Intelligent Negotiation Bot using Machine Learning Techniques

Conference Paper · August 2022

DOI: 10.1109/ASIANCON55314.2022.9908710

Dipti Jadhav
Ramrao Adik Institute of Technology

2022 2nd Asian Conference on Innovation in Technology (ASIANCON)
Pune, India. Aug 26-28, 2022

Intelligent Negotiation Bot using Machine Learning Techniques

Kabir Batra
Department of Information Technology
Ramrao Adik Institute of Technology,
D. Y. PATIL Deemed to be University
Navi Mumbai, India
kab.bat.rt18@rait.ac.in

Niraj Nair
Department of Information Technology
Ramrao Adik Institute of Technology,
D. Y. PATIL Deemed to be University
Navi Mumbai, India
nir.nai.rt18@rait.ac.in

Atharva Chaudhary
Department of Information Technology
Ramrao Adik Institute of Technology,
D. Y. PATIL Deemed to be University
Navi Mumbai, India
ath.cha.rt18@rait.ac.in

Dipti Jadhav
Department of Information Technology
Ramrao Adik Institute of Technology,
D. Y. PATIL Deemed to be University
Navi Mumbai, India
dipti.jadhav@rait.ac.in

2022 2nd Asian Conference on Innovation in Technology (ASIANCON) | 978-1-6654-6851-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASIANCON55314.2022.9908710

***Abstract*— E-commerce, in today's day and age is becoming increasingly competitive, and there's an emerging need for new technological advancements in order to stand in this competitive market. The employment of a negotiation bot is a new AI technology that has a growing appeal in E-commerce. The goal of a negotiation is not just to get the most money for yourself, but also to persuade your opponent that the deal is good for him. Businesses can use negotiation bots to personalize rates for each consumer and offer them discounts. The majority of people lack bargaining skills when it comes to trading, which causes them to miss out on a fair and lucrative offer. This paper proposes to develop an intelligent chatbot that can act as sellers in bargaining conversations with buyers.**

Keywords—Negotiation Bot, Intent Classification, Machine Learning, AI

I. INTRODUCTION

People naturally get an itch to negotiate, to get the best price for everything they buy. It's an important skill to have for everyday conversations as well as business collaborations. As seen from a wider perspective it benefits both the parties in closing a deal. Negotiation, given its spectrum of complexity varying in each situation, can take someone years of experience to have the intellect for presenting a lucrative offer. AI as we all know, works on a simple principle of responding to a set of algorithms and data that has previously been tested, predicting methods and calculating the figures more precisely and efficiently than humans. This predictive data can be used as a general framework for grouping unique characteristics. As a result, combining machines and planning has become the go-to strategy for improving people's decision-making in crucial situations. It could help a person who lacks the required understanding to decide whether to buy a product at a given price and help improve clients with a relative assessment for improvement.

The first and the most challenging step of any negotiation process is knowing how to execute a negotiation appeal and constructing a mutually beneficial agreement with limited knowledge and experience. AI can act as a liaison between vast volumes of data on a particular product and the skills and knowledge required to engage and sustain a negotiation. This whole process can be automated to enable both vendors and buyers to make money fairly and without bias. With the introduction of AI people can be more confident in closing deals which will be mutually beneficial and save the hassle of repetitive back and forth offers. In this subject, artificial intelligence will revolutionize the way people negotiate and instill the virtue of good faith in all areas. The main aim in this paper is to create a responsive chatbot that uses AI to overcome the shortcomings of in-person negotiations.

Negotiation varies from person to person as it can be simple or can also be complex at times. The process of negotiating and closing a deal can be an exhausting task. Unfortunately, many people lack the necessary skills to strongly communicate, comprehend the situation, and have a stance in order to seal the deal.

II. LITERATURE REVIEW

J. Park, H. A. Rahman, J. Suh, and H. Hussin [1] were able to develop a prototype of an agent which negotiated with humans to find a win-win situation for both parties. Integrative bargaining model and social judgment theory forms the basis of their research. The limitations as stated in [1] was that they couldn't find out the buyer's interest at the start of the negotiation.

T. Liu and Z. Zheng, [2] developed a chatbot which helped buyers interested in buying used cars in a smart manner using different machine learning and deep learning models at various stages like predicting the suitable price, deciding the next strategy in negotiation and judging the offered deal.

J. R. Oliver [3] compares the outcomes of negotiations performed by an artificial adaptive agent and humans. The agent is developed using genetic algorithms and it is evaluated on the basis of its performance for different bargaining problems.

Frank Burchill [4] in his work has discussed fundamental aspects of various theories of negotiation. The authors also discuss the cases in which these theories would help to find a solution in workplace conflict.

P. Henderson, S. Crouch, R. J. Walters, and Q. Ni [5] examined the performance of seven automated bargaining algorithms for a car hiring scenario by conducting a tournament. They concluded after the event that the optimal algorithm is the one that reaches to an agreement within the specified number of rounds.

He. He, D. Chen, A. Balakrishnan, and P. Liang [6] decouple negotiating strategy and natural language processing, allowing them to have more control over the approach and increase the likelihood of a fair transaction.

The above literature survey shows that negotiation bots developed did not focus on proper pricing in their bargains. This paper proposes a negotiation bot for one-on-one bargains with the customers.

III. DATASETS

Dataset lies at the heart of any machine learning project, the Craigslist Bargain dataset which was created in [6] consists of negotiation conversations between buyer and seller for products from varying domains like electronics, furniture, etc. This dataset contains 6,682 dialogues over multiple categories of products

Deal or No Deal dataset created in [7] is being used to improve prediction for some intents such as deal (agreement) or no deal (disagreement) instances which had less occurrence in Craigslist Bargain dataset. It contains a total of 12,234 dialogues from which helps us efficiently identify a deal (agreement) or no deal (disagreement) instance. This dataset helped us in increasing and diversifying the text mapped to the aforementioned intent.

In the proposed technique for Sentiment Analysis Dataset from Kaggle is used. It contains tweets annotated as positive, negative or neutral based on the sentiment of the tweet. The dataset contains about 32,296 tweets and other different data of those tweets like time of tweet, age of buyer, country, etc.

IV. PROPOSED METHODOLOGY

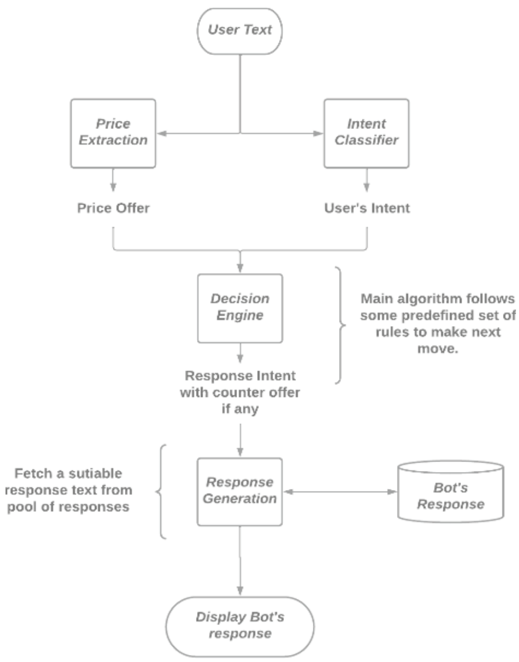

Fig. 1. Block Diagram of Proposed System

The proposed method for Negotiation Bot can be described by key components like taking input from the user, classifying intent and price extraction, deciding new offer and lastly response generation. Fig. 1 showcases a complete cycle to generate response for a user's input. Functions carried out at every stage are as follows:

1) Firstly, the buyer has to choose the desired product for negotiation and start a new conversation with a text.
2) In intent classification, the buyer's message is used to define the buyer's intention.
3) Also, the price extraction is performed on the buyer's message if any.
4) Both outputs from above two processes are used to generate a response according to buyer's intent with a new price offer.
5) The predicted price offer and bot's intent is passed to response generation to create a reply text from the bot to the buyer.
6) Finally, the reply text is displayed on the current chat window. The buyer has to then communicate back and forth to proceed negotiation and this process is repeatedly followed for all buyer inputs.

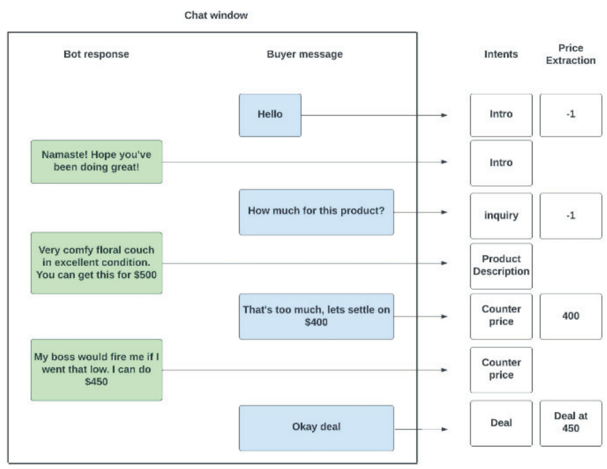

Fig. 2. Basic working of Negotiation Bot

A. Intent Classification

This process identifies the intent of the buyer's message as shown in Fig. 2, for e.g., identify whether the user is proposing an offer or accepting a previous offer or counter-pricing, etc. There are a total of 8 intent classes, including Introduction / Greeting, Inquiry, Counter- price, Agree/Accept, Disagree/Deny, Insist and Vague-price. Craigslist's Bargain Dataset is used which consists of negotiation conversations between buyers and sellers from an e-commerce platform.

Firstly, the conversations were broken down into single text replies in a row. Then pre-process the text data which involves removing punctuations, removing special characters, numbers, stemming and converting text to lowercase. The pre-processed data is then vectorized using CountVectorizer and TFIDF Vectorizer to create a vocabulary list of words present in the dataset. The vectorized output from both the vectorizers are then passed to seven classification models for training. Figure 3. shows the accuracies of various models trained with different vectorizers.

XGBClassifier along with Count Vectorizer was chosen to classify intents as it obtained the highest f1 score among all other models shown in Fig. 3. Whenever the buyer sends a message, the model predicts and saves its intent into a variable named "buyer_intent" and also saves this into an array named "intent_timeline".

For e.g. If the buyer sends a message like "Hello there", the model detects intent as "Intro" or for messages like "Can I get this product for $100?", the model detects intent as "Counter-price" i.e., buyer_intent = "Counter-price".

Index	Model	f1 score	Train accuracy	Test accuracy	Vectorizer
0	XGBClassifier	0.712978	90	71	Count vector
1	XGBClassifier	0.707597	94	71	Tfidf vector
7	RandomForestClassifier	0.702201	96	70	Tfidf vector
3	SVC	0.696578	79	70	Tfidf vector
6	RandomForestClassifier	0.694077	96	70	Count vector
5	LogisticRegression	0.682029	76	68	Tfidf vector
2	SVC	0.67864	83	68	Count vector
4	LogisticRegression	0.671222	79	67	Count vector
10	Decision TreeClassifier	0.623013	96	62	Count vector
11	Decision TreeClassifier	0.609111	96	61	Tfidf vector
8	MultinomialNB	0.598213	67	60	Count vector
9	MultinomialNB	0.547914	63	56	Tfidf vector
12	KNeighbors Classifier	0.534614	68	53	Count vector
13	KNeighbors Classifier	0.448942	56	40	Tfidf vector

Fig. 3. Comparison between various classifiers.

B. Price Extraction

The price offer is extracted from the message string when a buyer sends a message as shown in Fig. 2. If a buyer is making a price offer, he must put the number value (price offer) after the "$" sign.

Incorrect Format: "How about 100 for this?" Correct Format: "How about $100 for this?"

Current buyer's offer is saved in a variable called "buyer_offer" i.e., buyer_offer = 100

C. Decision Engine and Counter Pricing

It evaluates buyer's current intent (buyer_intent) and price offer (buyer_offer) from the previous process to decide the Bot's response strategy and whether there is a need to give them a discount or not, if yes then by how much.

The Bot will try to settle at a price offer that falls within the seller's pricing range for the product. The seller specifies this price range at the beginning as variables "product_higher_limit" and product_lower_limit". E.g if a seller sets a range between $100 to $150 i.e product_higher_limit = 150 and product_lower_limit = 100.

There are two variables to store Bot's response intent and offered price called "bot_offer" and "bot_intent". These values are predicted on the basis of the all-previous buyer intents (intent_timeline) and using some predefined strategy which are as follows:

1) If buyer_intent is "Intro" (buyer_intent = "Intro"), Bot should respond with an intro text. Therefore, bot_response = "Intro"
2) Similarly, if buyer_intent is "Inquiry" (buyer_intent = "Inquiry"), Bot should respond with product description. Therefore, bot_response = "product_desc".
3) If the buyer offers a price near the seller's defined price range (between product_higher_limit and product_lower_limit), then Bot either accepts the offer (bot_response = "Agree") or counter with a new price offer (bot_response = "Counter-price").

Decision tree is trained for the new price offer using the below preprocessed data that can predict the discount percentage.

Fig. 4 shows the dataset used for price prediction in which the first column contains messages from a bargaining conversation. Every message's intent is present in the second column. The discount granted at that particular conversation is listed in the third column.

The trained model uses all intents (intent_timeline) from the present conversation as input to predict discount percentage and stores this value in a variable named "discount". Then, by subtracting the discount % (percentage) from the product's higher limit (product_higher_limit) set by the seller, a new price offer is calculated.

For e.g., if intents in a current conversation are ["intro", "inquiry", "counter-price"," disagree", "counter-price"," counter-price"], this array of intents are fed as input to the model to predict discount percentage. The equation to calculate a new offer is as shown in Eq. no (1):

$$new\_offer = product\_higher\_limit \times (100 - discount) / 100 \quad (1)$$

If new_offer is smaller than buyer_offer, then bot_response switches to "Agree" at buyer_offer. Else if new_offer is lower than product_lower_limit, then new_offer is set to product_lower_limit i.e new_offer = product_lower_limit. Else if new_offer is greater than Bot's last offer, then bot_response switches to "Insist" with Bot's last offered price.

4) If the buyer insists on his previous offer (buyer_intent = "Insist"), Bot should respond with his final offer. Therefore, bot_response = "Insist" with final offer. And if the buyer agrees to this offer, then bot_response switches to "Agree" else "Disagree".
5) If the buyer offers a price too lower than the seller's defined price range (buyer_intent = "Vague-price"), Bot should respond with a recommendation text to increase the buyer's offer. Therefore, bot_response = "Vague-price".
6) If the buyer agrees to the current offer (buyer_intent = "Deal"), Bot should respond with a done deal text with a mutually settled offer. Therefore, bot_response = "Deal".
7) If the buyer disagrees with the current offer (buyer_intent = "Disagree"), Bot should respond with a counter offer. Therefore, bot_response = "Counter-price". And If the Bot predicts the same price offer more than twice then it switches to "Disagree". Therefore, bot_response = "Disagree".
8) It's a special case when a buyer points out some negative traits of the product. A flag named "Negative_context" is set to "True".

If Negative_context = True, then Bot gives an additional discount to the customer. The seller determines this percentage value for additional discount at the beginning.

Negative_context = “True”, only if buyer sentiment is “Negative” with Cosine similarity score greater than threshold value.

To detect buyer sentiment, a classification model was trained to determine whether the sentiment of a buyer's message is "Positive" or "Negative". A Logistic Regression model was fitted on the texts in the Sentiment Analysis Dataset after vectorizing it using CountVectorizer which obtained an accuracy of 89% in classification.

Understanding the sentiment of the buyer’s text as negative is not enough for the Bot to give additional discount. It also requires checking whether the buyer is actually talking about this product. In order to verify this, cosine similarity was used to measure the similarity between the buyer’s text and the product’s description. If the value of cosine similarity score exceeds the threshold value, then the Bot gives an additional discount. The threshold value depends on the length of the product’s description.

bargain_convo	intents	discount%
[Hi, not sure if the charger would work for my car. Can you sell it to me for $5?, It will work,...	[intro, counter-price, counter-price, agree]	30.0
[Hi,I am interested in your board!, Hi, do you have any questions?, Do you know what wood it is ...	[intro, disagree, vague-price, counter-price, agree, agree]	25.0
[hello, Hi, How old is the couch?, It is practically brand new, As you know, the value goes down...	[intro]	17.0
[man thats a nice trak, yes its really nice, how about you give it to me for 8000, That is way t...	[intro, counter-price, agree]	10.0
[Hi, I'd love to have this sofa, but I just can't pay that much for a used one. Would you take $...	[intro, counter-price, counter-price]	NaN
...	...	...
[Hello I was interested in your 2011 fusion hybrid. , Hello. Great! It is a nice little car , It...	[intro, inquiry, counter-price]	24.0
[Hi there! I'm interested in the chair you have posted. I need something for my home office an...	[intro, disagree, counter-price]	28.0
[hello I am interested in the bike you are selling, Awesome!! It's a great bike and I haven't ha...	[intro, disagree, counter-price, counter-price]	13.0
[Hi there! I saw those lamps you have posted, and I love them! Do they both work?, Yes they ar...	[intro, disagree, agree]	NaN
[Hi there. Are you interested in the place?, Yes, I am. The location is ideal. How many beds a...	[intro, init-price, counter-price]	11.0

Fig. 4. Dataset for discount prediction

D. Response Strategy

This part solely focuses on generating a response by using formulated output obtained in the previous part. A pool of responses was created for every Bot Response. For e.g. If

```

counterprice = [
"My boss would fire me if I went that low.. how about {}.",
"Umm no. let's settle on {}.",
"Best I can do is {}.",
"Your offer sounds low, I've got something better, let's do {}.",
"Let's end it at {}, it's best for both of us.",
"Let's meet somewhere in the middle at {}.",
"You're a tough one, but so am I.. {} is my offer.",
"Ahhh I've had a rough day, let's make it quick {} sounds awesome.",
"I've got a family to take care of let's do {}.",
"I've an offer no one would have given you {}."
]

deal = [
"Glad you made the right decision.",
"I've got another offer, a virtual hand shake.",
"Pleasure doing business with you!",
"phew, That was an intimidating negotiation. Cheers!",
"Glad we could reach to a consensus, enjoy your product.",
"You made the right call, you wont regret.",
"Good negotiating, Celebrate your victory.",
"Finally i can take some rest!",
"Trust me you got this at the best possible price.",
"We've got a deal, some of my favourite words."
]

```

Fig. 5. Pool of responses for specified intents

the bot_response = Deal then the Bot replies using a text from the pool of text responses mapping to the response “deal” as shown in Fig. 5.

V. EXPERIMENTAL RESULTS

In Fig. 6, User is trying to make a deal for a couch but failing to settle on the price.

In Fig. 8 User is trying to make a deal for a cabinet and successfully settling on a price with an additional discount because the user argued on it being old.

Fig. 7 and Fig. 9 show logs for our conversation.

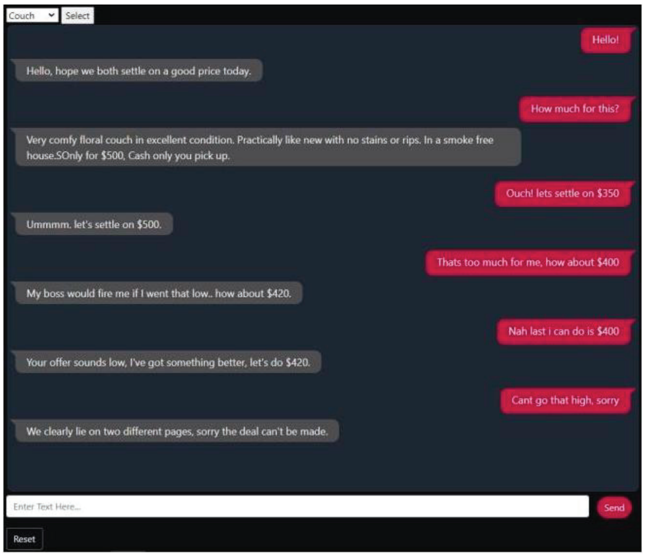

Fig. 6. No deal being made on couch

```
current buyer's intent: agree
buyer's sentiment intent: negative
Cosine Similarity:  [[0.]]
negative context: False
current buyer's offer: 0
Buyer's timeline:
[['buyer_intent' 'buyer_bid']
 ['intro' '0']
 ['inquiry' '0']
 ['counter-price' '3800']
 ['counter-price' '4000']
 ['agree' '0']]
Bot's timeline:
[['bot_intent' 'bot_bid']
 ['intro' '0']
 ['inquiry' '0']
 ['counter-price' '5000']
 ['counter-price' '4250']
 ['agree' '4250']]
```

Fig. 7. Snapshot of the terminal

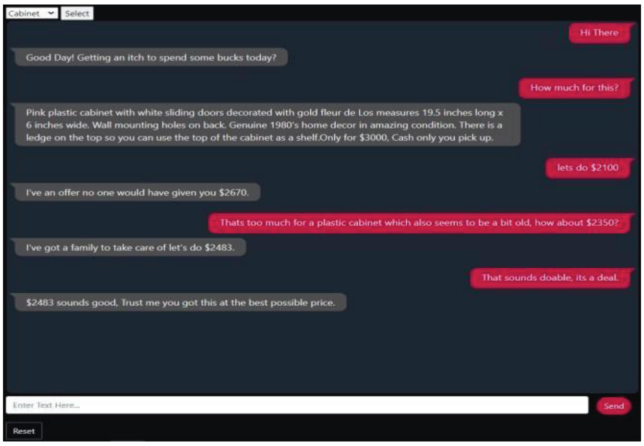

Fig. 8. Deal being made on cabinet

```
current buyer's intent: init-price
buyer's sentiment intent: negative
Cosine Similarity:  [[0.26516504]]
negative context: True
current buyer's offer: 2350
Buyer's timeline:
[['buyer_intent' 'buyer_bid']
 ['intro' '0']
 ['inquiry' '0']
 ['counter-price' '2100']
 ['init-price' '2350']]
C:\Users\athar\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTree
Classifier from version 0.24.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at y
our own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  warnings.warn(
['intro', 'inquiry', 'counter-price', 'init-price']
[1, 6, 4, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Price prediction: 3000.0 Discount% [0]
case4
True
log: Negative context set to true, offering additional discount
2483.1 2350
Price after approximation: 2483.1
Bot's timeline:
[['bot_intent' 'bot_bid']
 ['intro' '0']
 ['inquiry' '0']
 ['counter-price' '2670']
 ['init-price' '2483']]
```

Fig. 9. Snapshot of the terminal

VI. PERFORMANCE EVALUATION

The proposed technique for Negotiation Bot was subjectively reviewed by ten testers from diverse categories, including our coworkers, faculty members, and research scholars from our department. The response we received was fairly positive as shown in Table 1, with 8 out of 10 participants rating our negotiating bot remarkably, and the other two being just satisfied. The UI was simple for everyone to comprehend, and no one had any difficulty figuring out how to start the dialogue. Everyone was able to achieve some sort of agreement at the conclusion of negotiation. Users really enjoyed the additional discount granted on the product's bad aspects.

TABLE I. EVALUATION OF BOT

Good	Acceptable	Bad
80%	20%	0

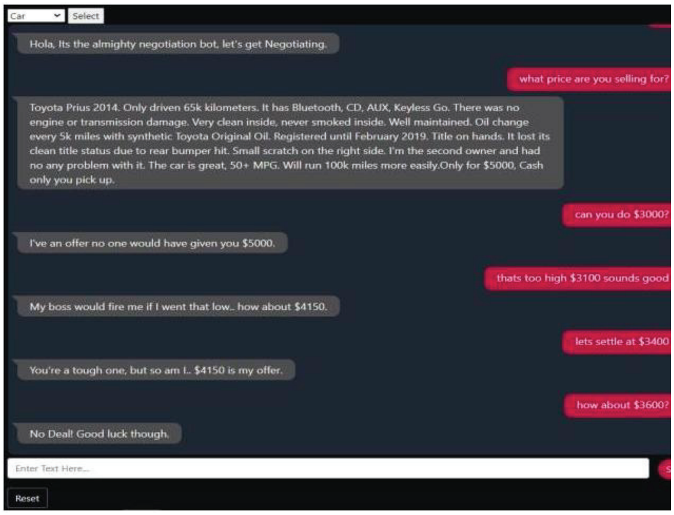

Fig. 10. Conversation between Bot and a participant

VII. CONCLUSION

The proposed intelligent chatbot assists sellers in negotiating by employing negotiation strategies and machine learning techniques. It is also able to present fair arguments and maintain a smooth communication process throughout. The evaluation results demonstrated that the bot correctly identifies the intents and responds accordingly aiming to close the deal in an effective manner. The results depict that artificial intelligence can be successfully employed to the development of bargaining abilities, which most people certainly tend to lack.

VIII. FUTURE SCOPE

Future scope is to improve our chatbot by adding text-to-speech so that our bot can fulfill a variety of user needs. With text-to-speech negotiation can be taken to a whole new level by setting up this bot in commercial spaces hence automating the bargaining process.

REFERENCES

[1] J. Park, H. A. Rahman, J. Suh, and H. Hussin, "A study of integrative bargaining model with argumentation-based negotiation," *Sustain.*, vol. 11, no. 23, Dec. 2019, doi: 10.3390/su11236832.

[2] T. Liu and Z. Zheng, "Negotiation Assistant Bot of Pricing Prediction Based on Machine Learning," *Int. J. Intell. Sci.*, vol. 10, no. 02, pp. 9– 21, 2020, doi: 10.4236/ijis.2020.102002.

[3] J. R. Oliver, "A Machine-Learning Approach to Automated Negotiation and Prospects for Electronic Commerce," *J. Manag. Inf. Syst.*, vol. 13, no. 3, pp. 83–112, 1996, doi: 10.1080/07421222.1996.11518135.

[4] F. Burchill, "Walton and McKersie, A Behavioral Theory of Labor Negotiations (1965)," *Hist. Stud. Ind. Relations*, no. 8, pp. 137–168, Sep. 1999, doi: 10.3828/hsir.1999.8.7.

[5] P. Henderson, S. Crouch, R. J. Walters, and Q. Ni, "Comparison of some negotiation algorithms using a tournament-based approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2003, vol. 2592, pp. 137–150, doi: 10.1007/3-540-36559-1_12.

[6] He. He, D. Chen, A. Balakrishnan, and P. Liang, "Decoupling strategy and generation in negotiation dialogues," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020, pp. 2333–2343, doi: 10.18653/v1/d18-1256.

[7] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, and D. Batra, "Deal or no deal? End-to-end learning for negotiation dialogues," EMNLP 2017- Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 2443–2453, 2017, doi: 10.18653/v1/d17-1259.