A Dynamic Strategy Coach for Effective Negotiation

Yiheng Zhou[♡] **He He**[◇] **Alan W Black**[♡] **Yulia Tsvetkov**[♡]

[♡]Language Technologies Institute, Carnegie Mellon University

[◇]Computer Science Department, Stanford University

{yihengz1, awb, ytsvetko}@cs.cmu.edu, hehe@cs.stanford.edu

Abstract

Negotiation is a complex activity involving strategic reasoning, persuasion, and psychology. An average person is often far from an expert in negotiation. Our goal is to assist humans to become better negotiators through a machine-in-the-loop approach that combines machine's advantage at data-driven decision-making and human's language generation ability. We consider a bargaining scenario where a seller and a buyer negotiate the price of an item for sale through a text-based dialog. Our negotiation coach monitors messages between them and recommends tactics in real time to the seller to get a better deal (e.g., "reject the proposal and propose a price", "talk about your personal experience with the product"). The best strategy and tactics largely depend on the context (e.g., the current price, the buyer's attitude). Therefore, we first identify a set of negotiation tactics, then learn to predict the best strategy and tactics in a given dialog context from a set of human–human bargaining dialogs. Evaluation on human–human dialogs shows that our coach increases the profits of the seller by almost 60%.[1]

1 Introduction

Negotiation is a social activity that requires both strategic reasoning and communication skills (Thompson, 2001; Thompson et al., 2010). Even humans require years of training to become a good negotiator. Past efforts on building automated negotiation agents (Traum et al., 2008; Cuayáhuitl et al., 2015; Keizer et al., 2017; Cao et al., 2018; Petukhova et al., 2017; Papangelis and Georgila, 2015) has primarily focused on the strategic aspect, where negotiation is formulated as a sequential decision-making process with a discrete action space, leaving aside the rhetorical aspect. Recently, there has been a growing interest in strategic goal-oriented dialog (He et al., 2017; Lewis et al., 2017; Yarats and Lewis, 2018; He et al., 2018) that aims to handle both reasoning and text generation. While the models are good at learning strategies from human–human dialog and self-play, there is still a huge gap between machine generated text and human utterances in terms of diversity and coherence (Li et al., 2016a,b).

In this paper, we introduce a machine-in-the-loop approach (cf. Clark et al., 2018) that combines the language skills of humans and the decision-making skills of machines in negotiation dialogs. Our **negotiation coach** assists users in real time to make good deals in a bargaining scenario between a buyer and a seller. We focus on helping the seller to achieve a better deal by providing suggestions on what to say and how to say it when responding to the buyer at each turn. As shown in Figure 1, during the (human–human) conversation, our coach analyzes the current dialog history, and makes both high-level strategic suggestions (e.g., ⟨*propose a price*⟩) and low-level rhetoric suggestions (e.g., ⟨*use hedge words*⟩). The seller then relies on these suggestions to formulate their response.

While there exists a huge body of literature on negotiation in behavioral economics (Pruitt, 1981; Bazerman et al., 2000; Fisher and Ury, 1981; Lax and Sebenius, 2006; Thompson et al., 2010), these studies typically provide case studies and generic principles such as "focus on mutual gain". Instead of using these abstract, static principles, we draw insights from prior negotiation literature and define actionable strategies and tactics conditioned on the negotiation scenario and the dialog context. We take a data-driven approach (§2) using human–human negotiation dialogs collected in a simulated online bargaining setting (He et al., 2018). First,

[1]The study was approved by the IRB. All sources and data are publicly released at https://github.com/zhouyiheng11/Negotiation-Coach.

Proceedings of the SIGDial 2019 Conference, pages 367–378

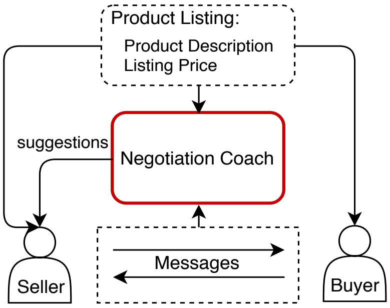

Figure 1: Our negotiation coach monitors the conversation between the seller and the buyer, and provides suggestions of negotiation tactics to the seller in each turn dynamically, depending on the negotiation scenario, the dialog context, and examples of previous similar dialogs.

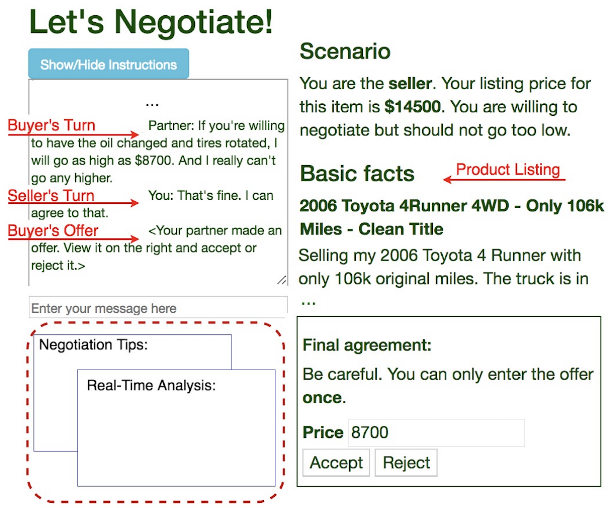

Figure 2: Negotiation interface with coaching.

we build detectors to extract negotiation tactics grounded in each turn, such as product embellishment ("*The TV works like a champ!*") and side offers ("*I can deliver it to you.*") (§3.1). These turn-level tactics allow us to dynamically predict the tactics used in a next utterance given the dialog context. To quantify the effectiveness of each tactic, we further build an outcome predictor to predict the final deal given past tactics sequence extracted from the dialog history (§5). At test time, given the dialog history in each turn, our coach (1) predicts possible tactics in the next turn (§4); (2) uses the outcome predictor to select tactics that will lead to a good deal; (3) retrieves (lexicalized) examples exhibiting the selected tactics and displays them to the seller (§6).

To evaluate the effectiveness of our negotiation coach, we integrate it into He et al.'s (2018) negotiation dialog chat interface and deploy the system on Amazon Mechanical Turk (AMT) (§7). We compare with two baselines: the default setting (no coaching) and the static coaching setting where a tutorial on effective negotiation strategies and tactics is given to the user upfront. The results show that our dynamic negotiation coach helps sellers increase profits by 59% and achieves the highest agreement rate.

2 Problem Statement

We follow the CraigslistBargain setting of He et al. (2018), where a buyer and a seller negotiate the price of an item for sale. The negotiation scenario is based on listings scraped from `craigslist.com`, including product description, product photos (if available), and the listing price. In addition, the buyer is given a private target price that they aim to achieve. Two AMT workers are randomly paired to play the role of the seller and the buyer. They negotiate through the chat interface shown in Figure 2 in a strict turn-taking manner. They are instructed to negotiate hard for a favorable price. Once an agreement is reached, either party can submit the price and the other chooses to accept or reject the deal; the task is then completed.

Our goal is to help the seller achieve a better deal (i.e. higher final price) by providing suggestions on how to respond to the buyer during the conversation. At each seller's turn, the coach takes the negotiation scenario and the current dialog history as input and predicts the best tactics to use in the next turn to achieve a higher final price. The seller has the freedom to choose whether to use the recommended tactics.

3 Approach

We define a set of diverse tactics $\mathcal{S}$ from past study on negotiation in behavioral economics, including both high-level dialog acts (e.g., ⟨*propose a price*⟩, ⟨*describe the product*⟩) and low-level lexical features (e.g. ⟨*use hedge words*⟩). Given the negotiation scenario and the dialog history, the coach takes the following steps (Figure 3) to generate suggestions:

1. The **tactics detectors** map each turn to a set of tactics in $\mathcal{S}$.

2. The **tactics predictor** predicts the set of possible tactics in the next turn given the dia-

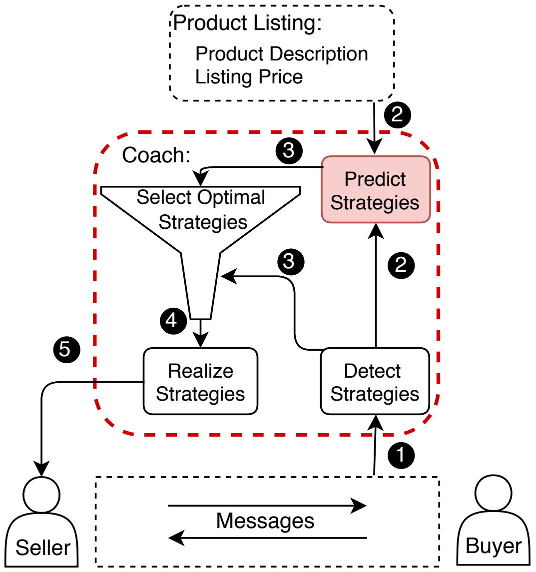

Figure 3: Negotiation Coach Framework. Numbers indicate the time flow.

log history. For example, if the buyer has proposed a price, possible tactics include proposing a counter price, agreeing with the price etc.

3. The **tactics selector** takes the candidate tactics from the tactics predictor and selects those that lead to a better final deal.

4. The **tactics realizer** converts the selected tactics to instructions and examples in natural language, which are then presented to the seller.

We detail each step in the following sections.

3.1 Tactics Detectors

We focus on two broad categories of strategies in behavioral research: (i) **integrative**, or win–win, negotiation, in which negotiators seek to build relationships and reach an agreement benefiting both parties; and (ii) **distributive**, or win–lose, negotiation, in which negotiators adversarially promote their own interests, exert power, bluff, and demand (Walton and McKersie, 1965). In practice, effective negotiation often involves both types of strategies (Fisher and Ury, 1981; Lax and Sebenius, 2006; Pruitt, 1981; K. et al., 2000, *inter alia*).

Prior work typically focuses on conceptual tactics (e.g., emphasize mutual interest), rather than *actionable* tactics in a specific negotiation scenario (e.g., politely decline to lower the price, but offer free delivery). Therefore, we develop data-driven ways to operationalize and quantify these abstract principles.

In Table 1, we list our actionable tactics motivated by various negotiation principles. To detect these tactics from turns, we use a mix of learned classifiers[2] for turn-level tactics (e.g., propose prices) and regular expression rules for lexical tactics (e.g., use polite words). To create the training set for learning tactic predictors, we randomly selected 200 dialogs and annotated them with tactics.[3] The detectors use the following features: (1) the number of words overlapping with the product description; (2) the METEOR score (Denkowski and Lavie, 2014) of the turn given the product description as reference; (3) the cosine distance between the turn embedding and the product description embedding.[4] For "Address buyer's concerns", we additionally include lexical features indicating a question (e.g.,"*why*", "*how*", "*does*") from the immediate previous buyer's turns. Table 2 summarizes the number pf training examples and prediction accuracies for each learned classifier. For lexical tactics, we have the following rules:

- ⟨*Do not propose first*⟩
Waiting for the buyer's proposal allows the seller to better estimate the buyer's target. The detector simply keeps track of who proposes a price first by detecting ⟨*propose a price*⟩.
- ⟨*Negotiate side offers*⟩
The seller sometimes negotiates side offers, e.g., offering a free gift card or free delivery. To detect this strategy, we match the turn against a set of phrases, e.g., "*throw in*", "*throwing in*", "*deliver*", "*delivery*", "*pick up*", "*pick it up*", "*in cash*".
- ⟨*Use factive verbs*⟩
defined in (Hooper, 1975) (e.g. *know*);
- ⟨*Use hedge words*⟩
defined in (Hyland, 2005) (e.g. *could, would*);
- ⟨*Use certainty words*⟩
defined in the LIWC dictionary (Tausczik and Pennebaker, 2010).
- ⟨*Communicate politely*⟩
We include several politeness-related negotiation tactics that were identified by Danescu-

[2] We use ℓ_2-regularized Logistic Regression classifiers.
[3] Each turn can be labeled with multiple tactics.
[4] Sentence embeddings were calculated as the mean of the word embeddings. We used pre-trained word2vec embeddings (Mikolov et al., 2013).

Principle	Action	Example	Detector
		Integrative strategies	
Focus on interests, not positions	Describe the product	*"The car has leather seats."*	classifier
	Rephrase product description	*"45k miles"* → *"less than 50k miles"*	classifier
	Embellish the product	*"a luxury car with attractive leather seats"*	classifier
	Address buyer's concerns	*"I've just taken it to maintainence."*	classifier
	Communicate your interests	*"I'd like to sell it asap."*	classifier
Invent options for mutual gain	Propose a price	*"How about $9k?"*	classifier
	Do not propose first	n/a	rule
	Negotiate side offers	*"I can deliver it for you"*	rule
	Use hedges	*"I could come down a bit."*	rule
Build trust	Communicate politely	greetings, gratitude, apology, *"please"*	rule
	Build rapport	*"My kid really liked this bike, but he outgrew it."*	rule
	Talk informally	*"Absolutely, ask away!"*	rule
		Distributive strategies	
Insist on your position	Show dominance	*"The absolute highest I can do is 640.0."*	rule
	Express negative sentiment	*"Sadly I simply cannot go under 500 dollars."*	rule
	Use certainty words	*"It has always had a screen protector"*	rule

Table 1: Actionable tactics designed based on negotiation principles. Some of them are detected by learning classifiers on annotated data, and the rest are detected using pattern matching.

Niculescu-Mizil et al. (2013) as most informative features. They include: gratitude, greetings , apology, "please" in the beginning of a turn, "please" later on. Keywords matching is used to detect these tactics.

- ⟨*Build rapport*⟩
  Deepening self-disclosure, e.g., "My kid really liked this bike, but he outgrew it", is one strategy for building rapport. We implemented three tactics detectors to identify self-disclosure. First, we count first-person pronouns (Derlaga and Berg, 1987; Joinson, 2001). Second, we count mentions of family members and friends, respectively (Wang et al., 2016). It is done by matching lexicons from *family* and *friend* categories in LIWC.

- ⟨*Talk informally*⟩
  It is detected by matching the keywords in the *informal language* category in LIWC.

- ⟨*Show dominance*⟩
  To detect stubbornness (Tan et al., 2016), we measure the average dominance score of all the words from the Warriner et al.'s (2013)'s dominance ratings of 14,000 words.

- ⟨*Express negative sentiment*⟩
  We measure both positive and negative sentiment by counting words from *positive* and *negative* categories in LIWC.

Strategy	# Ex	Acc
Describe the product	228	0.88
Rephrase product description	136	0.74
Embellish the product	200	0.70
Address buyer's concerns	192	0.95
Propose a price	290	0.88

Table 2: Number of turns annotated (# Ex) and prediction accuracies (Acc) by 5-fold cross validation for learned strategy predictors. Our classifiers achieve high accuracy on all tactics.

4 Tactics Predictor

Armed with a set of negotiation tactics from the dataset, the tactics predictor monitors a negotiation conversation and, at each turn, predicts the seller's next move (e.g., ⟨*propose a price*⟩ or ⟨*express negative sentiment*⟩) given the current dialog context.

Let $u_1, ..., u_t$ denote a sequence of turns, d be a product category, and o_t be a set of tactics occurred in turn u_t. At the $(t + 1)$-th turn in a dialog, given the current dialog context $u_{1:t}$ and d, we want to predict what tactics to use in the response, i.e. o_{t+1}.

The dialog context is represented by embedding the turns, tactics extracted from the turns (§3.1), and the product being discussed. The set of tactics o is a binary vector, where each dimension corresponds to the existence of a certain tactic.

Embedding the turns Embedding of the turns is computed using a standard LSTM encoder over concatenated sequences of words x_i in each turn:

$$h_i^u = \text{LSTM}^u\left(h_{i-1}^u, E^w(x_{i-1})\right),$$

where E^w is the word embedding to be learned.

Embedding the tactics By using the tactics detectors from §3.1, we extract a sequence of tactics $\{m_i\}$ for each turn u in the order of their occurrences from left to right. For example, "*Hi there, I've been using this phone for 2 years and it never had any problem.*" is mapped to "⟨*greetings*⟩ ⟨*use certainty words*⟩". Given turns $u_{1:t}$, we concatenate their tactics in order to form a single sequence, which is embedded by an LSTM:

$$h_i^s = \text{LSTM}^s\left(h_{i-1}^s, [E^o(m_{i-1}); b_{i-1}]\right),$$

where E^o is the one-hot embedding and b is a binary vector encoding tactics that are not specific to a particular word x_i but occur at the turn level (e.g. ⟨*describe the product*⟩).

Embedding the product Different products often induce different expressions and possibly different tactics; for example, renting an apartment often has conversation about a parking lot while selling a phone does not. Thus we also include the product embedding, E^p to encode the product category d, including *car, house, electronics, bike, furniture, and phone*.

The output set of tactics o_{t+1} is a 24-dimensional [5] binary vector, where each dimension represents whether a certain tactic occurred in u_{t+1}. Given the context embedding, we compute the probability of the j-th tactic occurring in u_{t+1} by

$$p(o_{t+1,j}|u_{1:t}, d) = \sigma(W_j[h_t^s; h_t^u; E^p(d)] + b_j),$$

where h_t^s and h_t^u are final hidden states of the tactics encoder and the utterance encoder respectively, and W_j and b_j are learnable parameters. We train the predictor by maximizing the log likelihood of tactics.

[5]Table 1 contains only 15 tactics because some tactics consist of multiple sub-tactics. For example, ⟨*build rapport*⟩ includes two sub-tactics: ⟨*mention family members*⟩ and ⟨*mention friends*⟩.

4.1 Evaluation of the Tactics Predictor

We evaluate the effect of different embeddings on predicting next tactics. We split our data into train, held-out development (20%) and test (20%) data. We then remove incomplete negotiation dialogs (e.g. when the chat got disconnected in the middle). Data sizes are 1,740, 647, and 527 dialogs for train, development and test data respectively. We initialize word embeddings with pre-trained word2vec embeddings. The LSTMs have 100 hidden units. We apply a dropout rate of 0.5 and train for 15 epochs with SGD.

Given the output probabilities $p(o_j)$, we need a list of thresholds γ to convert it into a binary vector, such that $o_j = \mathbb{1}(o_j > \gamma_j)$. We choose γ by maximizing the F1 score of the corresponding strategy on the development set. Specifically, for each strategy, we iterate through all threshold values $[0, 1]$ with a step size of 0.001 and select the one that produces the highest F1 score.

We conduct an ablation study and calculate micro and macro F1 scores. As shown in Table 3, we achieve the best result when combining all components.

Components	Macro F1	Micro F1
Turn Embedding	0.382	0.536
+Product Embedding	0.384	0.539
+Tactics Embedding	**0.397**	**0.592**

Table 3: Effectiveness of turn, product, and tactics embeddings in predicting the next move.

5 Tactics Selector

The tactics predictor outputs a set of tactics o_{t+1}, which can be non-optimal because we only model human behaviors. Now, we implement a *tactics selector* that selects optimal tactics from o_{t+1} under the current dialog context. The major component of the selector is a *negotiation outcome classifier*. This is a supervised classifier that predicts a binary outcome of whether the negotiation will be successful from the seller's standpoint. We next describe the classifier and its evaluation.

Given negotiation tactics and word and phrase choices used by both parties in the previous turns, we train a ℓ_2-regularized Logistic Regression classifier to predict the negotiation's outcome. The outcome is defined as *sale-to-list* ratio r, which is a standard valuation ratio in sales, corresponding to the ratio between the final sale price (i.e., what a buyer pays for the product) and the price originally listed by the seller, optionally smoothed by the buyer's target price (Eq. 1). If the agreed price is between the listed price and the buyer's budget, then $0 \leq r \leq 1$. If the agreed price is greater than the listed price, then $r > 1$. If the agreed price is less than the buyer's budget, then $r < 0$. We define a negotiation as successful if its sale-to-list ratio is in the top 22% of all negotiations in our training data; negative examples comprise the bottom 22%.[6]

$$r = \frac{\text{sale price} - \text{buyer target price}}{\text{listed price} - \text{buyer target price}} \quad (1)$$

The features are the counts of each negotiation tactic from §3.1, separately for the seller and the buyer. A typical negotiation often involves a smalltalk in the beginning of the conversation. Therefore, we split a negotiation into two stages: the 1^{st} stage consists of turns that happen before the first price was proposed, and the 2^{nd} stage includes the rest. We count each tactic separately for the two stages.

Lastly, we apply the classifier to select tactics that will make the negotiation more successful. For each tactic in o_{t+1}, we assume that the seller will use it next by modifying the corresponding input feature in the classifier, which outputs the probability of a successful negotiation outcome for the seller. If the modification results in a more successful negotiation, we select the tactic. For example, if incrementing the input feature of ⟨*describe the product*⟩ $\in o_{t+1}$ increases the probability outputted by the outcome classifier, we select ⟨*describe the product*⟩.

5.1 Evaluation of the Outcome Classifier

The accuracy on test data from Table 4 is given in Table 5. We also evaluate a baseline with shallow lexical features (1-, 2-, 3-grams).

One contribution of this work is that we not only present abstract tactics recommendations (e.g. ⟨*propose a price*⟩), but also propose lexical tactics and examples from successful negotiations (e.g. "Try to use the word *would* like in this sentence: ..."). Table 6 shows that removal of the lexical tactics drops the accuracy by 11%, which is similar to the removal of abstract negotiation tactics. We also find that it is important to separate features in the two stages (before/after the first offer). The 1^{st} stage has weaker influence on the success, while the removal of features in 2^{nd} stage makes the accuracy drop by 24%. Features from both stages contribute to the final score.

	Total	Successful	Unsuccessful
Training	1,740	872	868
Dev	647	316	331
Test	527	259	268

Table 4: Statistics of dialogs, split by successful/unsuccessful negotiations from the seller's standpoint.

Features	Accuracy
Shallow features	0.60
Strategy-motivated features	**0.83**

Table 5: Test accuracy of the outcome classifier with different feature groups

Removed Features	Δ Accuracy
Abstract strategies	-0.12
Lexical strategies	-0.11
Features from the 1^{st} stage	-0.02
Features from the 2^{nd} stage	-0.24

Table 6: Ablation of each subset features shows that lexical tactics are equally important as higher-level abstract tactics and both stages contribute to the final score.

We list seller's top weighted negotiation tactics for both stages in Table 7. ⟨*propose a price*⟩ has the highest weight, which is expected because giving an offer is a fundamental action of negotiation.[7] Following that, the negative weight of ⟨*do not propose first*⟩ indicates that seller should wait for buyer to propose the first price. It is probably because the seller can have a better estimation of the buyer's target price. The second most weighted strategy in the 2^{nd} stage is ⟨*negotiate side offers*⟩, which emphasizes the importance of exploring side offers to increase mutual gain. Moreover, building rapport can help develop trust and help get a better deal, which is supported by the positive weights of ⟨*build rapport*⟩.

Interestingly, some strategies are effective only

[6] The thresholds were set empirically during an early experimentation with the training data.

[7] The reason that ⟨*propose a price*⟩ has zero weights in the 1^{st} stage is that the 1^{st} stage is defined to be the conversations before any proposal is given.

in one stage, but not in the other (the strategies with an opposite sign). For example, ⟨*talk informally*⟩ is more preferable in the 1^{st} stage where people exchange information and establish relationship, while trying to further reduce social distance in the 2^{nd} can damage seller's profit. Another example is that ⟨*express negative sentiment*⟩ is not advised in the 1^{st} stage but has a high positive weight in the 2^{nd} stage. Overall these make sense: to get to a better deal the seller should be friendly in the 1^{st} stage, but firm, less nice, and more assertive in the 2^{nd}, when negotiating the price.

Features	1^{st} stage Weights	2^{nd} stage Weights
⟨*propose a price*⟩	0.0	2.28
⟨*do not propose first*⟩	-0.62	-0.62
⟨*negotiate side offers*⟩	-0.27	1.11
⟨*build rapport*⟩	0.08	0.26
⟨*talk informally*⟩	0.39	-0.39
⟨*express negative sentiment*⟩	-0.05	0.61

Table 7: The table shows the weights of seller's top weighted negotiation tactics in both stages. Positive weight means the feature is positively correlated with the success of a negotiation.

6 Giving Actionable Recommendations

Finally, given the selected tactics, the coach provides suggestions in natural language to the seller. We manually constructed a set of natural language suggestions that correspond to all possible combinations of strategies. For example, if the given tactics are {⟨*describe the product*⟩; ⟨*propose a price*⟩; ⟨*express negative sentiment*⟩}, then the corresponding suggestion is "*Reject the buyer's offer and propose a new price, provide a reason for the price using content from the Product Description*.

As discussed above, we also retrieved examples of some tactics. For instance, ⟨*use hedges*⟩ is not a clear suggestion to most people. To retrieve best examples of ⟨*use hedges*⟩, from all the turns that contain ⟨*use hedges*⟩ in the training data, we choose the one that has a most similar set of tactics to the set of tactics in the current dialog.

7 End-to-End Coaching Evaluation

We evaluate our negotiation coach by incorporating into mock negotiations on AMT. We compare the outcomes of negotiations using our coach, using a static coach, and using no coach.

7.1 Setup and Data

We modified the same interface that was used for collecting data in §2 for the experiments. Moreover, we created 6 test scenarios for the experiments and each scenario was chosen randomly for each negotiation task.

- **No coaching** For our baseline condition, we leave the interface unchanged and collect human–human chats without any interventions, as described in §2.
- **Static coaching** We add a box called "Negotiation Tips", which is shown in a red dashed square in Figure 2. At the beginning of each negotiation, we ask sellers to read the tips. The tips encourage the seller to use a subset of negotiation tactics in §3.1:
  - Use product description to negotiate the price.
  - Do not propose price before the buyer does.
  - You can propose a higher price but also give the buyer a gift card.
  - You can mention your family when rejecting buyer's unreasonable offer, e.g., my wife/husband won't let me go that low.

  Only a subset of tactics was used: the most important and most clear tactics that fit in the recommendation window.
- **Dynamic coaching** We replace "Negotiation Tips" with "Real-Time Analysis" box as shown in Figure 2. When it is the seller's turn to reply, the negotiation coach takes the current dialog context and updates the "Real-Time Analysis" box with contextualized suggestions.

We published three batches of assignments on AMT for three coaching conditions and only allow workers with greater than or equal to 95% approval rate, location in US, UK and Canada to do our assignments. Before negotiation starts, each participant is randomly paired with another participant and appointed to either seller or buyer. During negotiation, seller and buyer take turns to send text messages through an input box. The negotiation ends when one side accepts or rejects the final offer submitted by the other side, or either side disconnects.

We collected 482 dialogs over 3 days. We removed negotiations with 4 turns or less.[8] We further remove negotiations where the seller followed

[8]Sometimes sellers offered a price much lower than the listing price in order to complete the task quickly.

our suggested tactics less than 20% of the time (only 6 dialogs are removed). Our final dataset consists of 300 dialogs, 100 per each coaching condition[9] In the 300 final dialogs, 594 out of 600 workers were unique, only 6 workers participated in negotiations more than once.

7.2 Result

We use two metrics to evaluate each coaching condition: average *sale-to-list* ratio (defined in §5) and task completion rate (%Completion), the percentage of negotiations that have agreements. Moreover, to measure increase in profits (Δ%Profit), we calculate the percentage increase in sale-to-list ratio comparing to no coaching baseline. The result is in Table 8. Dynamic coaching achieves significantly higher sale-to-list ratio than the other coaching conditions, and it also has the highest task completion rate. Comparing with no coaching baseline, our negotiation coach helps the seller increase profits by 59%.

	No Coaching	Static Coaching	Dynamic Coaching
Sale-to-List	0.22	0.19	**0.35**
Δ%Profit	-	-13.6%	**+59.0%**
%Completion	66%	51%	**83%**

Table 8: Evaluation of three coaching models. Improvements are statistically significant ($p < 0.05$).

7.3 Analysis

Here, we first explore the reasons for effectiveness of our dynamic coach and then study why static coaching is least useful.

Why is dynamic coaching better? Manual analysis reveals that our coach encourages sellers to be more assertive while negotiating prices, whereas sellers without our coach give in more easily.[10] We measure *assertiveness* with the average number of proposals made by sellers ⟨*propose a price*⟩: sellers with dynamic coaching propose more often (1.93, compared to 1.32 and 1.08 for no coaching and static coaching respectively). The average number of turns is 8; the measured assertiveness of our coach (1.93) shows that we do not always suggest the seller to reject the buyer's proposal.

Intuitively, an assertive strategy could annoy the buyer and make them leave without completing the negotiation. But, negotiations using our coach have the highest task completion rate. This is likely because in addition to encouraging assertiveness, our coach suggests additional actionable tactics to make the proposal more acceptable to the buyer. We find that 96% of the time, sellers with dynamic coaching use additional strategies when proposing a price, as compared to 69% in static coaching and 61% with no coaching. For example, our coach suggests the seller negotiate side offers and use linguistic hedges, which can mitigate the assertiveness of the request. On the other hand, in no coaching settings, sellers often propose a price without using other tactics. Lastly, the seller often uses almost the same words as shown in the examples retrieved by our suggestions generator in §6. This is probably because sellers find it easier to copy the retrieved example than come up with their own.

The effectiveness of dynamic coaching could in large part be attributed to the *tactics selector* that selects optimal tactics under the current dialog context, but sellers might still use non-optimal tactics even if they are not suggested. To observe the effect of this selecting, we compute the average percentage of *non-optimally* applied tactics. Dynamic coaching has the lowest rate (26%), as compared to no coaching (33%) and static coaching (38%). Moreover, we find that sellers with dynamic coaching often have different chatting styles for exchanging information (1^{st} stage) and negotiating price, while sellers without our coach often use the same style. For example, we show several turns from two dialogs (D_1, D_2) for dynamic and no coaching, respectively. In the 1^{st} stage, our coach suggests sellers to ⟨*talk informally*⟩ with positive sentiment:

- D_1 with dynamic coaching:
  Buyer: "*I'd like to buy the truck.*"
  Seller: "*well that's great to hear! Only 106k miles on it and it runs amazingly. I've got a lot on my plate right now lol so I priced this lower to move it quickly*".
- D_2 with no coaching:
  Buyer: "*I am interested in this truck but I have a few questions.*"
  Seller: "*Absolutely, ask away!*"

The sellers in both dialogs chat in a positive

[9] We randomly sampled 100 dialogs from 108 for no-coaching

[10] For an example, refer to Table 9 in the Appendix; compare lines 24, 26, 28 (our system) against lines 4, 6, 14, 16.

and informal way. However, when negotiating the price, our coach chooses not to select ⟨*talk informally*⟩, but instead suggests formality and politeness, and ⟨*express negative sentiment*⟩ when rejecting buyer's proposal:

- D_1 with dynamic coaching:
  Buyer: "*Would you be willing to take 10k?*"
  Seller: "*That's a lot lower than I was hoping. what I could do, is if you wanted to come see it I could knock off $1500 if you wanted to buy.*".
- D_2 with no coaching:
  Buyer: "*I'm looking for around 10,000.*"
  Seller: "*Oh no. Lol. That's way too low!*"

While the seller with our coach changes style, the seller with no coaching stays the same. We attribute this to the tactics selector. We also find that dynamic coaching leads to a larger quantity and a richer diversity of tactics.

Lastly, we focus on diversity: we show that our coach almost always gives recommendations at each turn and does not recommend the same tactics in each dialog. Specifically, we measure how often our coach gives no suggestions and find out that only 1.8% of the time our coach recommends nothing (9 out of 487 sellers' turns). Then, we calculate how often our coach gives the same tactics within each dialog and find out that only 10% of the time our coach gives the same suggestions (49 out of 487 sellers' turns).

Why is static coaching even worse than no coaching? Surprisingly, static coaching has even lower scores in both metrics than no coaching does. Two possibilities are considered. One is that reading negotiation tips can limit seller's ability to think of other tactics, but we find that static and dynamic coaching use similar number of unique tactics. Then, we explore the second possibility: it is worse to use the tactics in the tips under non-optimal context. Therefore, we measure the average percentage of *non-optimally* applied strategies, but only consider the tactics mentioned in the tips. The result shows that static coaching uses non-optimal tactics 51% of the time, compared to 46% and 38% for no coaching and dynamic coaching, respectively.

8 Conclusion

This paper presents a dynamic negotiation coach that can make measurably good recommendations to sellers that can increase their profits. It benefits from grounding in strategies and tactics within the negotiation literature and uses natural language processing and machine learning techniques to identify and score the tactics' likelihood of being successful. We have tested this coach on human–human negotiations and shown that our techniques can substantially increase the profit of negotiators who follow our coach's recommendations.

A key contribution of this study is a new task and a framework of an automated coach-in-the-loop that provides on-the-fly autocomplete suggestions to the negotiating parties. This framework can seamlessly be integrated in goal-oriented negotiation dialog systems (Lewis et al., 2017; He et al., 2018), and it also has stand-alone educational and commercial values. For example, our coach can provide language and strategy guidance and help improve negotiation skills of non-expert negotiators. In commercial settings, it has a clear use case of assisting humans in sales and in customer service. An additional important contribution lies in aggregating negotiation strategies from economics and behavioral research, and proposing novel ways to operationalize the strategies using linguistic knowledge and resources.

Acknowledgments

We gratefully thank Silvia Saccardo, Anjalie Field, Sachin Kumar, Emily Ahn, Gayatri Bhat, and Aldrian Muis for their helpful feedback and suggestions.

References

Max H. Bazerman, Jared R Curhan, Don A Moore, and Kathleen L Valley. 2000. Negotiation. *Annual review of psychology*, 51(1):279–314.

Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z. Leibo, Karl Tuyls, and Stephen Clark. 2018. Emergent communication through negotiation. In *Proc. ICLR*.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, pages 329–340. ACM.

Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. 2015. Strategic dialogue management via deep reinforcement learning. In *NIPS Workshop on Deep Reinforcement Learning*.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proc. ACL*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proc. WMT*, pages 376–380.

Valerian J. Derlaga and John H. Berg. 1987. Self-disclosure: Theory, research and therapy:. *Springer2*.

Roger Fisher and William Ury. 1981. *Getting to Yes: Negotiating Agreement Without Giving In*. Boston, MA: Houghton Mifflin Company.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Association for Computational Linguistics (ACL)*.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *EMNLP*.

Joan B. Hooper. 1975. *On assertive predicates.* In J. Kimball, editor, Syntax and Semantics, volume 4, pages 91-124.Academic Press, New York.

Ken Hyland. 2005. Metadiscourse: Exploring interaction in writing. *Continuum, London and New York.*

Adam N. Joinson. 2001. Self-disclosure in computer-mediated communication: The role of self-awareness and visual anonymity. *European Journal of Social Psychology, 31(2), 177-192*.

De Dreu C. K., Weingart Laurie R., and Kwon Seungwoo. 2000. Influence of social motives on integrative negotiation: a meta-analytic review and test of two theories. *Journal of personality and social psychology*, 78(5):889.

Simon Keizer, Markus Guhe, Heriberto Cuayáhuitl, Ioannis Efstathiou, Klaus-Peter Engelbrecht, Mihai Dobre, Alex Lascarides, and Oliver Lemon. 2017. Evaluating persuasion strategies and deep reinforcement learning methods for negotiation dialogue agents. In *Proc. EACL*.

David A. Lax and James K. Sebenius. 2006. *3-D Negotiation: Powerful tools to change the game in your most important deals*. Harvard Business Press.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proc. EMNLP*, pages 2443–2453.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proc Association for Computational Linguistics (ACL)*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. In *Proc Conference on Empirical Methods in Natural Language Processing*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems, pp. 3111-3119*.

Alexandros Papangelis and Kallirroi Georgila. 2015. Reinforcement learning of multi-issue negotiation dialogue policies. In *SIGDIAL*.

V. Petukhova, H. Bunt, and A Malchanau. 2017. Computing negotiation update semantics in multi-issue bargaining dialogues. In *Proc. SEMDIAL 2017 (SaarDial) Workshop on the Semantics and Pragmatics of Dialogue, 87-97, DOI: 10.21437/SemDial.2017-10.*

Dean G. Pruitt. 1981. *Negotiation behavior*. New York: Academic Press.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith on-line discussions. In *In Proceedings of WWW*.

Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Leigh L Thompson. 2001. *The mind and heart of the negotiator*, volume 3. Prentice Hall Upper Saddle River, NJ.

Leigh L Thompson, Jiunwen Wang, and Brian C Gunia. 2010. Negotiation. *Annual review of psychology*, 61:491–515.

David Traum, Stacy C. Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. 2008. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *International Workshop on Intelligent Virtual Agents*, pages 117–130. Springer.

Richard E. Walton and Robert B. McKersie. 1965. *A behavioral theory of labor negotiations: An analysis of a social interaction system*. New York: McGraw-Hill.

Yi-Chia Wang, Moira Burke, and Robert Kraut. 2016. *Modeling Self-Disclosure in Social Networking Sites.* CSCW '16, SAN FRANCISCO, CA, USA.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. In *Behavior Research Methods, Volume 45, Issue 4, pp 1191-1207.*

Denis Yarats and Mike Lewis. 2018. Hierarchical text generation and planning for strategic dialogue. In *ICML*.

9 Appendix

Product Listing:

Listing Price: 14500
Buyer's Target Price: 8700
Title: "2006 Toyota 4Runner 4WD - Only 106k Miles - Clean Title"
Product Description:
"Selling my 2006 Toyota 4 Runner with only 106k original miles. The truck is in great condition with no mechanical flaws whatsoever and a clean accident history. Got new tires about 3,000 miles ago. Always has the oil changed on time (due in about 1k). Just got a thorough cleaning inside and a wash and wax outside (still wet in the photos). This truck has never been offroad, but the 4WD is working perfectly from the few times we've been up to Tahoe in it. However, it's a 10+ year old truck that's been driven, not babied and garaged all the time. It's got some scratches, paint is not perfect, but zero body damage."

No Coaching: Seller: S Buyer: B	Static Coaching: Seller: S Buyer: B
1. B: I just saw your ad for the 4Runner, can you send more picture of the scratches?	*10.* S: i would sell it for 1400
2. S: I don't have pictures of the scratches but I can assure you it's minor	*11.* B: you got a deal. 1400 it is
3. B: I might be interested,but all I can offer is $7500	*12.* S: sorry meant 14000
4. S: That is very low. Can I agree to 11000?	*13.* B: Oh c'mon. Now you got my hopes up. lol
5. B: That is too high for me, I mean it is 10 years old with over 100,000 miles. I can possible come up to $8,000	*14.* S: i can go low around 12000
6. S: I can agree to 9,000 and make sure it's had a oil change and tire rotation before you pick it up.	*15.* B: I'm looking at the blue book right now and that still seems a bit high.
7. B: If you're willing to have the oil changed and tires rotated, I will go as high as $8700. And I really can't go any higher.	*16.* S: well the lowest i can go is 10000
8. S: That's fine. I can agree to that.	*17.* B: You mention scratches. Lets be real. How bad are we talking?
9. B: Thanks, I'll be right over to pick it up.	*18.* S: its 10 yrs old it has some scratches but has zero body damage.
	19. B: Ok. Without seeing any more photos, 9000 is the best I can do.
	20. S: deal
	21. B: deal. Thanks!

Dynamic Coaching:
Seller: S Buyer: B

22.	S:	Would you be interested in buying my 4Runner?
23.	B:	Yes, I am possibly. interested Given that is is over 10 years old, all I can offer now is 8000.
24.	Coach:	*Reject Buyer's Offer and Propose a New Price, Reason the Price with Content in the Product Description* *Try to Use the Word "Would" Like This: "Sorry, I would really need $100 for it."* ...
	S:	I'm sorry, but I **would** really need $12,000 for it. **It's in great condition!**
25.	B:	Well given the mileage on the car it is about to be due for some major service which can be expensive. How about 9000?
26.	Coach:	*Reject Buyer's Offer and Propose a New Price* *Try to Use the Word "Could" Like This: "I could come down to $3."* *You Can Also Give the Buyer Something for Free (Gift Card) to Argue for Higher Price.* ...
	S:	I **could** come down to $11,000 and **include the roof rack as a bonus**.
27.	B:	Add in a full tank of gas and a cold beer and I could do 10500.
28.	Coach:	*Reject Buyer's Offer and Propose a New Price* *Talk About Your Personal Experience With The Product* *Try to Use the Word "Would" Like This: "$160 would be a good price."* ...
	S:	**I always took great care of the truck.** I think $10,750 **would** be a reasonable price.
29.	B:	Deal.
30.	S:	Great

Table 9: Examples of collected negotiation dialogs.