Journal on Multimodal User Interfaces
https://doi.org/10.1007/s12193-021-00368-w

ORIGINAL PAPER

An expert-model and machine learning hybrid approach to predicting human-agent negotiation outcomes in varied data

Johnathan Mell[1] · Markus Beissinger[2] · Jonathan Gratch[3]

Received: 1 November 2019 / Accepted: 3 February 2021

Abstract
We present the results of a machine-learning approach to the analysis of several human-agent negotiation studies. By combining expert knowledge of negotiating behavior compiled over a series of empirical studies with neural networks, we show that a hybrid approach to parameter selection yields promise for designing more effective and socially intelligent agents. Specifically, we show that a deep feedforward neural network using a theory-driven three-parameter model can be effective in predicting negotiation outcomes. Furthermore, it outperforms other expert-designed models that use more parameters, as well as those using other techniques (such as linear regression models or boosted decision trees). In a follow-up study, we show that the most successful models change as the dataset size increases and the prediction targets change, and show that boosted decision trees may not be suitable for the negotiation domain. We anticipate these results will have impact for those seeking to combine extensive domain knowledge with more automated approaches in human-computer negotiation. Further, we show that this approach can be a stepping stone from purely exploratory research to targeted human-behavioral experimentation. Through our approach, areas of social artificial intelligence that have historically benefited from expert knowledge and traditional AI approaches can be combined with more recent proven-effective machine learning algorithms.

Keyword Human-computer interaction · Human-subjects research · Machine learning · Neural networks · Expert models of human behaviour · IAGO negotiation platform

1 Background and motivation

1.1 Social AI and machine learning

Increasingly, artificial intelligence is being used to solve ever-more complex problems—such as those that involve understanding and reacting to human behavior. Interest has grown in artificial agents that can serve as representatives in negotiation [24], personal assistants [26], teachers, [18] and analysts of consumer behavior [28] as well as political affiliation [27, 31]. These new domains of interest for computational agents present exciting new problems, especially given the breakneck pace of progress in machine learning (ML) approaches in a variety of nearby problems (e.g., natural language understanding/generation).

But, there are stumbling blocks. While studies and datasets that examine ecologically valid human behavior are numerous, they are also often 1) small 2) messy 3) feature-rich and 4) (often) proprietary. This presents a problem for existing machine learning approaches which aim to predict social outcomes, as existing feature selection approaches that rely on purely automated techniques may have poor accuracy or suffer from overfitting. What may have once begun as straightforward classification problems in machine

✉ Johnathan Mell
Johnathan.Mell@ucf.edu

Markus Beissinger
mabeis@microsoft.com

Jonathan Gratch
gratch@ict.usc.edu

[1] University of Central Florida, Orlando, FL, USA

[2] Microsoft, San Francisco, CA, USA

[3] USC Institute for Creative Technologies, Los Angeles, CA, USA

Published online: 03 March 2021

learning are often difficult in social datasets, which are often "wider" than they are "deep".[1]

However, we argue that these problems can be addressed by integrating classical AI techniques as well as domain knowledge from the fields that have heretofore been highly active in analyzing human behavior historically: psychology and behavioral economics. To wit, we describe a particular thorny domain that has been described as a "challenge problem" for socially-aware AI and user interfacing: human-agent negotiation [3]. We first present the background of this problem, then present the results of a blended ML and expert-modeling study that shows the efficacy of our method.

1.2 Human-agent negotiation: challenge problem

Negotiation is a complex human social task that requires a diverse set of skills: from strategic planning to rhetorical argument. And while structured negotiation is still a relatively uncommon activity, most people engage in some degree of negotiation quite often—from deciding what to eat for dinner with a group of friends, to negotiating a job offer, to making a customer service request. Regardless of the domain, negotiation plays a critical role in human interaction. Research shows that people who are highly skilled negotiators tend to receive better salaries [14], and negotiation training pedagogy has long been the domain of prestigious business prep courses [6].

While negotiation has been traditionally seen as a uniquely human problem, that perception is quickly changing. As current technological tools continue to evolve into ever-more sophisticated artificial agents, humans find themselves relying on increasingly human-aware agents to interact with the world around them. Designing artificial agents that are capable of engaging in human-like negotiation has become a particularly challenging problem for researchers across a number of subfields, since it involves a variety of computational challenges, from social awareness to user modeling to learning [3]. In some cases, these new agents are designed to be high-fidelity simulacra of humans themselves, enabled to act with all the foibles of their mortal antecedents; in others, these agents merely need to understand and anticipate human negotiating behavior. Negotiating agents can act as representatives to humans, automating online bidding [2], providing the moral core to autonomous vehicles [5], and even negotiating the time of appointments for their users [26].

Currently, many of these agents are designed based on existing psychological/behavioral models of user behavior in negotiation (for example, see [21]). These agents draw from an immense literature on effective negotiation strategies in the business, psychology, and behavioral economics corpora. As these neophyte automated agents continue to negotiate with humans, however, they generate a massive amount of behavioral data. As such, analysis of human-agent interaction through machine learning approaches is becoming increasingly feasible.

While there are many reasons to have socially-capable agents, one key challenge is to use them to predict the outcomes of negotiation—**can an agent predict the outcome of a negotiation as it is happening (and perhaps, one day, adapt to it)?** But even when the stated goal is merely to predict the outcomes of negotiation, machine learning approaches are not panaceas. Human-agent negotiating datasets have a tendency to be "wide and short", with hundreds of behavioral and process variables being tracked, but relatively few subjects (due to the difficulty in conducting massive user studies). These problems lead to very noisy inputs into traditional machine learning algorithms, and make feature selection a chancy proposition at best. However, the possibility of rapid recruitment of human subjects through online platforms (such as Amazon's Mechanical Turk) and the increase in platforms for online human-agent interaction in social tasks has slightly ameliorated these problems [1, 23].

Still, even with overfitting precautions, there is a danger of blind application of algorithms to social computing problems. Fully "black-box" solutions do not generally allow for explainable AIs, which are increasingly in demand where the intersection of humans and computers are concerned. Where explainability is lacking there is danger of running afoul of legislation [15], drawing entirely incorrect conclusions [8], or at the very least failing to distinguish between seemingly similar classifiers [20, 29].

We therefore propose a hybrid approach for the analysis and development of agents in human-agent negotiation. By inputting expert knowledge of the domain into machine learning algorithms, we effectively create "priors" that allow these algorithms to more accurately account for input noise without needing massive numbers of data points. From the side of model-driven AI, this also allows for us to quickly and effectively evaluate a variety of potentially relevant behavioral parameter sets, while also circumventing some of the limitations of traditional evaluation approaches. To wit, much work in human behavioral examination in psychology has been done via traditional statistical analysis techniques, including hand-designed significance testing [17]. These methods have both practical (e.g., time) and methodological (e.g., statistical power loss) limitations. Indeed, creation of linear regression tests for more than three interaction terms quickly becomes impractical [10]. Therefore, techniques derived from machine learning provide tangible benefit, as

[1] "Wide" datasets have numerous measurable variables, while "deep" ones have many entries. Social dataset often have lots of variables to track but relatively few participants.

they allow automated and rapid evaluation of hypothetical behavioral models, thereby allowing for the creation of data-driven, model-informed AIs.

Much of the study of negotiation in general, and human-agent negotiation specifically, has focused on the application of certain techniques that will be effective in creating or claiming value in negotiations. This can entail such techniques as strategically withholding or sharing private information about preferences [30], using positive or negative emotion to manipulate other parties [12], or accurately modeling opponent preferences and crafting offers which “grow the pie” by finding integrative potential [11].

The remainder of this work primarily examines data from a set of three human-agent negotiating experiments conducted on the Interactive Arbitration Guide Online (IAGO) negotiation platform [23]. We show that a theoretically-sound and minimal-parameter neural network outperforms other models that use more simplistic approaches (linear regression) or more parameters (including those that are strict supersets). We also show the suitability of neural networks in general over other learning algorithms (such as XGBoost).

Extending this work, we then apply similar analysis to a larger dataset (inclusive of the first). Here, we show that as the number of data points increases, different classes of learning algorithm surface as most effective. We additionally perform new analysis predicting additional outcomes (specifically, the duration of the interaction) by examining the behavior measures that can be captured on-the-fly.

1.3 Models of human behavior

The goal of much of human-agent negotiation work is to predict outcomes using process measures found within the negotiation—observable parameters such as numbers or types of offers sent by each party. By predicting these outcomes in aggregate, we are able to design strategies that integrate this information on-the-fly, leading to more successful negotiation strategies, as well as insights about human behavior in negotiation scenarios. To this end, hundreds of variables may be tracked in an average human-agent negotiation (in this dataset, specifically, over 200). These variables include a variety of types:

- Process measures—number of messages sent by each party, emotional expressions detected, offer numbers and types, etc.
- Strategy variables—policies used by negotiating agents, such as whether they use emotional manipulation or attempt to withhold key information
- Demographic/Self-report variables—answers to survey questions by users, such as ratings of rapport or realism, and answers to psychometric surveys (e.g., Social Value Orientation or Machiavellianism) [25]

We focus on the first two types primarily in this work.

In a classical behavioral study, one or more strategy variables may be manipulated experimentally in order to see the resulting change on outcomes. There are a number of papers in this vein, which have discovered notable results that are key to the development of effectively social agents [2, 7, 12, 13]. Multiple regression is then performed to determine if there are any first-order or interaction effects on the dependent variables. However, this approach has limitations—traditional regression becomes untenable beyond a few independent inputs, as the statistical power quickly becomes weak. Furthermore, this method supports only linear combinations of variables, and any model which violates this assumption requires alternative approaches. Machine learning provides alternatives to this procedure through the construction of neural nets, but suffers from a sensitivity to noise in the data. Furthermore, in datasets that contain hundreds of potential input variables, a brute force approach to analysis (even with feature selection techniques) becomes absurd.

In our approach, we aim to predict outcome metrics of a negotiation based on a series of theory-driven agent models that contain measurable process parameters. For each agent parameters set, we wish to predict several outcome metrics of the negotiation as indicated by the target columns below.

- Scalar targets (unbounded numbers): agent points, user points, Nash points, and total points
- Percentage targets (bound to the range [0,1]): agent point %, user point %, total point %
- Binary target: isPerfect: a true/false value for determining if the negotiation solution is Pareto Optimal; i.e., no side of the negotiation could unilaterally do better without hurting the other side

These outcomes represent traditionally important results from a negotiation—by examining individual points such as agent points and user points, it is possible to objectively measure how well one side of a negotiation performed. However, by also tracking joint measures such as Nash points (product of user and agent points) and total points (sum of user and agent points), a sense of how much the negotiation resulted in “value creation” can be measured. This is an equally important in integrative scenarios, where the measurement of interest may be a sense of “cooperation” between participants, since negotiation is almost always a mixed-motive game. The percentage targets represent the same points as portions of the total amounts of points available (which helps make the data more domain-agnostic). In our first set of analyses (which cover the “Initial” Dataset), we focus on both sets of targets.

Table 1 Variable descriptions

Variable Name	Description
nice	A binary agent variable that describes the emotions the agents used
withholding	A binary agent variable that describes how the agents revealed information
competitive	A binary agent variable that describes if the agent reluctantly gave ground or built consensus
numUserOffers	The number of offers the human sent
numAgentOffers	The number of offers the agent sent
numUserMsg	The number of messages the human sent
numAgentMsg	The number of messages the agent sent
numUserHappy	The number of happy emojis the user sent
numAgentHappy	The number of happy emojis the agent sent
numUserAngry	The number of angry emojis the user sent
numAgentAngry	The number of angry emojis the agent sent
numUserEmote	The number of total emojis the user sent
numUserMsgOnly	The number of non-preference messages the user sent
numUserCombined	The number of preference messages the user sent
gameEndTime	The time, in seconds, when the negotiation concluded due to agreement

In the second set, we focus on the percentage targets only. More details on this focus is provided in Sect. 2.1.

While many models of human behavior rely on a number of topics and variables relevant to negotiation, including personality [25], in this work, we focus on a smaller subset. These variables generally deal with functions of the agents themselves (how they act) or with processes of the negotiation (how many times an act occurred). These variables are summarized and explained in Table 1.

1.4 Negotiation platform

In this work, we conduct examine negotiations conducted using the IAGO negotiation platform, as designed by Mell et al. [22]. IAGO negotiations are conducted online, using crowdsourcing technologies (such as Amazon's Mechanical Turk) to recruit subjects. The IAGO platform features an embodied virtual agent, as well as a text-based interface. The agent is capable of displaying emotion through the use of a visual avatar, and communicating using a chat-style interface. A virtual "negotiation table" is included in the platform, which allows the customizable creation of various types of multi-item negotiation scenarios. Users can move items on the board, display their own emotions through emoticons, and communicate with the agent. The agent can also take on a variety of different physical traits, by loading one of 4 predefined characters (varying in gender and photo-reality).

2 Experimental design

2.1 Datasets

The data used in this review is divided into two sets: the initial set, and the full set (a superset of the former). The initial set comprises 485 subjects (163 male, 126 female, 196 did not report.) collected over a series of 3 different human subjects studies. These studies were all conducted on the IAGO platform, a system for facilitating human-agent negotiation data collection, as well as the design of agents that use expert strategies to conduct negotiation [22]. IAGO automatically collects over 200 individual data columns for each participant (of which a handful were curated as being relevant by our negotiation expert). These columns are shown in Table 1.

Parts of this dataset have been used to analyze the behavior of agents pursuing different strategies of emotional manipulation and strategic information holding [23]. Other parts were collected for unrelated studies. In all cases, the studies were subject to ethical review by the originating university's Institutional Review Board, and the anonymous data was subsequently shared with us.

The full set includes both the initial set as well as an additional 769 subjects from two additional studies. These subsequent studies vary slightly in structure from the

Table 2 Models and parameters

Set name	Parameters
KnowThineEnemy	numUserOffers, numUserMsgOnly, numUserCombined, numUserHappy, numUserAngry
KnowAll	nice, withholding, competitive, numUserOffers, numUserMsgOnly, numUserCombined, numUserHappy, numUserAngry
Self-Reflection	nice, withholding, competitive, numAgentOffers, numAgentHappy, numAgentAngry, numAgentMsg
Emotional	nice, numUserHappy, numUserAngry, numAgentHappy, numAgentAngry
Strategic	competitive, numUserOffers, numAgentOffers
Chatty	withholding, numUserMsg, numAgentMsg
EverythingMakesSense	nice, withholding, competitive, numUserOffers, numAgentOffers, numUserMsg, numAgentMsg, numUserHappy, numAgentHappy, numUserAngry, numAgentAngry

previous three, and allow us to examine the efficacy of our initial models on this more diverse (and larger data set). As will be described, we find that the most successful models change as the size of the dataset grows, which leads to different recommendations based on the dataset size.

In all studies, the participants engaged in a multi-issue bargaining task with an IAGO agent, and attempted to maximize their own points. This standard task involved splitting up a number of items between the participants, while the value of the items was unknown to the opposing party. Participants were able to interact with the IAGO agent using pre-formed text responses in a dialogue tree. They were also able to move items on a virtual "table" to indicate their various offers, and were able to express their emotions using emoticons (the agent was an embodied head-and-shoulders character, who emoted back with prototypical facial expressions).

Participants were incentivized with real-world monetary lottery tickets to a cash pool based on their score at the end of the interaction. Several different types of agents were used; their behavior varied according to a number of dimensions including use of emotion, use of competitive bargaining techniques, and willingness to discuss strategic information (such as preferences and utilities). All studies in the initial set (coincidentally) involved a negotiation over 20 individual items, however the relative values of these items varied between the agent and the human, as well as between studies. The full set did differ in the amount of items and their point values—as such, analysis of the full set included only the percentage targets (which are meaningful given they are normalized measures), and eschews analysis of the scalar targets (which are not normalized). Nevertheless, in all studies, it was possible to "grow the pie" by finding issues across which there was integrative potential. Note that the structure of the task (the point values of the issues, e.g.) is not a feature of the learning, since this information is not fully observable by an agent. Future work may include examination of partial structural parameters by differentiating between those terms which are and are not observable (for instance, the agent could legitimately view its own point values), but we do not consider that distinction in this work.

The studies were all broadly similar, featuring agents that had the same visual fidelity (but occasionally did have different genders). There were some specific differences. Specifically, the first (initial set) of data included the results of three experiments. These experiments all contained an agent that were visually identical. The agent's behavior varied according to its emotional behavior, competitive bargaining strategy, and its willingness to withhold information. As previously stated, there was an integrative solution wherein both parties could do well.

The full set of data included the results of two additional experiments. The first experiment varied similar metrics, but also changed the agent's perception of the problems (it was initialized with priors that led it to believe there was no integrative potential, in some cases). The second experiment contained data from the Automated Negotiating Agents Competition (ANAC), and therefore contained several researcher-created agents, all competing on an identical negotiation game.

2.2 Parameter set creation

Based on existing theories of negotiation, we designed 7 parameter sets that served as basic models in order to predict negotiation outputs. The parameters in each set are listed in Table 2. Our first three parameters sets explored different combinations of information about the player and the agent. *KnowThineEnemy* focused on user variables that agents could track about the human, while being completely agnostic about the agent's own behavior. *KnowAll* included all the information from *KnowThineEnemy*, but also included parameters that defined how the agent acted—in short, the agent was aware of its own trends in its behavior. This included information about how the agent acted in its use of emotion (**nice**), information revelation strategy (**withholding**), and general offer structure (**competitive**). Thirdly, *Self-Reflection* included only information on the agent itself, including all the aforementioned strategy measures as well as a number of the same process variables, but for the agent's side.

The second set of three parameter sets included models that focused on one particular channel of communication. *Emotional*, for example, looks at variables relating to human and agent affective choices, like the use of anger and happiness. It also included **nice**, the parameter governing this behavior within the agent. *Chatty* focuses on the idea that the messages exchanged in negotiation may be predictive due to their effect on rapport between the human user, and thus includes both message quantity variables as well as the agent strategy variable **withholding**. Finally, *Strategic* focuses on examining the strategy of the agent in making offers (**competitive**), and the quantities (but not types) of offers exchanged by both parties. The final model, *EverythingMakesSense* simply included all the variables—this is the most likely model to be attempted by someone with little awareness of negotiation theory. While each of the parameters it contains do have some basis (i.e., they all do relate in some way to negotiation), simply adding them all at once is brute force approach. Therefore, it serves as a reasonable baseline for "traditional" machine learning approaches—improved in the sense that it does involve expert-generated variables, but not crafted with a particular psychological model in mind.

2.3 Machine learning methodology

We compared the parameter sets above by using three machine learning methods on the cleaned subset of data gathered from experiments. The data from our user experiments was cleaned to include the union of rows where each model's input columns and targeted prediction columns did not have any missing values. As an example of this, the initial experiment included some entries that did not track the number of times the user expressed happiness. Later experiments did, but we excised rows missing this value variable as it was not present in the union of the two sets. In this way, we followed standard procedure for preparing a null-free set—eliminating all data points where the targeted prediction or input columns contained a missing value.

This resulted in 289 data points to train the machine learning methods in the initial set, and 654 data points in the second set, for a total of 943 cleaned interactions in the total, combined set.[2] For each of the parameter sets, we trained a linear regression baseline, an XGBoost [9], and a Deep Neural Network (DNN) to compare predictive performance of the specified input variables to the target scalar, percentage, and binary columns. This resulted in 7 (parameter sets) $\times$ 3 (ML algorithm) = 21 learning models.

We utilized k-folds cross-validation to compare machine learning methods across models, where $k = 10$ (based on [19]). This means the dataset was randomly split into 10 subsections, where 9 subsections are combined as the training set and 1 is left as the validation set for calculating mean square error (MSE)/root mean square error (RMSE) and filling in the predicted values. This training process is repeated 10 times such that the machine learning method is able to make test predictions for every row in our dataset. This procedure was replicated similarly for the full set.

Our linear regression baseline was implemented using scikit-learn,[3] and a separate linear regression was fit for each target column using all of the input columns specified by the model.

XGBoost was chosen as another comparison due to its successes in Kaggle competitions [16]. XGBoost functions by sequentially fitting weak learners (such as decision trees), then using gradient descent to learn the split parameters for the tree nodes. We created separate XGBoost regressions for each scalar target column, and separate XGBoost classifiers for each percentage target column to keep the output in the range [0,1]. The XGBoost classifier objective function is binary-logistic and can be interpreted as a probability output. This probability output is being trained against a [0,1] target, which can also be interpreted as a probability. To parameterize both XGBoost regression and classification, we used maximum depth = 7, learning rate = 0.08, number of estimators = 100, and subsample = 0.9. The subsampling parameter was used to help prevent overfitting.

Lastly, we used a DNN to compare model performance. Deep learning has gained popularity for its ability to create higher-level representations of input features [4]. Our DNN consisted of Feedforward layers interleaved with Dropout noise to reduce overfitting. We used Keras[4] for hyperparameter experimentation to find number of layers, layer sizes, activations, and learning rate. The DNN we used was parameterized with the following layers:

- Feedforward (256 units, SELU activation, Glorot uniform initialization) Alpha Dropout (0.4 noise)
- Feedforward (256 units, SELU activation, Glorot uniform initialization) Alpha Dropout (0.4 noise)
- Feedforward (128 units, SELU activation, Glorot uniform initialization) Alpha Dropout (0.4 noise)
- Feedforward (128 units, SELU activation, Glorot uniform initialization) Alpha Dropout (0.4 noise)

[2] Since not all experiments contained the same inputs, statistical differences in the retention rate of data after cleaning (289/485 = 60% vs. 654/769 = 85%) are to be expected.

[3] https://scikit-learn.org/.

[4] https://keras.io/.

Table 3 Average root mean squared error for all models, negotiation outcomes (initial dataset)

Model Name	User Points RMSE	Agent Points RMSE	Nash Points RMSE	Total Points RMSE	User % RMSE	VH % RMSE	Total % RMSE	isPerfect RMSE	# of Categories "won" in same p-set.
ChattyDNN	6.31	6.54	194.92	8.26	0.128	0.111	0.128	0.446	7
SelfReflectionDNN	6.85	7.09	220.23	9.94	0.125	0.115	0.145	0.402	4
StrategicDNN	6.90	7.01	221.55	9.79	0.125	0.115	0.145	0.398	4
EverthingMakesSenseDNN	6.76	8.05	221.89	10.80	0.125	0.111	0.138	0.430	5
ChattyLinear	6.87	5.70	222.58	9.17	0.137	0.114	0.141	0.460	1
StrategicLinear	6.73	5.51	226.20	9.52	0.135	0.110	0.146	0.455	4
KnowThineEnemyLinear	6.93	6.42	227.27	9.81	0.139	0.128	0.151	0.455	4
SelfReflectionLinear	6.78	5.60	228.51	9.67	0.136	0.112	0.149	0.459	4
KnowAllLinear	6.87	5.89	230.21	9.84	0.137	0.118	0.151	0.458	5
EverythingMakesSenseLinear	7.04	5.33	235.39	9.76	0.141	0.107	0.150	0.460	3
EmotionalLinear	7.36	6.29	237.50	9.97	0.147	0.126	0.153	0.490	6
KnowAllDNN	7.53	8.24	239.77	12.00	0.136	0.123	0.148	0.453	3
KnowThineEnemyDNN	7.33	8.12	240.66	11.89	0.136	0.125	0.150	0.451	4
EmotionalDNN	7.75	7.22	241.96	11.32	0.141	0.126	0.149	0.487	3
EverythingMakesSenseXGBoost	7.08	6.03	244.36	10.31	0.157	0.117	0.173	0.496	0
ChattyXGBoost	7.91	6.29	256.02	10.86	0.168	0.137	0.161	0.552	0
KnowThineEnemyXGBoost	7.66	6.68	260.74	10.62	0.160	0.142	0.155	0.579	0
SelfReflectionXGBoost	7.98	6.93	262.37	12.01	0.175	0.141	0.184	0.492	0
StrategicXGBoost	7.50	6.87	263.01	11.98	0.153	0.131	0.169	0.492	0
KnowAllXGBoost	7.78	6.64	263.62	10.82	0.167	0.140	0.172	0.576	0
EmotionalXGBoost	8.50	6.96	275.39	11.65	0.183	0.145	0.158	0.617	0

Blue items are the lowest (best) values in the column, while yellow items are the second lowest in the column. The final column compares each model type to the others of the same parameter set: e.g., "ChattyLinear"–"ChattyDNN"–"ChattyXGBoost" and displays the amount of categories in which it was the top performer

- Feedforward (64 units, SELU activation, Glorot uniform initialization) Alpha Dropout (0.4 noise)
- Feedforward (64 units, SELU activation, Glorot uniform initialization) Alpha Dropout (0.4 noise)

For predicting the scalar target columns, we used the DNN specified above with an added Feedforward (4 units, Linear activation, Glorot uniform initialization) layer for the final outputs and used MSE loss for training. For our percentage and binary targets, we added a Feedforward (4 units, Sigmoid activation,[5] Glorot uniform initialization) layer to keep each of the final outputs in the [0,1] range, like we did with XGBoost, and used binary cross-entropy loss for training. In both cases, we trained the DNN with stochastic gradient descent using the Adam optimizer with learning rate = 0.0001, beta1 = 0.9, and beta2 = 0.999, with a batch size of 64 examples. We trained the networks for 100,000 epochs with early stopping if the validation set loss did not improve over a patience value of 1,000 epochs.

All hyperparameters for XGBoost and DNN were chosen from a manual search of small value ranges around each variable. The best performing model on a held-out evaluation set was chosen for our subsequent k-folds training and results analysis.

3 Results: initial set

3.1 Absolute performance

For each of the 7 theoretical parameter sets, there are three variants: linear regression (Linear), XGBoost (Boost), and deep neural network (DNN). The resulting 21 models are listed in Table 3, along with the root mean squared errors (RMSE) for each of the 8 target outputs in the initial set.

[5] A sigmoid serves our purposes to force 0–1 range to form a percentage. A different approach, such as softmax, would force a percentage across all outputs so they must sum to 1. We are looking to get a percentage for each output individually, so we chose the sigmoid.

RMSE was calculated from the average across each of the 10 folds in k-folds cross validation. The table also highlights the best performing model for each output in blue, and the second best in yellow (ties are allowed). We particularly note the top performing model: *ChattyDNN*, a neural network model that is the best performer in 4 out of 7 categories (and second best in 1). However, in general, the linear models performed better when compared to the same parameter set (e.g., *EmotionalLinear* vs. *EmotionalXGBoost* vs. *EmotionalDNN*). Indeed, when compared triple-wise in this manner, XGBoost models were *never* the top performers (see final column in Table 3).

Generally speaking, the models explained a reasonable amount of the variance in the results. The RMSE of *ChattyDNN* for **agent points** was 6.31. Since the negotiations involved between 65 and 70 total points, this prediction implies a 95% confidence interval within 12.62 points. Given the relative simplicity of the measures involved (*ChattyDNN* did not analyze the content of messages at all—only their quantity), this result is encouraging.

3.2 Comparative performance

ChattyDNN outperforms its baseline linear regression counterparts in every category (except **agent points**, where *ChattyDNN* is worse than *ChattyLinear*). *ChattyDNN* also performs adequately in one of the categories in which it is not in the top two—specifically *ChattyDNN* is the top 3 models for **agent point %**. Indeed, *ChattyDNN* performs quite well even accounting for its relative shortcomings in predicting agents' points.

In general, performing traditional significance testing when comparing models based solely on their RMSE is statistically difficult. However, by examining each fold of the models as a separate data point (and extracting the RMSE from those folds), we can craft a rudimentary univariate ANOVA on each model's total scalar target performance.

To accomplish this, we first report the estimated "mean of errors" for each of the 10 folds, for each of the 21 models, and report this in **Fig. 1**. Each model is shown with their performance on all scalar targets z-scored, and then averaged (**user points**, **agent points**, **total points**, and **Nash points**). This results in a normalized combination of each of the 4 scalar targets. *ChattyDNN* is the outlier in terms of best performance. The process is thus as follows:

1. Determine the error for each of the 10 folds, on each model, for each target
2. Z-score those 10 data points for each target.
3. Average the 4 targets together.

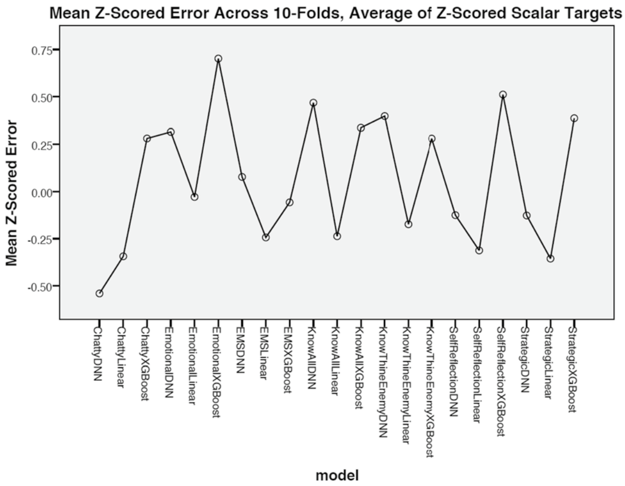

Fig. 1 10-Fold means of error for all models, average of all Z-Scored scalar targets (lower is better)

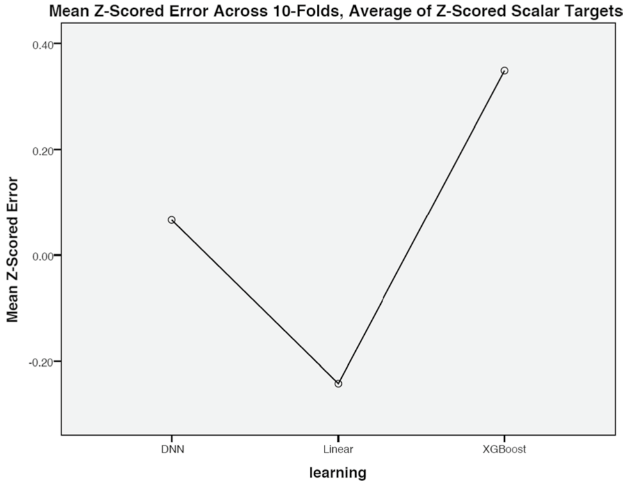

Fig. 2 10-Fold means of error for model classes, average of all Z-Scored scalar targets (lower is better)

In addition to comparing each model individually, we conducted a 3 (learning type) by 7 (parameter set) ANOVA analysis to determine if any particular parameter set or learning algorithm dominated. Indeed this appears to be the case (as seen in Fig. 2). There is a main effect of learning type on scalar target accuracy ($p < 0.001$). This effect is driven by the XGBoost models, which performed poorly compared to the other types, and the linear models which performed better, in general, than any other type.

3.3 Additional analysis

When analyzing a new data set, there is a tendency to include all relevant features (even when accounting for

Table 4 Root mean squared error for all models, comparable negotiation outcomes (full dataset)

Model Name	User % RMSE	Agent % RMSE	Total % RMSE	isPerfect RMSE	Game Length RMSE
KnowAllLinear	0.103	0.109	0.091	0.429	105.59
SelfReflectionLinear	0.103	0.108	0.094	0.439	94.49
KnowThineEnemyLinear	0.103	0.109	0.091	0.439	105.53
EverythingMakesSenseLinear	0.104	0.108	0.092	0.439	93.49
StrategicLinear	0.105	0.109	0.093	0.447	121.12
ChattyLinear	0.104	0.109	0.092	0.453	96.77
EmotionalLinear	0.105	0.108	0.094	0.461	122.65
SelfReflectionDNN	0.217	0.189	0.216	0.520	8518.39
KnowAllDNN	0.183	0.184	0.114	0.521	7976.32
EverthingMakesSenseDNN	0.204	0.194	0.257	0.525	8097.96
ChattyDNN	0.132	0.136	0.145	0.526	8686.70
KnowAllXGBoost	0.126	0.137	0.105	0.535	110.49
EverythingMakesSenseXGBoost	0.120	0.122	0.108	0.544	87.40
KnowThineEnemyDNN	0.164	0.161	0.116	0.553	9131.89
KnowThineEnemyXGBoost	0.122	0.130	0.109	0.555	111.86
StrategicXGBoost	0.114	0.124	0.110	0.558	119.20
SelfReflectionXGBoost	0.128	0.125	0.119	0.560	88.62
EmotionalDNN	0.151	0.174	0.116	0.573	6446.58
EmotionalXGBoost	0.123	0.124	0.111	0.579	128.59
ChattyXGBoost	0.125	0.144	0.116	0.596	101.68
StrategicDNN	0.143	0.193	0.131	0.607	7763.38

Blue items are the lowest (best) values in the column, while yellow items are the second lowest in the column

overfitting). One danger in including *all* features is that without theory-driven caution, the process can lead to misleading results. To showcase this problem, we included a final, eighth parameter set called *EMSTime*, which included all the parameters from the standard *EMS*, but additionally included **gameEndTime** as an input variable. For these models' performance, **gameEndTime** proved to be a very valuable parameter—*EMSTimeBoost* outperformed *ChattyDNN* on a number of target outputs. From a theoretical perspective, this is not particularly surprising—**gameEndTime** is a post-hoc measure that likely indicates how difficult it was to reach agreement in the negotiation. Being a post-hoc parameter, **gameEndTime** is furthermore unsuitable for online learning methods. Removing this single metric (thus resulting in the standard *EMS* models) leads to substantial performance degradation—*ChattyDNN* tends to dominate all the *EMS* variants on the majority of output targets. We revisit this concept again in the full dataset discussion below, but instead of using **gameEndTime** as an input, we reinterpret it as a prediction target.

4 Results: full set

4.1 Performance changes

We again examined the 21 models, but for the full dataset of 943 interactions. These results are now listed in Table 4, which includes only the RMSE of the percentage targets (since the scalar targets are less meaningful across differing studies due to their lack of normalization). Again, this table also highlights the best performing model for each output in blue, and the second best in yellow (ties are allowed).

In terms of absolute performance, we see comparable performance for the targeted measures in *ChattyDNN*, although the error rate for isPerfect (detecting whether or not the negotiation will end with a Pareto Optimal solution) has substantially increased. Indeed, the most improvement is seen among the simpler, linear models, with *KnowAllLinear, SelfReflectionLinear, and KnowThineEnemyLinear* becoming the top performing models in the larger dataset. On average, KnowAllLinear has improved by 19.1%, with its average mean squared error on percentage targets decreasing from 0.066 to 0.054.

There are a few possible takeaways from this change. The simplest explanation is that the linear models perform adequately on smaller datasets, but improve even quicker to their DNN counterparts. The full dataset more than doubles the size of the initial dataset, and it could be that this increase in data points allows the linear models to overtake the DNNs. This interpretation is supported by the fact that the DNNs do not degrade (much) in performance as more data points are added. It is somewhat unexpected, however, as DNNs traditionally do improve in performance when given more data. It is likely that the DNNs have already "peaked" earlier, and the linear models continue to benefit from additional data for longer. Given the same cross-fold validation procedure is followed for both the full and initial datasets, this is encouraging, as it means that choosing to use DNN models early on does not come at much penalty as new data is added—even if that data has moderately different characteristics from the initial.

The second primary interpretation is more speculative, but deals with the fact that the top-performing models in the full dataset all include more parameters than the previous top-performing models. Notably, *KnowAllLinear, KnowThineEnemyLinear,* and *EverythingMakesSenseLinear* all perform better than *ChattyLinear* in this expanded dataset. This is perhaps unsurprising—all of the priors included in these models were added because there is some theoretical basis behind them. With a larger dataset, it is reasonable to assume that weaker but still present correlative effects may be teased out of these more complex models, allowing them to pull ahead of the relatively simple *ChattyLinear*. In short, in smaller datasets, simple models like *ChattyLinear* may be superior since they are likely to find the strongest effects and avoiding overfitting on minor parameters. However, complex models, when fed enough data, may capture weak but present correlative effects, leading to their ascendency.

Regardless, it is important to note that *KnowAllLinear* still did outperform *EverythingMakesSenseLinear* (the superset model), which indicates that the previous observation (more parameters is *not* always better) still holds.

4.2 Analyzing negotiation length

One additional target value is analyzed in the full dataset that was not previously analyzed in the partial dataset. Due to evidence that suggests that the amount of time spent working out a deal may have a correlation to the quality of the deal achieved, we aimed to predict the total length of the negotiation interaction (i.e., the time until agreement was reached) using the aforementioned models.

The results indicate that there was varied success in predicting this time. However, the best models for predicting game length showed little overlap with the best models for the other prediction targets. Indeed, contrary to Fig. 2, the best two models for predicting game length were XGBoost models (*EverythingMakesSenseBoost* and *SelfReflectionBoost*). The RMS accuracy even with these two was not highly accurate, but this result is informative; the best model type for predicting temporal aspects of the interaction may well differ from the type best used for predicting the quality of the interaction. Future work that examines potential mediation or moderation effects of time on outcome would be welcome to disentangle this relationship.

5 Discussion

5.1 Initial set

The implications of the results in this work for those primarily interested in negotiation behavioral results indicate the importance of rapport (or at least, message exchange) to negotiated outcomes. Yet, more broadly for system design, they also have methodological implications for the design of human-agent systems and studies. Social computing problems such as human–computer negotiation are plagued by uncertain inputs, massive numbers of input variables, and relatively small datasets. These problems make them tenuous targets for much of the current work in machine learning. Furthermore, the domains in which social problems are most relevant are these where explainable AI and model-driven approaches are most valued (for ethical/legal/commercialization reasons). Therefore, approaches which can both leverage current machine learning approaches to process data, but can also "initialize their priors" using expert knowledge are of particular interest. Certainly, even if purely automatic learning solutions were tenable from a societal perspective, the performance of automatic feature selection is not always perfect. Indeed, as pointed out by Lucas et al. [21], domain knowledge can inform multimodal fusion by assisting with feature selection, and often models with fewer features can outperform those with more features.

The results of this work are largely in line with these intuitions. We did achieve generally good performance, with most models predicting user score within half a dozen points or so (12–18% of the total). This shows us that the approach is a reasonable predictor, and supplies the aforementioned benefits with regards to explainability and theory.

We can examine the best-performing parameter set, *Chatty*, in two ways. This model uses only three parameters (**withholding, numUserMsg,** and **numAgentMsg**). *Chatty* relies on the observation that communication is key to building rapport in negotiation, and this increased rapport and understanding of opponent preferences can lead to an increase in joint value.

In the traditional approach, we can analyze this variables as a regression-based problem. We can conduct follow-up analysis on a manually-constructed regression model, which shows that there are indeed significant main and interaction effects for this model on both **total points** and **agent points**. While a simple F-test indicates that the model performs better than a model that assumes the mean (not a particularly high bar), the presence of interaction and main effects hint at the idea that the *Chatty* parameter set has hit on some real behavioral results.

But, we observe from the results that the absolute best model is *ChattyDNN*, which outperforms its linear cousins in our initial results. *ChattyDNN*, which is based on theory-driven intuition about communication, is a neural network, *not* a standard regression model. As such, it represents a *hybrid* approach to hypothesis testing and modeling. And we can see that this second approach provides better performance than our initial approach, given *ChattyDNN*'s good performance at predicting **Nash points**, **total points**, and **total point** %.

Still, *ChattyDNN* is not particularly successful at predicting the points that the agent receives, being outperformed by a number of different models in that single category, but performs well elsewhere. Most notably, *ChattyDNN* outperforms its strict superset models in most categories. *EverthingMakesSenseDNN*, performs worse than *ChattyDNN* over most of its outputs. Given that *ChattyDNN* is an expert-inspired model, rather than one that was directly learned, this validates our hybrid approach. In an (increasingly common) example of "more parameters is not better", it outperforms a number of more complicated models, while having a theoretical groundwork based on the idea of the importance of information exchange.

However, *ChattyDNN* does not merely outperform more parameter-heavy models. It also outperforms its different-algorithm counterparts, *ChattyBoost* and *ChattyLinear*. One potential reason the DNN outperforms both XGBoost and linear regression for the *Chatty* parameter set is that its internal parameters are trained jointly across all target outputs of the same type—scalar or percentage and binary—while the other models have to train a single model for every target output. This allows for complex nonlinear dependencies within parameters to be affected by the joint training across potentially related target outputs. In other words, the DNNs have two different output layers: one for predicting the absolute targets, and one for the percentage and the binary output. However, the models use the same hidden layers. The other models, on the other hand, are trained individually for each output, so one individual "model" is trained based on the same input but for the 8 different output targets. DNNs can therefore profit from the correlations between the outputs, whereas the other models cannot. Still, mostly, the best models are Linear, not DNN. So, *ChattyDNN's* abnormal success remains notable (and somewhat difficult to pin down).

Nevertheless, the implications of *ChattyDNN*'s success are twofold. First, it is notable that reasonable prediction of outcomes is possible with an expert-informed parameter set (e.g., the *Chatty* parameter sets in general) which is then fed into an ML algorithm. What we have indicated with this result is that indeed, automation of lessons from traditional psychometric significance testing can lead to good results. This lends support to our approach in general. Secondly, we provide evidence of a DNN providing better results than XGBoost or a learned linear regression. This speaks to the potential benefit of nonlinear models learning complex relationships between the raw inputs, and (perhaps more strongly) about the improperness of using XGBoost, which overall performed poorly.

5.2 Full set

These results are complicated by the addition of new data to the analysis. The full dataset, which includes two new studies with different structures to the original allows us to test the robustness of these models by increasing the dataset size by adding new data that may not share the same context.

We are able to conclude that the baseline linear models perform quite admirably in this new context. Whether this is due to some deficiency in the DNNs when new data is added (e.g., due to overfitting) or due to the ability of linear models to achieve superior performance gains to DNNs when fed enough data is, as of yet, undetermined. However, we do believe these results clearly delineate the benefits and drawbacks of these different methods depending on the targeted prediction variables as well as the data size.

To formalize this, we compared the effectiveness of two very different machine learning algorithms across all the models in our social negotiation data. From the results, it is not yet clear the "right" DNNs will always reliably outperform linear regression on the whole, although the best model in the initial set, *ChattyDNN,* did. Rather, the results seems to indicate that the "safe" bet is often a linear model, which generally performed well. Still, while linear models may be the overall good choice in this problem, the best performing model is often determined by the particular parameter set and the prediction targets. And, as analysis of the full set reveals, these lessons evolve with the size of the data set as well.

However, we did show the superior predictive model for each variable was almost always a linear or DNN model in both the initial and full sets. We believe this speaks to the questionable efficacy of the XGBoost models in this application. In particular, every XGBoost model performed substantially worse than the best model for every variable in both sets, with the exception of the game length prediction in the full set. Moreover, the top performing models, as judged by the amount of variables predicted with the least error, were all linear models in the full dataset. The comparison between XGBoost and DNN models in the full set is less clear, since some outperform each other on particular variable.

What remains clear, however, is that in this result, a hybrid approach to analysis can yield reasonable prediction accuracy, especially in domains that have datasets with features similar to human-agent negotiation. Naïvely-implemented approaches may simply contain too many parameters, such as the "kitchen sink" approach of the *EverythingMakesSense* parameter set. In our analysis, these EMS models were not the top-performing ones (although EMSLinear did show improvement in the full set). Our hybridized approach benefits from prior knowledge of the domain, and ensures we get the most "value" for each parameter that is added.

Moreover, we have also demonstrated the power of learned models over purely hand-tuned ones. Manual regression testing of models with too many parameters is impractical at best (and impossible at worst). Machine learning-aided analysis isolates areas of particular interest, and allows post-hoc analysis or follow-up experimentation that will further clarify the mechanisms of parameter sets that perform well. This also dovetails nicely with the idea of explainability, since these models have theoretical basis.

Of course, such approaches have limits. In cases when parameter-heavy neural nets like *EverythingMakesSenseDNN do* outperform those with fewer parameters like *ChattyLinear*, there is not an easy theoretical explanation as to why. And, even when a model performs well, the direct mechanism by which it does may still be hidden. Our approach does not claim to be panacæa for explainability—but hybrid approaches such as ours *do* provide a starting point to further experiments. In this particular case, it allows us to target particular parameters that are contained in high-performing models that are omitted in lower-performing models and design specific experiments around them. As such, hybrid approaches may be highly suitable for early/mid-stage research, wherein several competing mechanisms are proposed, evaluated, and then pursued. Compare this approach to one in which no expert-tuned parameters were provided; in which case follow-up experiments are vastly less directable.

Therefore, given that this hybridized approach actively directs further research and analysis to areas where mechanisms may exist, its importance should not be underestimated. In future work, the hybrid approach can be used both as a mechanism for the analysis of existing human-agent datasets, as well as a tool for the design of agents that use online, hybrid expert-learned systems for performing negotiation.

Acknowledgements The authors want to thank our colleague Gale Lucas for the insights on the analytical approach to these results. This research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-20-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Alto KM, McCullough KM, Levant RF (2018) Who is on craigslist? a novel approach to participant recruitment for masculinities scholarship. Psychol Men Masc 19(2):319
2. Anthony P, Jennings NR (2003) Developing a bidding agent for multiple heterogeneous auctions. ACM Trans Internet Technol (TOIT) 3(3):185–217
3. Baarslag T, Kaisers M, Gerding E, Jonker CM, Gratch J (2017) When will negotiation agents be able to represent us? The challenges and opportunities for autonomous negotiators. Sierra, Carles (ed.) In Proceedings of the twenty-sixth international joint conference on artificial intelligence. International joint conferences on artificial intelligence. pp. 4684–4690.
4. Bengio Y (2013) Deep learning of representations: looking forward. In international conference on statistical language and speech processing (pp. 1–37). Springer, Berlin
5. Bonnefon JF, Shariff A, Rahwan I (2016) The social dilemma of autonomous vehicles. Science 352(6293):1573–1576
6. Bordone RC (2000) Teaching interpersonal skills for negotiation and for life. Negot J 16(4):377–385
7. Castelfranchi C, Falcone R, (1998) Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In Multi Agent Systems, 1998. Proceedings. International conference on (pp. 72–79). IEEE
8. Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N (2015) Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. In proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1721–1730). ACM
9. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785–794). ACM
10. Cohen J (2013) Statistical power analysis for the behavioral sciences. Academic press, Cambridge

11. De Dreu CK, Koole SL, Steinel W (2000) Unfixing the fixed pie: a motivated information-processing approach to integrative negotiation. J Pers Soc Psychol 79(6):975
12. de Melo CM, Carnevale P, Gratch J, (2011) The effect of expression of anger and happiness in computer agents on negotiations with humans. In the 10th international conference on autonomous agents and multiagent systems-vol. 3 (pp. 937–944). International foundation for autonomous agents and multiagent systems.
13. Fox J, Ahn SJ, Janssen JH, Yeykelis L, Segovia KY, Bailenson JN (2015) Avatars versus agents: a meta-analysis quantifying the effect of agency on social influence. Human Comput Interact 30(5):401–432
14. Gerhart B, Rynes S (1991) Determinants and consequences of salary negotiations by male and female MBA graduates. J Appl Psychol 76(2):256
15. Goodman B, Flaxman S (2017) European Union regulations on algorithmic decision-making and a “right to explanation.” AI Magazine 38(3):50–57
16. Gupta T (2018) How i got in the top 1% on Kaggle. towards data science, https://towardsdatascience.com/how-i-got-in-the-top-1-on-kaggle-79ddd7c07f1c
17. Hubbard R, Ryan PA (2000) The historical growth of statistical significance testing in psychology—And its future prospects. Educ Psychol Measur 60(5):661–681
18. Johnson E, Gratch J, DeVault D (2017) Towards an autonomous agent that provides automated feedback on students’ negotiation skills. In proceedings of the 16th conference on autonomous agents and multiagent systems (pp. 410–418). International foundation for autonomous agents and multiagent systems
19. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. In: Proceedings of the 31st international conference on neural information processing systems, pp 972–981
20. Lapuschkin S, Binder A, Montavon G, Muller KR Samek W (2016) Analyzing classifiers: fisher vectors and deep neural networks. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2912–2920)
21. Lucas G, Stratou G, Lieblich S, Gratch J (2016) Trust me: multimodal signals of trustworthiness. In Proceedings of the 18th ACM international conference on multimodal interaction (pp. 5–12). ACM
22. Mell J, Gratch J (2016) IAGO: interactive arbitration guide online. In AAMAS (pp. 1510–1512)
23. Mell J, Gratch J (2017) Grumpy and Pinocchio: answering human-agent negotiation questions through realistic agent design. In Proceedings of the 16th conference on autonomous agents and multiagent systems (pp. 401–409). International foundation for autonomous agents and multiagent systems
24. Mell J, Lucas GM, Gratch J (2018) Welcome to the real world: how agent strategy increases human willingness to deceive. In Proceedings of the 17th international conference on autonomous agents and multi agent systems (pp. 1250–1257). International foundation for autonomous agents and multi agent systems
25. Mell J, Lucas G, Mozgai S, Boberg J, Artstein R, Gratch J (2018) Towards a repeated negotiating agent that treats people individually: cooperation, social value orientation, & machiavellianism. In Proceedings of the 18th international conference on intelligent virtual agents (pp. 125–132). ACM
26. Metz Rachel (2018) Google demos Duplex, its AI that sounds exactly like a weird, nice human. Intelligent machines. Downloaded from https://www.technologyreview.com/s/611539/google-demos-duplex-its-ai-that-sounds-exactly-like-a-very-weird-nice-human/
27. Pennacchiotti M, Popescu AM (2011) A machine learning approach to twitter user classification. In Fifth international AAAI conference on weblogs and social media
28. Perlich C, Dalessandro B, Raeder T, Stitelman O, Provost F (2014) Machine learning for targeted display advertising: transfer learning in action. Mach Learn 95(1):103–127
29. Samek W, Wiegand T, Müller KR (2017) Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models. arXiv:1708.08296
30. Thompson LL (1991) Information exchange in negotiation. J Exp Soc Psychol 27(2):161–179
31. Yu B, Kaufmann S, Diermeier D (2008) Classifying party affiliation from political speech. J Inf Technol Politics 5(1):33–48