavioral features. The hotel choice rate mapping into the bins is: Bin 1: $\textit{choice rate} < 0.25$, Bin 2: $0.25 \leq \textit{choice rate} < 0.5$, Bin 3: $0.5 \leq \textit{choice rate} < 0.75$, and Bin 4: $\textit{choice rate} \geq 0.75$.

Focusing on the results of the LSTM-based models, the choice rate model (LSTM-CR) outperforms the per-trial model (LSTM-TR) and the joint model (LSTM-TRCR) when considering the RMSE score, but the per-trial (LSTM-TR) model outperforms the other two models when considering the bin macro average F1-score. These patterns hold regardless of the type of textual features used by the models. These results and the results presented in Table 9, indicate that directly optimizing for the hotel choice rate is particularly useful for the overall RMSE performance. However, trial-based models better capture less frequent outcomes. Moreover, joint learning of both trial outcomes and the overall choice rate (with the LSTM-TRCR model) does not improve over learning only the trial-based outcome.

7.3 The Impact of Different Textual Feature Sets

Figures 8 and 9 compare the performance of our models when using the different textual feature sets and the behavioral feature set. Figure 8 presents the hotel choice rate results, and it shows that except for SVM-CR, all our models achieve their best RMSE score (left part of the figure) when using our hand-crafted features. Likewise, the right part of Figure 8 indicates that all our models excel on the bin macro average F1-score when using the hand-crafted textual features, and the gaps are even larger. Figure 9 presents very similar patterns for the per-trial prediction models. While

[8]Although the MED baseline assigns to each test-set example the median hotel choice rate, as computed over the training set examples, it achieves a positive F1-score for two bins. This is because in our six fold cross-validation protocol, in two folds the median hotel choice rate of the training set examples falls within the third bin, while in the other four folds it falls within the fourth bin.

the figures focus on models that use both the textual and the behavioral features, we observed very similar patterns when comparing models that use only the textual features. These results provide an answer to our fourth research question as they clearly indicate the value of our hand-crafted features, even when compared to a strong language representation model, like BERT.

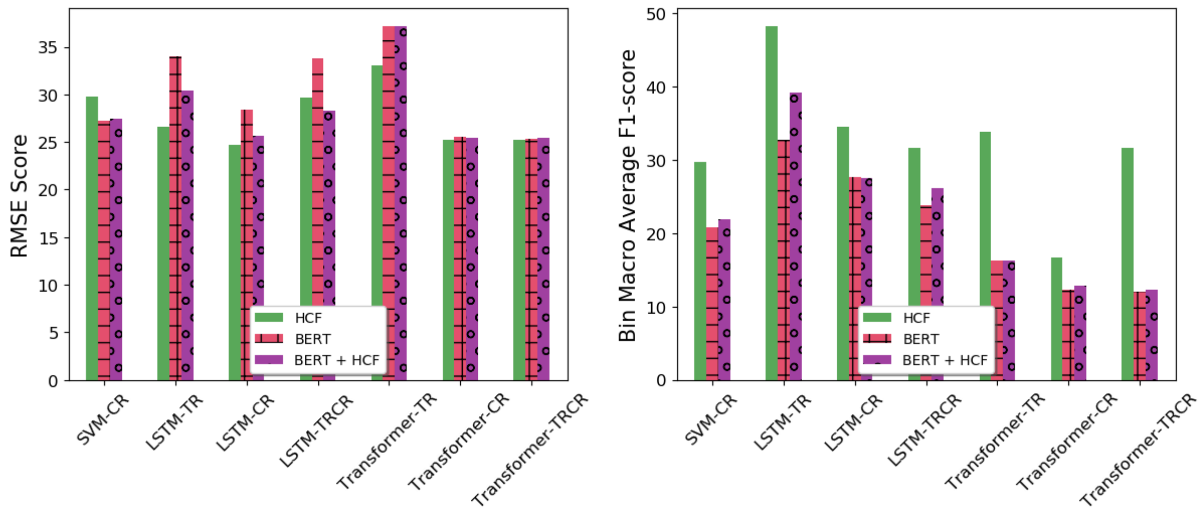


Figure 8: Textual features comparison for the hotel choice rate prediction task. The histogram presents the RMSE score (lower is better) and the bin macro average F1-score (higher is better) of each of our models, when using the behavioral features and each of our textual feature sets (HCF stands for 'hand-crafted features').

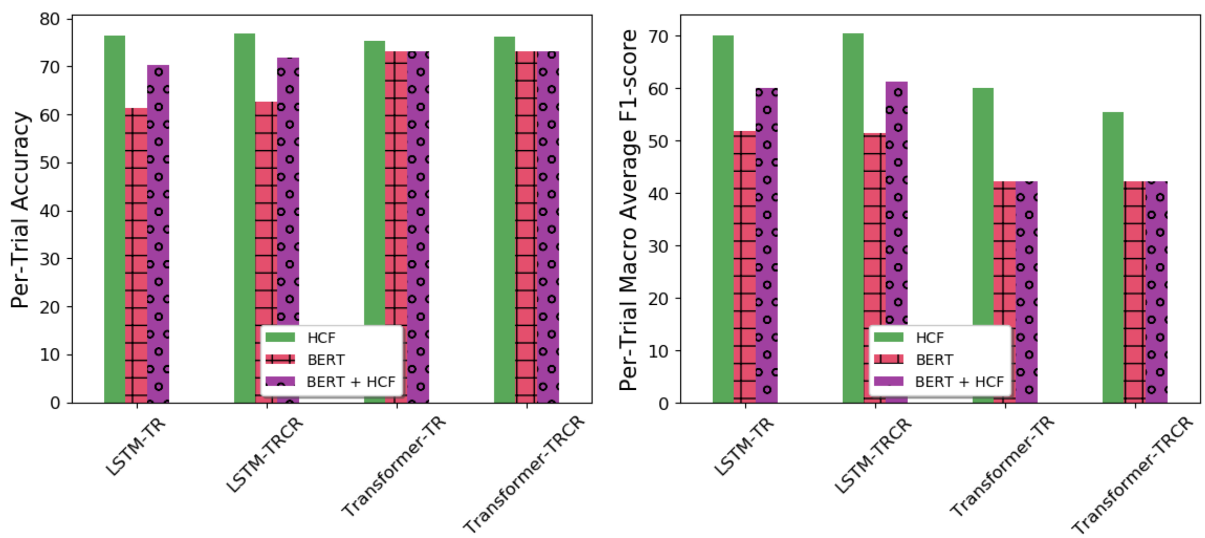


Figure 9: Textual features comparison for the per-trial prediction models. The histogram presents the per-trial accuracy (left) and the macro average F1-score (right) of each of our models, when using the behavioral features and each of our textual feature sets.

7.4 The Complementary Impact of Textual and Behavioral Features

Table 10 presents the results of our set of comparisons which focuses on Question #5, regarding the impact of the different feature sets: Behavioral and textual. As indicated above, in most cases our models perform best with our hand-crafted textual features. We hence focus on these textual features in the current comparison.

Here, the table provides a mixed answer, where for some models and evaluation measures the joint feature set yields superior results, while for others it is better to use only the textual features. These results suggest that we should still deepen our understanding of the complementary effect of the two feature sets. This is a clear direction for future work.

Model	**Feature sets**	**RMSE**	**Bin Macro Average F1-score**	**Per-Trial Accuracy**	**Per-Trial Macro Average F1-score**
SVM-CR	Only Textual Features	**29.0**	**29.5**	-	-
SVM-CR	All Features	29.8	26.6	-	-
LSTM-TR	Only Textual Features	27	47	**77.6**	**71.2**
LSTM-TR	All Features	**26.6**	**48.3**	76.5	70
LSTM-CR	Only Textual Features	26.6	**34.7**	-	-
LSTM-CR	All Features	**24.7**	34.5	-	-
LSTM-TRCR	Only Textual Features	**25.9**	27.7	**78.8**	**72**
LSTM-TRCR	All Features	29.7	**31.7**	76.9	70.5
Transformer-TR	Only Textual Features	33.2	31.7	**75.5**	57.7
Transformer-TR	All Features	**33.1**	**33.9**	75.3	**60.1**
Transformer-CR	Only Textual Features	**25.1**	**21.8**	-	-
Transformer-CR	All Features	25.2	16.7	-	-
Transformer-TRCR	Only Textual Features	**25.1**	**32.8**	75.6	64.2
Transformer-TRCR	All Features	25.3	31.6	**76.2**	**65.5**

Table 10: A comparison between all models with hand-crafted textual features (from both the prefix and suffix trials), behavioral features (from the prefix trials), as well as with a joint feature set. For each model and measure separately, we bold the feature set that outperforms the others.

8 Ablation Analysis

In Section 7, we addressed our five pre-defined research questions. In this section, we address additional aspects of our data, experiments, and results. In particular, we would like to discuss the quality of the models and baselines in predicting the labels of examples with various prefix sizes, and their quality in predicting the decisions in various stages of the interaction between the expert and the decision-maker.

8.1 Prefix Size Impact

In this part, we analyze the impact of the prefix size on our models. We are focusing on models that use behavioral features and hand-crafted textual features, that have demonstrated very effective in Section 7. If the prefix size affects the performance of our models, we may want to consider this parameter in the model selection process.

Figure 10 presents the hotel choice rate performance, measured by RMSE (left) and bin macro average F1-score (right), as a function of the prefix size. The graphs indicate that while the RMSE score increases with the prefix size, for the bin macro average F1-score there is no strong correlation between the prefix size and the performance of the models, except for LSTM-TR, Transformer-TR, and the EWG and MVC baselines.

Figure 11 complements this hotel choice rate analysis and presents the per-trial accuracy (left) and the macro average F1 (right) of the per-trial and the joint models, as a function of the prefix size. The results show that the per-trial accuracy of the baselines somewhat decreases with the prefix size and are consistently worse than the performances of our models. In contrast to the choice rate analysis, in this case there is no strong correlation between the prefix size and the performance of the models.

The different correlations presented in these two figures, and particularly the difference between their left graphs, indicate that the prefix size has an impact when predicting the hotel choice rate, but not when predicting the decision at each trial. Understanding this difference is a potential important direction of future research.

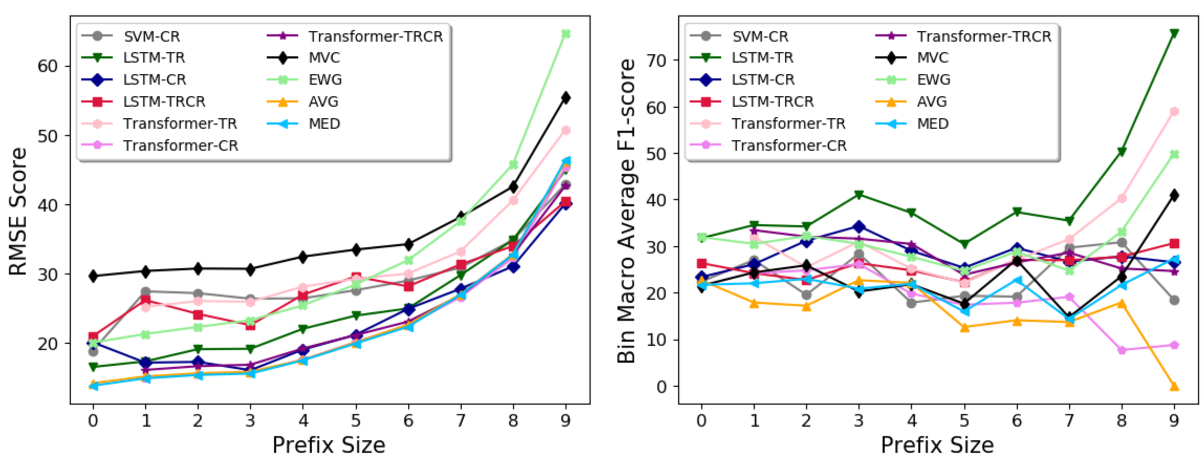


Figure 10: Prefix size impact on hotel choice rate performance. The graphs present the RMSE score (left) and the bin macro average F1-score (right) as a function of the prefix size.

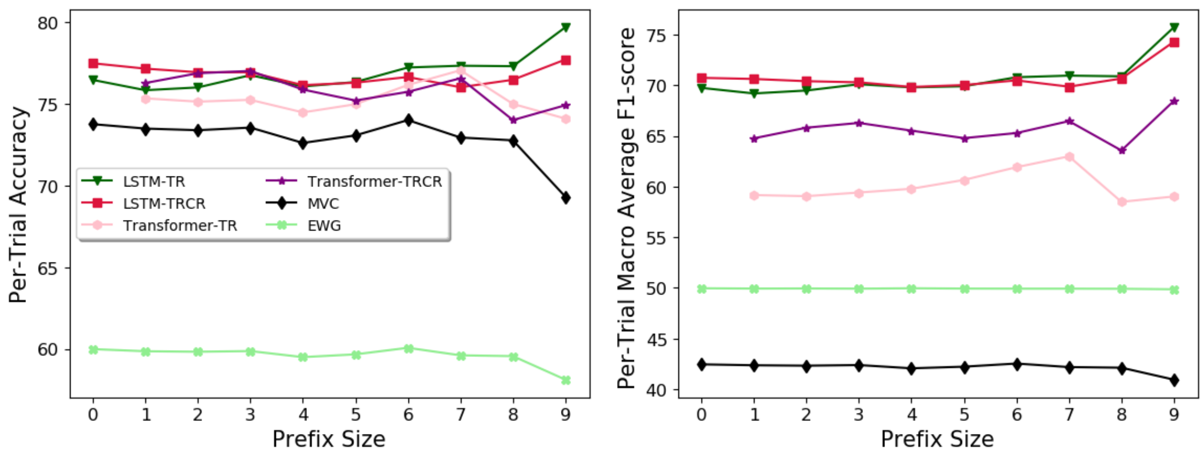


Figure 11: The effect of the prefix size on per-trail prediction performance. The graphs present the per-trial accuracy (left) and the macro average F1-score (right) as a function of the prefix size.

8.2 Trial Number Impact

Figure 12 presents the per-trial accuracy (left) and the macro average F1-score (right) as a function of the trial number. Our motivation for this analysis is exploring the temporal dynamic between the expert and the decision-maker and the development of their mutual trust. In Section 4, we show that the hotel choice rate changes as the experiment progresses, but the changes are very small. We also show that the decision in each trial depends on the decision and the feedback in the previous trial. Here we explore whether temporal patterns can also be observed in the predictions of our models.

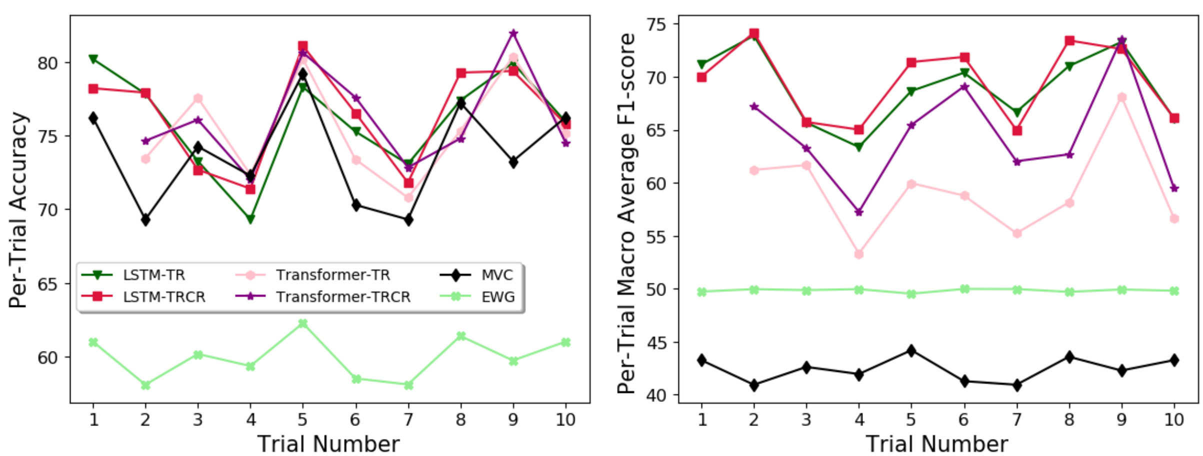


Figure 12: The trial number impact on the per-trial prediction results. The graphs present the per-trial accuracy (left) and the macro average F1-score (right) as a function of the trial number.

The figures demonstrate a temporal dynamic in the predictions of our models, although not necessarily an expected one. Particularly, the performance seem to have a periodical behavior such that after a maximum or a minimum point is achieved, the performance starts to move in the opposite direction. Since this is also the pattern we observe for the MVC baseline, these results seem to be explained by the pattern of human deviation from the majority behavior. Explaining this pattern is another interesting direction of future work.

9 Discussion

We explored the task of predicting the decisions made in repeated persuasion games. For this purpose, we conducted an online experiment with more than 500 pairs of participants. In contrast to previous behavioral economics work where the experts and decision-makers communicate through numerical messages, the messages are verbal in our setting. We explored five research questions, and we will next summarize the main findings and conclusions related to each.

The **first** question focuses on our ability to predict the decisions in each trial given the history of *pr* trials, where *pr* is an integer number between 0 and 9. We demonstrated that DNN modeling combined with our hand-crafted textual features as well as behavioral features (LSTM-TR) is the superior modeling strategy. Interestingly, when considering the per-trial accuracy evaluation measure, the MVC baseline is comparable to our strongest models. In contrast, our models are superior when the evaluation is done with the macro average F1-score. This is mostly because the baselines fail to detect the less dominant class in our data - decisions not to choose the hotel. This failure makes them poor prediction models that only excel in detecting the dominant class in the data, while our best models can go beyond that and also predict minority classes.

Our **second** research question targets our models' ability to accurately predict the hotel choice rate in a given suffix of a communication sequence. More than half of the participants select the hotel choice in seven or eight trials, and the average hotel choice rate was 7.18, with a standard deviation of 1.52. Therefore, predicting the average or median hotel choice rate, in our case, is a good approximation of the behavior of most participants. Indeed, in terms of the RMSE score, the simple MED and AVG baselines were the superior modeling strategies (together with LSTM-CR). However, unlike our models, these baselines fail to predict decisions that deviate from our population's average behavior. This is reflected by the superior bin macro average F1-score performance of our models. As in our findings for the first question, we observe that our models are valuable for predicting deviations from the population's most frequent behaviors.

Our **third** research question aims to identify the ideal modeling strategy for our setup. We explored three strategies: a non-structured classifier, a sequence model, and an attention-based approach. We show that in the per-trial prediction task, LSTM-TRCR outperforms the other models and baselines. We also show that in the hotel choice rate prediction task, LSTM-CR outperforms the other models in terms of the RMSE score, and LSTM-TR outperforms the other models and baselines in terms of the bin macro average F1-score. These results indicate that a sequence model is the ideal modeling strategy for our setup. We hypothesize that the attention-based models are inferior to the sequence models in our setup, due to the relatively small training data set which is particularly challenging for highly parameterized models such as the transformer. Moreover, we show that the SVM-CR non-structured classifier is consistently inferior to the other models, emphasizing the need for DNN modeling in our setup.

We have further considered models (both sequential and attention-based) that jointly consider the hotel choice rate and the per-trial decisions. These models achieved good results in both tasks and outperformed the other models and baselines in the per-trial prediction.

The **fourth** question is designed to find the textual features that most serve our prediction model. We compared two sets of textual features: hand-crafted and DNN-based (BERT). We show that all our models, except for SVM-CR, achieved their best results when using the hand-crafted features, regardless of the evaluation measure used. This may be an indication that BERT is not capable of capturing the high-level task-related semantics of the task, as encoded in our hand-crafted features.

Finally, our **fifth** question focuses on the different aspects of the data which are crucial for our prediction. Particularly, we explored the impact of the textual messages, and the complementary effect of both the textual messages and the decision-makers' behavior throughout the game. Interestingly, our results indicate a mixed answer to this question, with different models performing best with different feature sets. Hence, we cannot provide a conclusive answer to this question.

In this work, we have chosen to focus on predicting the decision-maker's decisions. These decisions partially determine the payoff of both the expert and the decision-maker, although the latter's payoff also depends on a random coin flip. Naturally, in the scope of one paper we could not focus on other important tasks, such as predicting the expert's decisions, or on predictions that would give in-depth insights about our setup, such as predicting the hotel choice rate as a function of the hotel's expected value. We leave these as well as other important questions for future work, hoping that our novel task and data set will pave the way for other researchers to explore the use of natural language in persuasion games.

One challenging aspect of our work is the generalization to new hotels. As described in Section 4.1, we used one set of hotels in the train and development set and another set of hotels in the test set. Therefore, our models should generalize across different reviews in order to perform well on the test set. We also performed experiments where the test set has the same set of hotels and reviews as the training and development set. In the hotel choice rate prediction task, the results show that the SVM-CR model, when using the hand-crafted textual features and the behavioral features, outperforms all other models and baselines in terms of both measures. The results also show that LSTM-CR and LSTM-TRCR, when using the hand-crafted textual features and the behavioral features, outperform the baselines on both measures. This is in contrast to the main results of this paper, where cross-hotel generalization is required and the AVG and MED baselines achieve the best choice rate RMSE score (together with LSTM-CR), while LSTM-TR performs best in the choice rate bin analysis. For the per-trial decision task, the results are similar to those we show in Section 7 in terms of the relative ranking of the models and the baselines, with slightly better performance of our models and slightly worse performance of the baselines. This comparison shows that the LSTM-based models excel in both conditions, indicating their robustness.

Another particular challenging aspect of our work, which may be considered a limitation, is the use of lab data, as opposed to real-world interactions. On the one hand, the use of lab data lets us control the various aspects of our experiments and allow us to draw intricate conclusions. On the other hand, previous studies revealed an interesting gap when comparing lab and field studies of social interactions (see review in Levitt & List, 2007 and one demonstration of this gap in Gneezy, Haruvy, & Yafe, 2004). Therefore, one of our main directions for future work is to explore whether our results generalize to real-world setups.

We would finally like to highlight two future work directions that seem of particular interest to us. The first direction has to do with online learning. While we predict the future behavior of the decision maker under a pre-defined policy of the expert (i.e. a pre-defined set of reviews chosen to describe the hotels in the suffix), in online learning the policy of both agents may change as a function of the input they receive from the world (e.g. the random hotel score (feedback) observed by both agents after the decision-maker makes her decision). In contrast to our task that focuses on pre-defined policy evaluation, online learning provides more dynamic policy adjustment capabilities. The second direction goes even further and utilizes our data in order to learn automatic agents that can simulate either the expert or the decision maker, performing their task optimally under a pre-defined criterion (e.g. maximizing the agent's gains or, alternatively, the mutual benefits of both players). Moving from behavior prediction to generation reflects a deeper understanding of our task and data. As noted above, we hope that our task, data and algorithms will draw the attention of the research community to language-based persuasion games and will facilitate further research on these questions as well as others.

Acknowledgement

The work of R, Apel and M. Tennenholtz is funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 740435).

References

Altman, A., Bercovici-Boden, A., & Tennenholtz, M. 2006. Learning in one-shot strategic form games. In Fürnkranz, J., Scheffer, T., & Spiliopoulou, M. (Eds.), *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, Vol. 4212 of *Lecture Notes in Computer Science*, pp. 6–17. Springer.

Arieli, I., & Babichenko, Y. 2019. Private bayesian persuasion. *Journal Economic Theory*, *182*, 185–217.

Aumann, R. J., Maschler, M. B., & Stearns, R. E. 1995. *Repeated games with incomplete information*. MIT Press.

Azaria, A., Rabinovich, Z., Kraus, S., & Goldman, C. V. 2011. Giving advice to people in path selection problems. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Azaria, A., Rabinovich, Z., Kraus, S., Goldman, C. V., & Gal, Y. 2012. Strategic advice provision in repeated human-agent interactions. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Bahar, G., Smorodinsky, R., & Tennenholtz, M. 2016. Economic recommendation systems: One page abstract. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pp. 757–757.

Ben-Porat, O., Hirsch, S., Kuchi, L., Elad, G., Reichart, R., & Tennenholtz, M. 2020. Predicting strategic behavior from free text. *Journal of Artificial Intelligence Research*, *68*, 413–445.

Beyth-Marom, R. 1982. How probable is probable? a numerical translation of verbal probability expressions. *Journal of forecasting*, *1*(3), 257–269.

Budescu, D. V., Weinberg, S., & Wallsten, T. S. 1988. Decisions based on numerically and verbally expressed uncertainties.. *Journal of Experimental Psychology: Human Perception and Performance*, *14*(2), 281.

Camerer, C. F., Ho, T.-H., & Chong, J.-K. 2004. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, *119*(3), 861–898.

Carlile, W., Gurrapadi, N., Ke, Z., & Ng, V. 2018. Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 621–631.

Chatterjee, M., Park, S., Shim, H. S., Sagae, K., & Morency, L.-P. 2014. Verbal behaviors and persuasiveness in online multimedia content. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, pp. 50–58.

Chen, D. L., Schonger, M., & Wickens, C. 2016. otree—an open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance*, *9*, 88–97.

Cortes, C., & Vapnik, V. 1995. Support-vector networks. *Machine learning*, *20*(3), 273–297.

Costa-Gomes, M., Crawford, V. P., & Broseta, B. 2001. Cognition and behavior in normal-form games: An experimental study. *Econometrica*, *69*(5), 1193–1235.

Crawford, V. P., & Sobel, J. 1982. Strategic information transmission. *Econometrica*, *50*(6), 1431–1451.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

Dong, L., Xu, S., & Xu, B. 2018. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888. IEEE.

Emek, Y., Feldman, M., Gamzu, I., PaesLeme, R., & Tennenholtz, M. 2014. Signaling schemes for revenue maximization. *ACM Transactions on Economics and Computation (TEAC)*, *2*(2), 1–19.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M. E., Schmitz, M., & Zettlemoyer, L. 2018. Allennlp: A deep semantic natural language processing platform. *CoRR*, *abs/1803.07640*.

Gill, A. J., Nowson, S., & Oberlander, J. 2009. What are they blogging about? personality, topic and motivation in blogs. In *Third International AAAI Conference on Weblogs and Social Media*.

Gneezy, U., Haruvy, E., & Yafe, H. 2004. The inefficiency of splitting the bill. *The Economic Journal*, *114*(495), 265–280.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. 2016. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222–2232.

Hartford, J. S., Wright, J. R., & Leyton-Brown, K. 2016a. Deep learning for predicting human strategic behavior. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 29*, pp. 2424–2432. Curran Associates, Inc.

Hartford, J. S., Wright, J. R., & Leyton-Brown, K. 2016b. Deep learning for predicting human strategic behavior. In *Advances in Neural Information Processing Systems*, pp. 2424–2432.

Hochreiter, S., & Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, *9*, 1735–80.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*.

Joshi, M., Das, D., Gimpel, K., & Smith, N. A. 2010. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 293–296. Association for Computational Linguistics.

Jurafsky, D., & Martin, J. H. 2018. *Speech and language processing*. Pearson.

Kamenica, E., & Gentzkow, M. 2011. Bayesian persuasion. *American Economic Review*, *101*, 2590–2615.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *CoRR*, *abs/1408.5882*.

Kingma, D. P., & Ba, J. 2015. Adam: A method for stochastic optimization. In Bengio, Y., & LeCun, Y. (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., & Smith, N. A. 2009. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Kolumbus, Y., & Noti, G. 2019. Neural networks for predicting human interactions in repeated games..

Kraus, S., & Arkin, R. C. 2001. *Strategic negotiation in multiagent environments*. MIT press.

Kreps, D. M., Milgrom, P., Roberts, J., & Wilson, R. 1982. Rational cooperation in the finitely repeated prisoners dilemma. *Journal of Economic Theory*, *27*(2), 245–252.

Levitt, S. D., & List, J. A. 2007. What do laboratory experiments measuring social preferences reveal about the real world?. *Journal of Economic perspectives*, *21*(2), 153–174.

Malca, R., & Reichart, R. 2018. Neural transition based parsing of web queries: An entity based approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2700–2710, Brussels, Belgium. Association for Computational Linguistics.

Niculae, V., Kumar, S., Boyd-Graber, J., & Danescu-Niculescu-Mizil, C. 2015. Linguistic harbingers of betrayal: A case study on an online strategy game. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJNLP)*.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., Smith, N. A., et al. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Fourth international AAAI Conference on Weblogs and Social Media (ICWSM)*.

Oved, N., Feder, A., & Reichart, R. 2020. Predicting in-game actions from interviews of nba players. *Computational Linguistics*, *46*(3), 667–712.

Pang, B., & Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Peled, N., Gal, K., & Kraus, S. 2012. Learning to reveal information in repeated human-computer negotiation. *Institute for Advanced Computer Studies. University of Maryland*, *1500*, 20742.

Persing, I., & Ng, V. 2017. Why can't you convince me? modeling weaknesses in unpersuasive arguments.. In *IJCAI*, pp. 4082–4088.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. 2018. Deep contextualized word representations. *CoRR*, *abs/1802.05365*.

Plonsky, O., Erev, I., Hazan, T., & Tennenholtz, M. 2017. Psychological forest: Predicting human behavior. In Singh, S. P., & Markovitch, S. (Eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 656–662. AAAI Press.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. 2018. Improving language understanding by generative pre-training..

Reichart, R., Tomanek, K., Hahn, U., & Rappoport, A. 2008. Multi-task active learning for linguistic annotations. In *Proceedings of ACL-08: HLT*, pp. 861–869.

Rosenfeld, A., & Kraus, S. 2016. Strategical argumentative agent for human persuasion: A preliminary report. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Dagsthul, Germany*.

Rosenfeld, A., & Kraus, S. 2018. Predicting human decision-making: From prediction to action. *Synthesis lectures on artificial intelligence and machine learning*, *12*(1), 1–150.

Rotman, G., Vulić, I., & Reichart, R. 2018. Bridging languages through images with deep partial canonical correlation analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 910–921.

Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, *abs/1706.05098*.

Shaikh, O., Chen, J., Saad-Falcon, J., Chau, P., & Yang, D. 2020. Examining the ordering of rhetorical strategies in persuasive requests. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1299–1306, Online. Association for Computational Linguistics.

Shapiro, C., & Varian, H. R. 2008. *Information rules: a strategic guide to the network economy*. Harvard Business School Press.

Shaw, P., Uszkoreit, J., & Vaswani, A. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Sim, Y., Routledge, B., & Smith, N. A. 2015. A utility model of authors in the scientific community. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1510–1519.

Smith, N. A. 2010. Text-driven forecasting. Tech. rep., University of Washington.

Søgaard, A., & Goldberg, Y. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 231–235, Berlin, Germany. Association for Computational Linguistics.

Spence, A. M. 1973. Job market signaling. *Quarterly Journal of Economics*, *87*, 355–374.

Steyvers, M., & Griffiths, T. 2007. Probabilistic topic models. *Handbook of latent semantic analysis*, *427*(7), 424–440.

Sundermeyer, M., Schlüter, R., & Ney, H. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., & Lee, L. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, p. 613–624, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. 2017. Attention is all you need. *CoRR*, *abs/1706.03762*.

Vulkan, N. 2000. An economist's perspective on probability matching. *Journal of economic surveys*, *14*(1), 101–118.

Walton, D. 2009. Argumentation theory: A very short introduction. In *Argumentation in Artificial Intelligence*, pp. 1–22. Springer.

Wang, X., Shi, W., Kim, R., Oh, Y., Yang, S., Zhang, J., & Yu, Z. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5635–5649, Florence, Italy. Association for Computational Linguistics.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yang, D., Chen, J., Yang, Z., Jurafsky, D., & Hovy, E. 2019. Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3620–3630.

Yogatama, D., Heilman, M., O'Connor, B., Dyer, C., Routledge, B. R., & Smith, N. A. 2011. Predicting a scientific community's response to an article. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ziser, Y., & Reichart, R. 2018. Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 238–249.