CONSIDERING ARGUMENTS IN HUMAN-AGENT NEGOTIATIONS

A Thesis

by

Anıl Doğru

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
January 2023

CONSIDERING ARGUMENTS IN HUMAN-AGENT NEGOTIATIONS

Approved by:

Asst. Prof. Reyhan Aydoğan
Dept. of Computer Science
Özyeğin University

Prof. Olcay Taner Yıldız
Dept. of Computer Science
Özyeğin University

Assoc. Prof. Arzucan Özgür
Dept. of Computer Eng.
Boğaziçi University

Date Approved: 22 November 2022

To my precious family

ABSTRACT

Autonomous negotiating agents, which can interact with other agents, aim to solve decision-making problems involving participants with conflicting interests. Designing agents capable of negotiating with human partners requires considering some human factors, such as emotional states and arguments. For this purpose, we introduce an extended taxonomy of argument types capturing human speech acts during the negotiation and propose an argument-based automated negotiating agent that can extract human arguments from a chat-based environment using a hierarchical classifier. Consequently, the proposed agent can understand the received arguments and adapt its strategy accordingly while negotiating with its human counterparts. We initially conducted human-agent negotiation experiments to construct a negotiation corpus to train our classifier. According to the experimental results, it is seen that the proposed hierarchical classifier successfully extracted the arguments from the given text. Moreover, we conducted a second experiment where we tested the performance of the designed negotiation strategy considering the human opponent's arguments and emotions. Our results showed that the proposed agent beats the human negotiator and gains higher utility than the baseline agent.

ÖZETÇE

Diğer etmenlerle etkileşime girebilen otonom müzakere etmenleri, birbirleriyle ilgi ve çıkarları çatışan katılımcıların karar verme sorunlarını çözmeyi amaçmaktadır. İnsanlarla müzakere edebilen etmenleri tasarlamak, rakibin duygu durumu ve argümanları gibi bazı faktörleri göz önünde bulundurmayı gerektirmektedir. Bu amaçla, bu tezde müzakere sırasında insan rakibin konuşma eylemlerini yakalayan argüman türlerinin genişletilmiş bir taksonomisini sunulmakta ve hiyerarşik bir sınıflandırıcı kullanarak sohbet tabanlı bir ortamdan insan argümanlarını işleyebilen argüman tabanlı otomatik bir müzakere etmeni önerilmektedir. Böylece, önerilen etmen alınan argümanları anlayabilir ve insan rakipleriyle müzakere ederken stratejisini bu argümanlardan elde ettiği bilgiye göre uyarlayabilmektedir. İlk önce, sınıflandırıcımızı eğitmek için kullanacağımız müzakere külliyatını oluşturmak üzere, insan-etmen müzakere deneyleri alınmıştır. Deneysel sonuçlara göre, önerilen hiyerarşik sınıflandırıcının verilen metinden argümanları başarıyla çıkardığı gözlemlenmiştir. Ayrıca, insan rakibin argümanlarını ve duygularını dikkate alarak tasarlanan müzakere stratejisinin performansını incelemek için ikinci bir deney daha gerçekleştirilmiştir. Deneylerden elde edilen sonuçlar, önerilen etmenin insan rakibini genelde yendiğini ve kıyaslanan temel etmenden daha yüksek fayda elde ettiğini göstermektedir.

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to Asst. Prof. Reyhan Aydoğan who gave me the golden opportunity to do this wonderful thesis on the topic. I would also like to thank all members of Interactive Intelligent Systems especially Onur Keskin and Berk Buzcu for their support.

I am deeply grateful to Assoc. Prof. Arzucan Özgür and Prof. Olcay Taner Yıldız for participating in my thesis jury and giving me feedback.

Last but not least, I would love to thank to my dearest family for their endless support over those years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

α	A parameter of the issue weight update in the opponent model
α_{lower}	Lower bound for the target utility in our bidding strategy
β	A parameter of the issue weight update in the opponent model
δ	Thresholds for our windowed offer search strategy
γ	Smoothing factor in the opponent model
$\hat{V}_j^i$	The estimated value evaluation of an issue value in a negotiation
$\hat{W}_i$	The estimated importance of a negotiation issue
E^t	Emotion effect at the corresponding negotiation time
i	Negotiation issue
j	An issue value in a negotiation
o	An offer in negotiation
O^t	Emotional state at the corresponding negotiation time
O_h^t	Offer of human at the corresponding negotiation time
O_j^t	Offer of agent at the corresponding negotiation time
P_0	The maximum target utility at the beginning of the negotiation for bidding strategy
P_1	The concession speed for bidding strategy
P_2	The minimum target utility at the end of the negotiation for bidding strategy

P_3	The regulator parameter for behaviour-based agent for bidding strategy
P_4	The regulator parameter for emotion-based agent
P_A	Opponent awareness coefficient for bidding strategy
P_E	Emotion coefficient for bidding strategy
t	Normalized negotiation time
$U(.)$	Utility function
V_i	Valance score of an emoji
V_j^i	The value evaluation of an issue value
W_i	The importance of an issue
AC	Acceptance strategy
RU	Reservation Utility
TU	Target utility

LIST OF ACRONYMS/ABBREVIATIONS

BERT Bidirectional Encoder Representations from Transformers.

BOW Bag of Words.

FN False Negative.

FP False Positive.

KNN K-Nearest Neighbour.

LSTM Long-Short Term Memory.

Mix. Average Mixture of Experts with Averaging Method.

Mix. Weighted Mixture of Experts with Weighted Averaging Method.

MLC Multi-Label Classification.

MLP Multilayer Perceptron.

NLG Natural Language Generating.

NLP Natural Language Processing.

NLU Natural Language Understanding.

RL Reinforcement Learning.

RMSE Root-Mean-Square Error.

SL Supervised Learning.

TN True Negative.

TP True Positive.

CHAPTER I

INTRODUCTION

Negotiation is one of the most widely-used resolution process for decision-making problems involving participants with conflicting interests in multi-agent systems [3]. Participants can negotiate over any joint decision (e.g., joint plan, resource allocation). This process can be automatically performed via the use of intelligent agents. During the negotiation, the behavior of an autonomous agent depends on various factors such as time pressure, opponent's behavior, negotiation domain, and so on [3, 4]. Negotiating with their human counterpart requires designing more sophisticated strategies taking into account human arguments and emotions for a better understanding of their preferences and behaviors so that well-targeted offers could be generated. Consequently, agents can act wisely to achieve better negotiation outcomes.

Depending on the context, autonomous negotiating agents aim to maximize their utility or increase social welfare under some extent of uncertainty about their opponent. The main components of such an agent involve bidding strategy, opponent modeling, and acceptance strategy [5]. The bidding strategy decides what to offer. For this decision, agents mostly calculate a target utility considering the remaining time and/or opponent's behavior and make an offer whose utility is closest to the calculated target utility [6, 7, 8, 9, 10]. The acceptance strategy decides when to accept the opponent's offer. In the literature, some strategies only consider the received utility, whereas others consider remaining time as well [11]. Lastly, agents negotiate without revealing their preferences fully. Therefore, one of main the challenges is to estimate the opponent's preferences and, eventually, strategy based on the offer

exchanges [2, 12]. Accordingly, an agent can adapt its behavior and bidding strategy. To achieve this, opponent's bid history needs to be analyzed to estimate an accurate opponent model. Furthermore, agents should utilize other inputs, such as arguments exchanged by their opponents.

Negotiating agents concerning the human opponents' expectations would perform better against their human counterparts. For instance, the Solver Agent analyzing the facial expressions of the human negotiators and adapting its behavior depending on the opponent's emotional state achieves better negotiation outcomes [9]. Moreover, acting like a human negotiator (e.g., robot gestures, facial expressions with avatars) improves the human-agent negotiation process [13, 14, 15]. Therefore, a multi-modal agent design utilizing these factors as much as possible plays a key role for better negotiation outcomes as well as an effective interaction.

During the negotiation, human negotiators can express their preferences, share arguments, express their emotional states, and specify their hard constraints in a chosen natural language [16, 17, 18, 19]. Therefore, understanding and generating arguments in a natural language is one of the essential research questions in artificial intelligence [20, 21]. Understanding human arguments and incorporating the knowledge embedded in natural language may significantly improve negotiation ability and, consequently, negotiation outcome. Accordingly, this thesis addresses how agents can recognize human arguments and utilize them to improve their gain while negotiating with a human negotiator.

Enabling users to use their natural language to express their interests and emotional states has attracted attention in recent decades. Researchers develop several frameworks such as NegoChat [22] and IAGO [23] supporting natural language. In contrast, others aim to collect some datasets from human-human negotiation sessions to enable agents to generate its offers/arguments in a chosen natural language

[16, 17]. The CaSiNo dataset contains some labeled argument types [18], but it cannot recognize issue-value constraints directly. Some datasets involve emoji exchanges to express emotional states but need to capture free-format sentences to get insights about their opponent's emotional states. Therefore, we aim to construct a dataset involving various arguments that could be useful to utilize during the negotiation.

Güngör *et al.* categorize three main argument types (i.e., rewarding, explanatory, and threatening), which can be exchanged in human-agent negotiations [24]. However, those argument types are not detailed enough for a sophisticated agent design. This study extends those argument types by defining new sub-categories. The proposed hierarchical argument-type structure is an opportunity for designing an agent concerning human arguments. Thus, this study proposes a novel agent design that considers human arguments in enhancing opponent modeling and in well-targeted offer generation mechanism. To meet the aforementioned needs, we collected a dataset consisting of human arguments with corresponding labels. A hierarchical machine learning approach is applied while training models to extract argument types from a text. Consequently, we present a negotiation strategy utilizing those arguments and feeding the opponent modeling with exchanged preference statements. Following, a human-agent negotiation experiment is designed and conducted to evaluate the performance of the underlying negotiation strategy.

The rest of this thesis is organized as follows: Chapter 2 briefly reviews related work with a discussion and Chapter 3 provides the necessary background for agent design and classification of arguments. The novel argument-type taxonomy is introduced in Chapter 4. The developed human-agent negotiation framework and collected corpus are explained in Chapter 5, while the proposed approach is explained in Chapter 6. Furthermore, the evaluation of the proposed agent strategy is reported in Chapter 7. Finally, Chapter 8 concludes this thesis with the future work.

CHAPTER II

RELATED WORK

One of the challenges in human-agent negotiation is preprocessing statements in a given natural language, such as English, and adapting behavior accordingly. Recent studies in this field focus on how to deal with negotiating in natural language. This chapter provides an overview of these studies.

Mell *et al.* present a human-agent negotiation framework where a human negotiator can interact with an automated negotiating agent via a chat-based interface [23]. In their framework, human negotiators can make offers and send predefined arguments during their negotiation. Moreover, they can express their emotional states by exchanging emojis. It is worth noting that the framework is also used for human-agent negotiation competition [25] to motivate researchers to design effective negotiating strategies. However, this framework does not support free-speech exchanges between an agent and a human negotiator. The human negotiator can choose sentences from available predefined templates. It would be necessary to emphasize that this work does not mainly focus on dealing with the difficulties of using natural languages in negotiations.

Unlike template-based sentences, processing free-speech texts requires more sophisticated methods, such as machine learning. It is proven that deep learning techniques provide effective solutions to handle this kind of Natural Language Processing (NLP) task [26, 27, 28]. For instance, Bakker *et al.* propose a deep learning approach that can generate a consensus opinion based on the participants' statements about a debate [19]. Firstly, the model collects the opinion of each participant on a debatable question. For this purpose, they trained a language model called "*chinchilla*"

by using a large-scale corpus [29]. On top of this model, a few-shot learning technique is adopted to learn how to build consensus statements from the given opinions. To enhance the model, the participants' feedback on the generated statements (i.e., Likert-scale score) is collected to increase the social welfare (i.e., the total score of participants). Apart from this, they develop another "*chinchilla*" language model called the "*Reward Model*" to estimate the total score of the participants. Hence, the "*Reward Model*" enables the selection of the consensus text among the generated candidate texts with the highest estimated score. In contrast to their approach in which the agent plays like a mediator to find a consensus among human participants, our work utilizes human arguments to generate convincing offers in a bilateral negotiation.

Furthermore, Lewis *et al.* introduce an end-to-end approach called DealOrNoDeal [16] where the agent receives input sentences in English from human negotiators and produces a text as a response during the negotiation. They first built a corpus involving text message exchanges in human-human negotiations, which is available on the Internet[1]. Afterward, they apply a deep Reinforcement Learning (RL) approach to train an end-to-end model producing the text to be sent to the human negotiation as a response. The reward mechanism is based on the received utility of the outcome and whether or not an agreement is reached. In contrast to that work, our aim is not to generate offers in text based on a black box machine learning approach; instead, our ultimate aim is to design opponent modeling and bidding strategies benefiting from analysis of the human arguments.

Besides, the RL model may generate poor, short, and insufficient sentences compared to human arguments. He *et al.* propose a novel approach splitting the end-to-end structure into two models [17]. The first model takes the opponent's text

[1]The human-human negotiation corpus can be found in the following github repository: https://github.com/facebookresearch/end-to-end-negotiator.

and negotiation history as input and determines an action to take among a set of predefined actions (e.g., greeting, asking a question, offering, accepting) by utilizing an RL approach. The second model takes the chosen action by the first model and selects the most appropriate template sentence from its corpus (CragList), which can fit the best with the chosen action. This sentence template is adapted to the current action. Note that the corpus with the corresponding action labels is available on the Internet[2]. Consequently, their approach can generate more complex sentences during the negotiation.

Furthermore, Yang *et al.* revise the Craiglist model by taking the personal behavior of the opponent into account [30]. They analyze the negotiation behavior of each participant and incorporate this analysis into their decision model, determining the action to be taken by the agent. They improve the reward function as well. Moreover, Zhou *et al.* present a decision support system utilizing the same corpus [31]. To sum up, the designers must have complete control over the negotiation strategy developed in the RL-based NLP studies. On the other hand, we focus on manually designing strategies benefiting from argument exchanges in human-agent negotiations instead of relying on a black box strategy.

Chawla *et al.* point out the significance of understanding human arguments during negotiation [18]. They built a new negotiation corpus called CaSiNo from human-human negotiations and defined relevant argument types. In that work, the exchanged text can be associated with multiple labels. It is worth noting that the negotiation addressed in that work is a resource allocation problem. Similarly, they introduce a classifier to extract corresponding argument types from the given text. They adopted BERT-based models [26], whereas we rely on classical machine learning techniques with relatively low computational requirements. Note that the text in the CaSiNo

[2]The human-human negotiation corpus can be found on the following website: https://huggingface.co/datasets/craigslist_bargains

dataset also involves emojis. They analyzed the CaSiNo dataset by considering the personal information, negotiation behavior, and emojis that appeared in the sentences received from each participant to extract their emotional state [32]. Accordingly, they propose an opponent model to predict the human negotiator's emotional state [33]. Note that the CaSiNo dataset is not fully available on the Internet. Following that study, Hoegen *et al.* build a novel negotiation corpus called *DyNego-WOZ*, which also contains the facial expression of the human participants [15]. They aim to recognize the emotional state along with the underlying reason, not only from the types of arguments extracted from the text but also from the participant's facial expression. It is worth noting that they utilize the argument types defined in the CaSiNo corpus.

Like our work, Rosenfeld *et al.* support natural language processing and allow researchers to design agent strategies for human-agent negotiation [22]. Their dialog manager consists of three components: natural language understanding (NLU), autonomous negotiating agent, and natural language generating (NLG). The NLU component extracts the structured offer content from the received text. The autonomous negotiating agent applies its strategy by considering the given content by the NLU component to determine its next offer. This offer is converted to a corresponding text by the NLG component. Their main focus was dealing with partial offers instead of analyzing human arguments expressing their constraints, preferences, and feedback.

As we mentioned, we want to analyze arguments exchanged between human negotiators. Güngör *et al.* have already presented a hierarchy of arguments (e.g., rewarding, threat, explanations) [24] based on existing literature [34, 35, 36]. We enriched this hierarchy by introducing fine-grained categories. Although that framework enables human negotiators to express their emotions and arguments freely, it does not analyze the arguments as we did in this work. We labeled our corpus with the enriched set of categories, and some of those arguments are used to strengthen

the opponent modeling and bidding. To summarize, we listed the comparison of text-based human negotiation studies in Table 1.

Table 1: Comparison Matrix for Text-Based Human-Agent Negotiation Studies

	Data Labeling	**Value Based Labeling**	**Strategy**	**Emotional Statement**
Güngör[24]	No	No	No	Yes
NegoChat[22]	No	Yes	Partial Offering	No
DealOrNoDeal[16]	No	No	End-to-End	No
CraigList[17, 30, 31]	Action Types	Only Offer	Action-based	No
CaSiNo[18, 33]	Argument Types	No	Opponent Model	Emojis
DyNego-WOZ[15]	Argument Types	No	Opponent Model	Facial Expressions
Our Approach	Argument Types	Yes	Opponent Model & Bidding Strategy	Emojis with rates

In the given table, the first column denotes whether the corpus includes arguments and the second column indicates whether the content of the offers is labeled in the corpus. Similarly, the last column presents whether emotional statement labeling is performed in their corpus. The third column is about the strategy proposed in the corresponding work. As seen in Table 1, only a few studies aim to apply a data-driven approach to process arguments in human-agent negotiations. Moreover, issue-value labeling is rarely done (e.g., offer content as well as explanation using issue values). While some of the studies, such as [18, 33, 15] use their corpus to model the opponent during the negotiation (e.g., predicting their emotional state from a given sentence). Some studies, like [16], focus on learning how to generate offer texts from given counter-offers. In contrast, we use arguments in both opponent modeling and manually-designed bidding strategy benefiting from the analysis of some arguments.

To sum up, an argument-based agent with a rich argument-type taxonomy is required for human-agent negotiations. Agents can exploit a corpus involving issue-value and emotional state labeling. Accordingly, this thesis aims to meet the needs mentioned above as explained in Chapter 6.

CHAPTER III

BACKGROUND

This chapter explains the necessary background information for agent-based negotiation (Section 3.1). Since we apply machine learning techniques to predict the category of the given human arguments, we also briefly explain the classification approaches we used in this thesis (Section 3.2).

3.1 Agent-based Negotiation

In automated negotiation, intelligent software agents negotiate with each other to reach an agreement on their users' behalves. Mostly, they do not disclose their preferences and strategies beforehand. To interact with each other coherently, they need to follow a protocol determining the course of valid actions at a given time and the rules for reaching a consensus. For this purpose, various negotiation protocols governing the interaction among the agents have been proposed [37, 38]. Among them, the *stacked alternative offers protocol* is the most widely used [37]. According to this protocol, one of the parties initiates the negotiation by making the first offer. The opponent can accept the offer, make a counteroffer and end the negotiation without agreement. This process continues in a turn-taking fashion until reaching an agreement or the deadline.

Figure 1 illustrates how a human and an agent negotiate according to this protocol. When they reach an agreement, agents receive the utility of the outcome concerning their private utility functions. Both sides get their reservation utility (i.e., mostly utility of zero) when the deadline is up. In this thesis, the agents are allowed to exchange their emotional states via emojis and arguments via chat in addition to their offers.

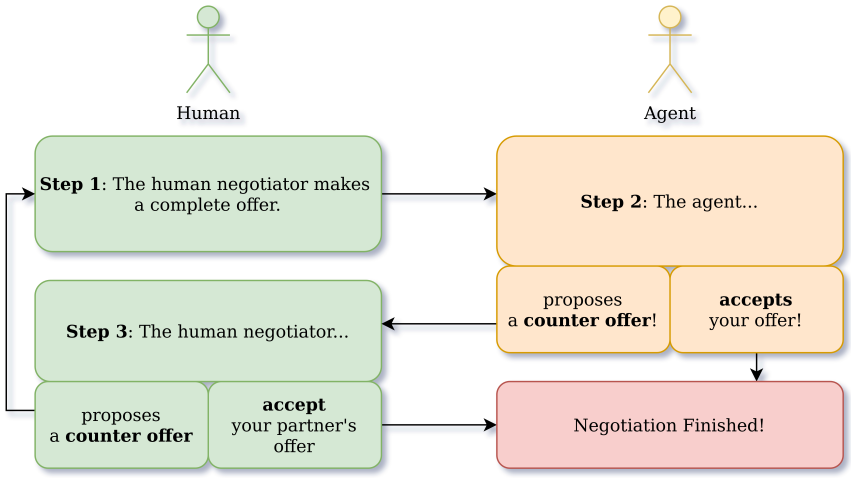


Figure 1: The Alternative Offer Protocol in the Human-Agent Negotiations

The preferences of each party are represented mainly by utilizing an additive utility function. Equation 1 shows the calculation of the utility of an offer, o. The negotiators can infer to what extent an offer is preferred; consequently, they can compare offers during the negotiation. In Equation 1, W_i represents the importance of an issue i, while V_i denotes the value evaluation for the i^{th} issue value. The summation of W_i always equals one, and V_i is between 0 and 1. An autonomous negotiating agent is usually designed based on the utility score of the given offers. It is worth noting that each agent only knows their preferences.

$$U(o) = \sum_i W_i * V_i(o) \tag{1}$$

Designing a sophisticated negotiating agent is a crucial research question of multi-agent negotiation systems. This process requires an advanced framework and design patterns. The main components of an negotiating agent are the bidding strategy, the acceptance strategy, and the opponent model [5]. The bidding strategy decides what offer will be made (Section 3.1.1), the acceptance strategy decides when to accept the opponent's offer (Section 3.1.2), and the opponent model predicts the opponent's preferences and strategy (Section 3.1.3).

3.1.1 Bidding Strategy

A negotiator mainly aims to maximize the received utility of the agreement at the end of the negotiation. Accordingly, it adopts a bidding strategy that determines the target utility at the current round. Various bidding strategies have been proposed so far [39, 12, 6, 7, 8, 9, 10]. This section presents the fundamental bidding strategies adopted in negotiation and the Solver agent strategy that is extended in this study as follows:

- **Time-dependent Bidding Strategy:** The fact that underlying negotiations have deadlines causes time pressure on the negotiators. This pressure makes the negotiators tend to concede more as time passes to reach an agreement. Therefore, they start with the maximum desired target utility (1.0) and go down to the minimum desired target utility ($\geq RU$), where RU is the reservation utility – a minimum utility that can be acceptable for the agent. The speed of the concession may vary concerning the remaining time. Faratin *et al.* propose three time-dependent bidding behaviors: Boulware, Linear, and Conceder [7] where Figure 2 shows the target utility calculation function of those behaviors separately.

  The fact that underlying negotiations have deadlines causes time pressure on the negotiators. This pressure makes the negotiators tend to concede more as time passes to reach an agreement. Therefore, they start with the maximum desired target utility (1.0) and go down to the minimum desired target utility ($\geq RU$), where RU is the reservation utility – a minimum utility that can be acceptable for the agent. The speed of the concession may vary concerning the remaining time. Faratin *et al.* propose three time-dependent bidding behaviors: Boulware, Linear, and Conceder [7] where Figure 2 shows the target utility calculation function of those behaviors separately.

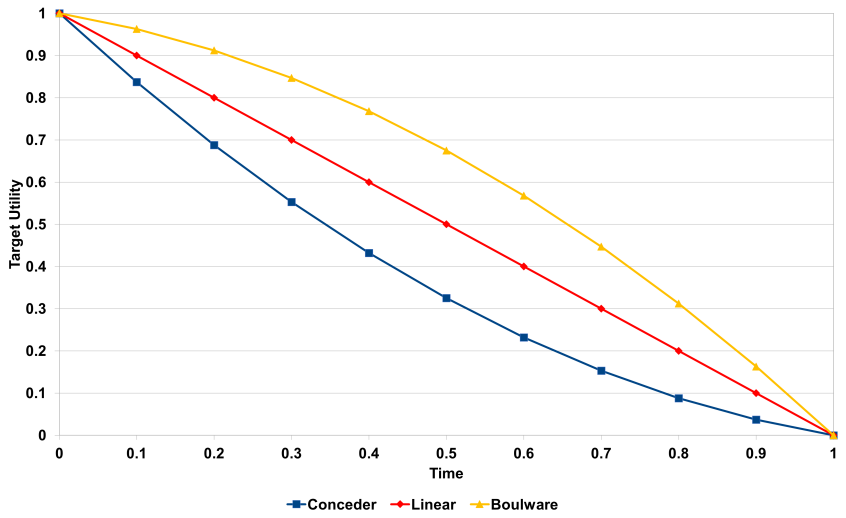


Figure 2: The Target Utility Over Time of the Time-Based Behaviors

Moreover, Vahidov *et al.* present a concession strategy for human-agent negotiation that adopts the aforementioned time-based concession idea [40]. Equation 2 shows the calculation of The target utility (TU) where P_0 is the maximum target utility at the beginning of the negotiation, P_2 is the minimum target utility at the end of the negotiation. Here, P_1 determines the concession speed. Equation 3 shows what kind of behavior the agent adopts depending on the values of those parameters.

$$TU = (1-t)^2 * P_0 + (1-t) * t * P_1 + t^2 * P_2 \tag{2}$$

$$\text{Behavior} = \begin{cases} \text{Boulware}, & \text{if } P_1 > \frac{P_0+P_2}{2} \\ \text{Linear}, & \text{if } P_1 = \frac{P_0+P_2}{2} \\ \text{Conceder}, & \text{if } P_1 < \frac{P_0+P_2}{2} \end{cases} \tag{3}$$

- **Behavior-dependent Bidding Strategy:** An autonomous negotiating agent can adapt its behavior concerning the opponent's behavior [39, 9]. For this adaptation, the opponent's behavior must be analyzed. Hindriks *et al.* define

move categories by comparing the utility change of both parties [1]. For a negotiating party, the previous and current offered bids are examined to compare the both sides' utility. Table 2 states the move types with the corresponding self (Δ_S) and opponent (Δ_O) utility changes.

Table 2: The Move Classes with the Corresponding Utility Change [1]

Move Type	**Self**	**Opponent**
Fortunate	$\Delta_S > 0$	$\Delta_O > 0$
Selfish	$\Delta_S > 0$	$\Delta_O \leq 0$
Concession	$\Delta_S < 0$	$\Delta_O \geq 0$
Unfortunate	$\Delta_S \leq 0$	$\Delta_O < 0$
Nice	$\Delta_S = 0$	$\Delta_O > 0$
Silent	$\Delta_S = 0$	$\Delta_O = 0$

These types help classify the opponent's behavior. The percentage of the move types could be analyzed to determine the opponent's general attitude. Moreover, one of the analyses is the sensitivity to the opponent's behavior [1]. The behavior sensitivity (*Sensitivity*) can be calculated by dividing the positive moves (i.e., fortunate, concession, nice) by the negative moves (i.e. selfish, unfortunate, silent). Equation 4 shows how to calculate the behavior sensitivity.

$$Sensitivity = \frac{\%_{Fortunate} + \%_{Concession} + \%_{Nice}}{\%_{Selfish} + \%_{Unfortunate} + \%_{Silent}} \tag{4}$$

This analysis indicates to what extent the opponent is collaborative. The opponent can be considered collaborative if the *Sensitivity* is positive. Otherwise, the opponent can be considered competitive. Most behavior-based agents adapt their behavior when the opponent changes its move direction. In addition to this analysis, the opponent's behavior awareness (*Awareness*) can be examined. If the agent changes its move from positive to negative, or vice versa, a behavior-based opponent accordingly changes the direction. For the calculation, the number of the opponent's move direction changes when the agent

changes the move direction is divided by the number of the agent's direction changes. Equation 5 demonstrates the calculation of *Awareness*, and its range is between 0 and 1. The higher *Awareness* means that the opponent is aware of the agent's behavior. For these analyses, the opponent's preferences have to be fully known; however, the agents do not access their opponent's preferences. Therefore, the agent should estimate the opponent's preferences via the opponent models by analyzing the bid exchanges. This fact indicates the importance of the opponent model since a better opponent model brings more accurate predictions for the analysis.

$$Awareness = \frac{\text{\# of the opponent's move direction changes accordingly}}{\text{\# of the agent's move direction changes}} \quad (5)$$

With an opponent model, a behavior-based agent can be implemented. Mimicking the opponent's behavior is the most common strategy used in behavior-based autonomous agent design [39, 13]. According to the behavior-based strategy presented in [13], the agent initially sets a high target utility and updates it over time based on its opponent's moves (i.e., concession, silent, and selfish moves). In other words, the agent determines the current target utility ($TU_{current}$) by considering its utility change in its opponent's consecutive offers (ΔU). Hence, the agent concedes if the opponent concedes regarding the received utility. Equation 6 denotes the sign of the ΔU with the corresponding move that the agent makes. Besides, the mimicking amount is also adjusted depending on the current time and a parameter (P_3) regulating the magnitude of the target utility update. Equation 7 shows the target utility calculation. It is worth noting that the magnitude of the updates is smaller at the beginning

of the negotiation but increases toward the deadline.

$$Move = \begin{cases} \text{Concession}, & \text{if } \Delta U < 0 \\ \text{Silent}, & \text{if } \Delta U = 0 \\ \text{Selfish}, & \text{if } \Delta U > 0 \end{cases} \tag{6}$$

$$TU_{current} = U(O_j^{t-1}) - \mu \times \Delta U \tag{7}$$

$$\mu = P_3 + P_3 * t$$

However, comparing the last two subsequent offers may not capture the real intention or behavior of the opponent. The opponent may manipulate the utility changes and use them in her/his favor. Therefore, Keskin *et al.* introduce a window mechanism for target utility calculation [9]. They calculate the agent utility change in a window involving a certain number of subsequent offers as shown in Equation 8. Here, the utility changes are multiplied by their weights (W_i) – the more recent pairs have higher weights.

$$\Delta U = \sum_{i=1}^{4} [W_i \times (U(O_h^{t-i}) - U(O_h^{t-i-1}))] \tag{8}$$

- **Hybrid Agent:** The hybrid strategy determines the target utility by combining time and behavior-based strategies [9]. Hybrid Agent calculates a time-based target utility (TU_{time}) and a windowed behavior-based target utility ($TU_{behavior}$) as mentioned above and combines them as shown in Equation 9. At the beginning of the negotiation, $TU_{behavior}$ has a higher weight, whereas the weight of the time-based target utility calculation increases when the deadline is approached.

$$TU_{hybrid} = t^2 * TU_{time} + (1 - t^2) * TU_{behavior} \tag{9}$$

- **Solver Agent:** A human negotiator uses various factors to achieve an agreement during a negotiation. A good instance of these factors is the emotional state of the opponent. Keskin *et al.* extend the hybrid negotiation strategy by incorporating the opponent's emotional feedback into bidding behavior as shown in Equation 10 where P_E and P_A denote the emotion coefficient, and awareness constant, respectively.

$$TU_{Behavior} = (1 - P_A) * [U(O_j^{t-1}) - (\mu \times \Delta U)] + (P_A * P_E) \tag{10}$$

The calculation of P_A is defined in Equation 5. P_A is decided as a weight between the emotion-based and the windowed behavior-based strategies [1]. Human negotiators with higher awareness tend to express emotional state. Accordingly, the weight of the emotion-based target utility increases.

One of the challenges is estimating the emotion coefficient capturing the opponent's emotional feedback during the negotiation. There are various methods to extract the emotional state of a human, such as sentiment analysis, facial expressions, emojis, and so on. Instead of assigning primary emotional feedback (e.g., sad, happy), a vector of different emotions is considered. In contrast, a percentage per each relevant emotion in negotiation is considered in this thesis. The calculation of P_E is described in Chapter 6. Consequently, how emotion-based bidding strategy decides which move to take is shown in Equation 11.

$$Move = \begin{cases} \text{Concession}, & \text{if } P_E < 0 \\ \text{Silent}, & \text{if } P_E = 0 \\ \text{Selfish}, & \text{if } P_E > 0 \end{cases} \tag{11}$$

[1] Opponent awareness could be estimated during the negotiation. Since it requires some minimum number of interactions, we take a constant value for this parameter in this thesis.

3.1.2 Acceptance Strategy

A negotiation lasts until reaching an agreement or the deadline. Both sides get a zero utility score or the reservation value if available when the deadline is up. Since a negotiator aims to maximize the utility score, determining when to accept the opponent's offer is important. Hence, several acceptance strategies exist in the literature [11]. They can be a time based strategy (AC_{time}), a utility based strategy (AC_{const}), or the combination of them (AC_{combi}). The most common and successful strategy is the next utility acceptance strategy (AC_{next}). According to this strategy, the agent accepts its opponent's offer when the received utility of the opponent's current offer is equal to or greater than the agent's target utility estimated for the current round. The demonstration of the AC_{next} is shown in Equation 12 where TU is the target utility defined by the bidding strategy and $U(o)$ is the received utility of the counter offer of the opponent. If the AC_{next} is *"accept"*, the agent will accept the opponent's offer. Otherwise, the agent proposes a counter-offer. Note that the performance of the AC_{next} strategy depends on the performance of the bidding strategy. Besides, the proposed agent of this thesis also implements the same acceptance strategy.

$$AC_{next} = \begin{cases} accept, & \text{if } TU \leq U(o) \\ reject, & \text{if } TU > U(o) \end{cases} \tag{12}$$

3.1.3 Opponent Model

The opponent modeling in this thesis mainly aims to estimate the opponent's utility function by analyzing the bid exchanges during the negotiation. In the literature, it is mostly assumed that agents concede over time and have additive utility functions to capture their preferences. By comparing the values in subsequent offers made by the opponent, the models try to guess which issues are more critical and which values

are preferred.

The most widely used opponent modeling approaches are the frequency-based opponent models [41, 2, 12]. In the basic frequency-based approaches [41], agents have two heuristics: (i) the opponent concedes less on important issues, and (ii) the preferred values appear more often in the opponent's offers (i.e., a negotiator tends to offer the most desired values). If a given value for an issue appears more frequently, the evaluation value of that value increases. As seen in Equation 13, the evaluation value for issue values is increased by one when they appear in the opponent's subsequent offer again. We scaled those updated values by dividing the maximum evaluation value so that the range will stay between zero and one, as shown in Equation 14. Note that they are assumed to be equally essential and preferred at the beginning.

$$\hat{V}_i^j = \hat{V}_i^j + 1 \tag{13}$$

$$\hat{V}_i^j = \frac{\hat{V}_i^j}{\max(\hat{V}_i)} \tag{14}$$

If the value of an issue remains the same in the subsequent offer, its importance value ($\hat{W}_i$) is increased by a coefficient (n) as seen in Equation 15. A negotiator tends to concede on the least significant issues. However, the agent may need to concede even on the most significant issues due to the time pressure. Hence, the remaining time is taken into consideration in the basic frequency-based approach of [41] [2]. Moreover, the summation of the issue weights always has to equal 1. Therefore, the issue weights should also be normalized. The issue score ($\hat{W}_i$) updates are shown in Equation 16.

$$\hat{W}_i = \hat{W}_i + n \tag{15}$$

[2]The coefficient of the issue update depends on negotiation time ($n = 1 - t$).

$$\hat{W}_i = \frac{\hat{W}_i}{\sum_i \hat{W}_i} \tag{16}$$

Tunalı *et al.* extend the frequency-based approach by introducing the window approach [2]. Instead of comparing two subsequent offers, they compare the fixed number of subsequent offers (i.e., window). They test whether there is a statistical difference between the distribution of issue values in a current window to determine whether the issue weights should be updated. For updating issue weights, α and β parameters are used to adjust the amount of update, as seen in Equation 17. The issue update approach works well if the opponent makes a concession. However, opponents can make other moves, such as silent and selfish moves. Therefore, Tunalı *et al.* apply the issue update if there is a concession between the windows. Note that these windows are consecutive and disjoint windows of the negotiation history of the opponent. Figure 3 demonstrates the consecutive windows, and every window has k offers received from the opponent. The normalization of the value weight is similar to the basic frequency-based approach, but this approach uses a smoothing factor (γ) to make value utility estimation more stable. The value weight normalization is shown in Equation 18.

$$\hat{W}_i = \hat{W}_i + \alpha * (1-t)^{\beta} \tag{17}$$

$$\hat{V}_i^j = (\frac{\hat{V}_i^j}{max(\hat{V}_i)})^{\gamma} \tag{18}$$

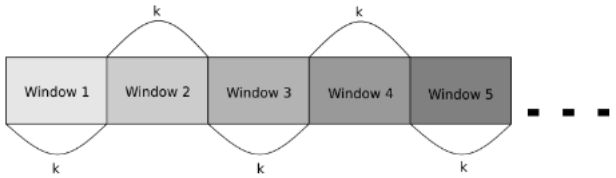


Figure 3: The Demonstration of the Consecutive Windows [2]

3.2 Text Classification

Natural Language Processing (NLP) is the subfield of Artificial Intelligence that aims to process and analyze textual data in natural language and extract information and insights from a given text [42]. Since this thesis focuses on chat-based human-agent negotiation, it is necessary to process the human arguments to extract the offer content and explanations. Mainly, we aim to classify the given arguments. With this purpose, we apply several machine learning algorithms which are briefly explained in Section 3.2.1. Section 3.2.2 presents how those techniques could be applied to text classification tasks.

3.2.1 Classification Methods

ML algorithms aim to learn from the provided data by relying on statistics and information theory. Depending on the problem, the available methods may vary. Basically, ML algorithms can be categorized as Supervised Learning, Unsupervised Learning, and Reinforcement Learning based on the task [43]. Supervised Learning (SL) aims to find a model which can take input data and map it to a target value by processing labeled data (i.e., training data). In other words, the label data indicates possible inputs and the desired corresponding target values. Accordingly, we try to find a model capturing the relationship between the inputs and the outputs. A model can be seen as a collection of parameters. Thus, the most convenient parameter values are determined by exploring the given training set. Afterward, the model can be used for the unseen data to predict their outputs. When the output is a categorical or discrete data type, this problem is called classification. Classifiers are the algorithms for finding models for this type of problem where they produce a class probability of a given instance and accordingly predict to which class the given instance belongs. The classification algorithms that we applied in this thesis are listed below.

- **Logistic Regression**:This method applies the logistic (i.e., sigmoid) activation

function to calculate the probability of the corresponding class [44]. The sigmoid activation function takes the weighted sum of the inputs and produces a value between zero and one. This value corresponds to a probability value of the given instance belonging to that class. Given the threshold (e.g., 0.5), the algorithm determines whether the instance will be assigned to that class. Even if it is a binary classifier, it can be used for multi-class problems with more than two class labels by applying softmax activation function instead of sigmoid activation function. During the training, the weight parameters are estimated. The model is shown in Equation 19, where $\hat{y}_i$ equals to the corresponding class probability.

$$\hat{y}_i = \sigma(w_i^T \times X_i + b_i) \tag{19}$$
$$\sigma(z)_i = \frac{\exp^{z_i}}{\sum_j \exp^{z_i}}$$

- **Multi Layer Perceptron (MLP)**: MLP [45] is a neural network consisting of input, hidden, and output layers, as seen in Figure 4. Each input feature is associated with an input neuron that is fully connected with all neurons in the hidden layer. The nodes in the hidden layers are fully connected with the output node. Note that each connection in the neural network is associated with a weight parameter. In the hidden layer, each node applies an activation function, such as ReLU on the weighted sum of the input values. Similarly, the weighted sum of those values is given to the output node producing a probability of class labels (i.e. softmax activation function). The given instance is assigned to a class label depending on those probabilities. During the training, weight values are estimated by using a backpropagation algorithm.

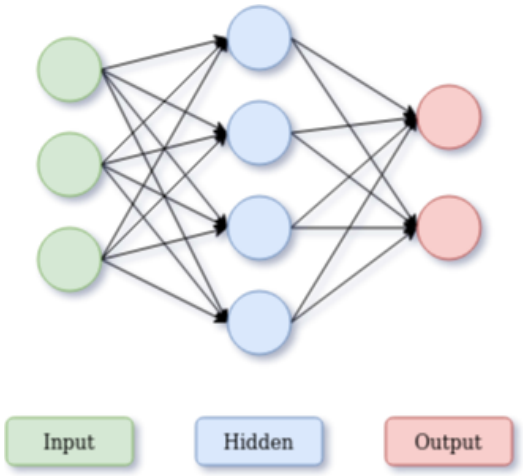


Figure 4: Demonstration of a Simple MLP

- **K-Nearest Neighbour (KNN)**:KNN is a non-parametric classifier predicting the class of a given instance by exploring the training set [46]. It keeps all training instances for prediction. It finds the most similar k instances (i.e., neighbors) of a given instance and makes a corresponding prediction based on their class labels. For calculating the similarities between instances, the Euclidean distance is usually used.

- **Naive Bayes**: This approach relies on Bayesian Theory. Given the data distribution, it calculates the posterior probabilities (i.e., the probability of belonging to a particular class given the input values) [47].

- **Decision Tree**: The algorithm tries to find the decision boundaries partitioning the space into disjoint decision regions [48]. Those boundaries correspond to a rule that could be extracted from a decision tree. The algorithm generates a decision tree from the given training instances so that the non-leaf nodes include the splitting conditions (e.g., condition on a chosen feature) and the leaf nodes capture the class labels of that partition. Selection of the best split relies on impurity metrics such as entropy and the gini index.

- **Random Forest**: Random Forest is an ensemble learning method producing more than one random decision tree and making predictions based on their overall predictions [49]. To achieve this, it randomly picks some training samples with replacements (i.e., bootstrapping) and produces a decision tree using different feature sets. In other words, different parts of the training instances are used to generate each tree. The class decision of a given sample is made by using a plurality voting on the predictions of those trees as illustrated in Figure 5.

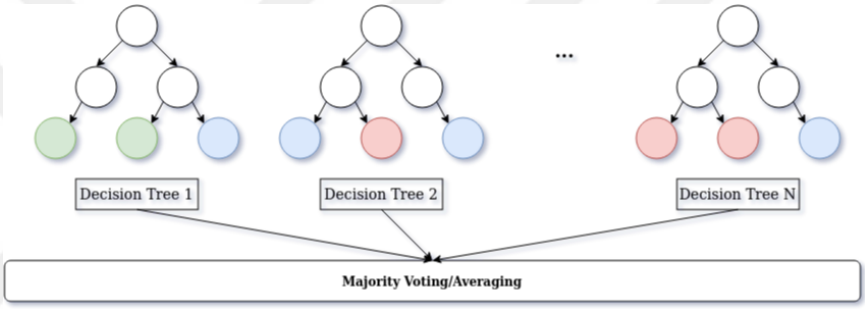


Figure 5: Demonstration of a Sample Random Forest

- **Long-Short Term Memory (LSTM)**: LSTM, a Recurrent Neural Network model, is mainly used for sequential and time-series inputs [50]. Iteratively, it calculates the output of each item in a given sequential data by examining the output of the item at the previous time steps. The output of the model is the output of the last item in the given sequence. An MLP takes the output of the LSTM model to find the probability of each class. Besides, LSTM has a gate mechanism to control the information flow between the time steps. The gate mechanism can also avoid the Vanishing Gradient problem. Additionally, it has lots of variations to solve various tasks [27, 51]. Figure 6 demonstrates a LSTM cell where the processing of an iteration takes place.

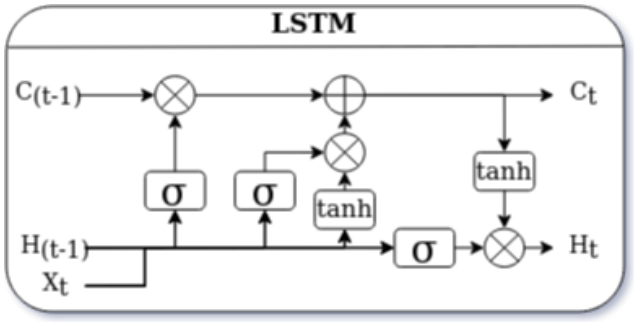


Figure 6: Demonstration of a Simple LSTM Cell

Additionally, the classification decision of several classifiers can be combined for more accurate predictions. This ensembling approach is called "Mixture of Experts" in the literature [52]. There are several combination methods for the mixture of experts method. In this thesis, "Averaging", and "Weighted Averaging" were applied. *Averaging* takes the average of class probabilities estimated by the chosen classifiers, whereas *Weighted Averaging* calculates the weighted average of the probabilities where the weights are set with the performance of the provided algorithms.

While some of the data is used for training the model, the remaining part (i.e., the test set) is reserved for evaluating the classifiers. The most common and reliable data splitting method is k-Fold Cross-Validation [43]. This approach partitions the given dataset into k folds, and iterates over each fold to generate training and test sets. In an iteration, one of the pieces is chosen as the test set while the remaining ones are taken as the training set. Figure 7 demonstrates the k-Fold cross-validation approach where $k = 10$. In Figure 7, the green ones are the test sets, while the remaining ones are the training sets.

3.2.2 Text Classification with Machine Learning

We can apply machine learning algorithms for processing a given text in natural language as their success has been proven in many studies [53, 51, 27, 26, 54]. However, machine learning algorithms require a numeric representation of the inputs. In other

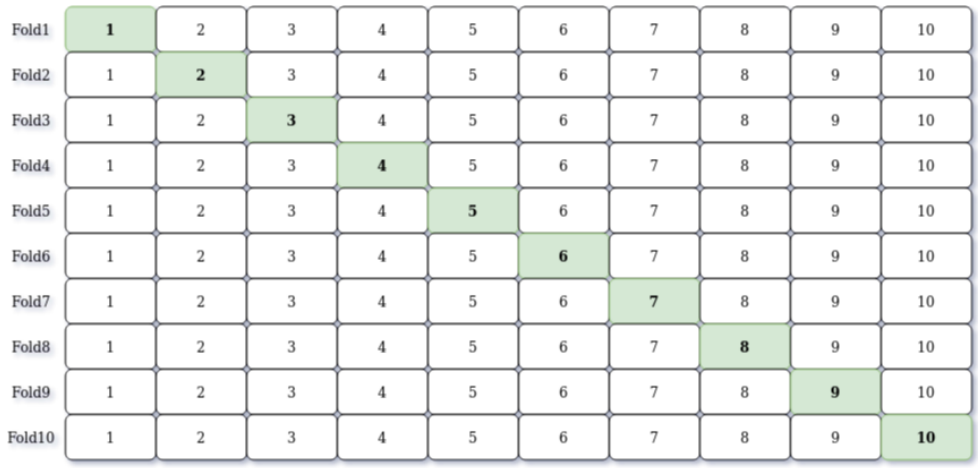


Figure 7: 10-Fold Cross-Validation Demonstration

words, the given text needs be transformed into a numeric form. The Bag of words method (BOW) is used to convert a text into a vector based on the occurrence of words in the given text [55]. In this thesis, we adopted a simple approach and converted the given text into a binary vector where each binary value denotes the presence or absence of words. BOW generates a vector where each element represents a unique token in the corpus. If the given text contains the corresponding word, the element is set to one; otherwise, zero. An example demonstration of the BOW approach is shown in Figure 8.

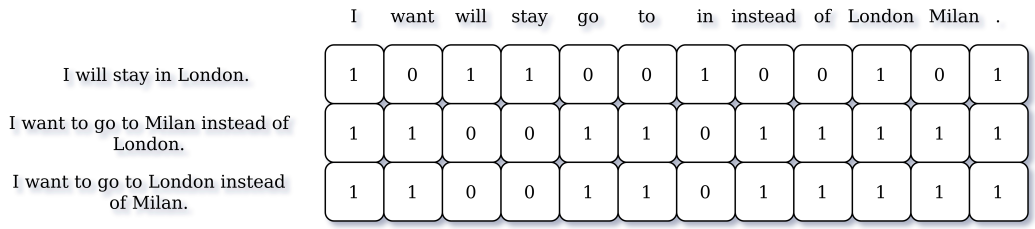


Figure 8: Example Demonstration of Bag-Of-Words

Furthermore, there are some variations of BOW methods such as TF-IDF and Polarity [56, 57]. However, the order information of the tokens is lost with the aforementioned approaches. In other words, the relationship between tokens is not captured. On the other hand, the relationship between tokens may play a vital role in the

accuracy of the predictors. Therefore, some techniques, such as the NGram method and its variations, can be adopted to use the BOW effectively. The NGram method finds the corresponding tokens in the corpus by adopting a window-based approach. Hence, the relationship with tokens can be utilized with this representation approach if they appear in the same window. Depending on the size of the window, the NGram method has different variants such as *"unigram"*, *"bigram"*, *"trigram"*, and so on. For example, the sample demonstration of the BOW approach in Figure 8 is also a BOW method with *"unigram"*. Figure 9 exhibits a *"bigram"* version of the previous BOW example. Each element of a *"unigram"* vector represents only one token, whereas that of a *"bigram"* denotes a couple of tokens. Besides, the BOW vector of the texts diverges each other in the *"bigram"* example more than the *"unigram"* one. In our thesis, we applied the BOW method with a combination of *"unigram"*, *"bigram"*, and *"trigram"* approaches to our corpus.

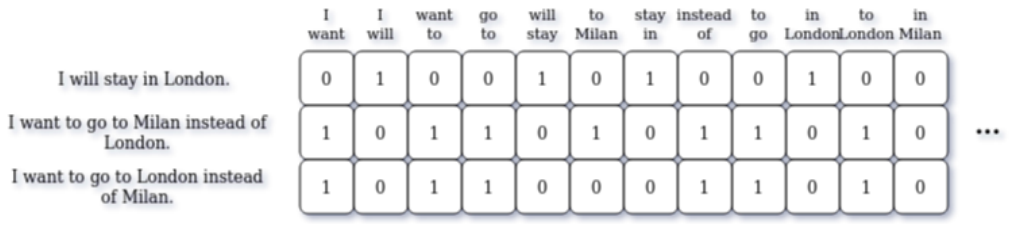


Figure 9: Example Demonstration of BOW with BiGram

CHAPTER IV

PROPOSED ARGUMENT TAXONOMY

In human-agent negotiation, different types of arguments could be exchanged to convince the opponent or give feedback so that the opponent can generate better offers that can benefit both sides. In literature, some argument taxonomies [36, 35, 34, 24] have already been introduced. We extend those taxonomies to capture more fine-grained argument types as depicted in Figure 10.

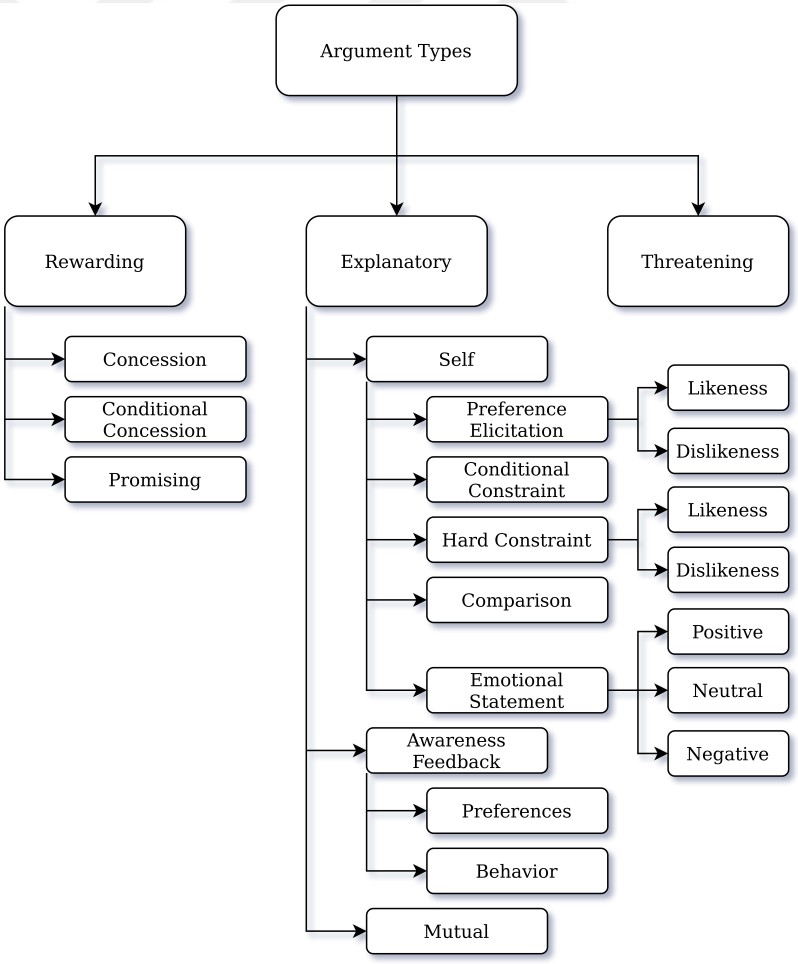


Figure 10: The Proposed Argument Type Taxonomy

As seen in Figure 10, some argument types contain sub-argument types. These sub-argument types and some example sentences are listed below:

- **Rewarding**
  - **Concession Statement**: Expressing the negotiator making a concession.
    "Let's go to Barcelona as you wish."
  - **Conditional Concession Statement**: Asking for something in exchange for concessions on another issue.
    "If we go to Barcelona as you wish, I want to stay in a tent in summer."
    "If we are going to Barcelona, you can choose the accommodation option."
  - **Promise Statement**: Promising something for future negotiations.
    "If you accept my offer, I promise that our next vacation will be as you wish."
- **Explanatory**
  - **Self**: Explaining about itself.
    * **Preference Statement (i.e., soft constraints)**: Stating preferences (e.g., likeness or dislikeness).
      "I do not want to go to London."
      "London is the best option for me."
      "Location is not important for me."
    * **Conditional Statement (i.e, conditional constraints)**: Stating preferences/constraints depend on two or more issues.
      "If we are going to Barcelona, I prefer summer."
      "Staying in a tent in winter would be cold, I cannot accept."
    * **Hard Constraint**: Indicating something desired or undesired.
      "Destination is the most important issue for me, it should be London."

"I have no free time in winter. Winter is unacceptable for me."

* **Comparison**: Comparing two items according to preferences.

  "I prefer Tokyo rather than Milan."

  "Location is more important than the transportation for me."

* **Emotional Statement**: Expressing emotional state (i.e., positive or negative)

  "I feel happier."

  "I am tired of negotiating with you."

– **Awareness Feedback**: Expressing of being aware of the opponent's preferences or behavior.

* **Preferences**: Being aware of the opponent's preferences.

  "You do not want to go to Istanbul."

  "You prefer Istanbul rather than London."

* **Behaviour**: Being aware of the opponent's behavior.

  "You are insisting on Istanbul."

  "Why are you so selfish?"

  "Thank you for your concession."

– **Mutual**: Stating some explanation indicating why the offer is good for both side.

"Going to Istanbul makes us happier."

"Taking a taxi will be expensive for us."

"We both love Barcelona."

- **Threatening**: Threatening or warning the opponent.

"We cannot agree like that."

"It is your last chance."

"Time is almost up!"

The example sentences above are chosen from the collected data. An automated agent should adapt its strategy respect to the arguments received from the opponent during a human-agent negotiation. The argument types and examples of how to utilize them during a negotiation are listed below:

- **Rewarding Argument Types**:A human negotiator indicates their concession move via the rewarding argument type. This means that the rewarding argument types may help in understanding the opponent's attitude during the negotiation. That is, we can exploit this type of argument to utilize opponent modeling considering the concession moves in its updates (e.g., [58, 12, 2]).

- **Constraint Argument Types**: A negotiator can reach an agreement sooner by taking into account the constraints of the opponent (i.e., Soft, Hard, and Conditional) while generating its offers. A bidding strategy can be enhanced by the constraints extracted from human arguments.

- **Comparison Argument Type**: By processing this type of argument, the importance of the issues and the ranking of the issue values could be updated more accurately in the opponent model.

- **Emotional Statement Argument Types**:Suppose the negotiation strategy takes human negotiators' emotional states. In that case, emotional statement argument types could contribute to the detection of the opponent's emotional state more accurately, thus improving the underlying strategy's performance.

- **Awareness Feedback Argument Types**:Awareness feedback argument types state to what extent a human negotiator is aware of the opponent's behavior and preferences. This awareness information of the opponent could be utilized in a bidding strategy.

CHAPTER V

HUMAN-AGENT NEGOTIATION FRAMEWORK & NEGOTIATION CORPUS

Since the currently available negotiation corpora primarily focus on resource allocation problems and do not cover all argument types introduced in our taxonomy. Therefore, we developed a human-agent negotiation framework to build up our corpus in line with the taxonomy described in Chapter 5. This framework enables human negotiators to make their offer in English and specify it through more structured drop-down boxes. They can express their emotional feedback on relevant emotions such as anger, sadness, neutral, happiness, and excitement elaborately (e.g., specifying their percentages where the summation of these percentages always equals 100%). They can also write arguments to convince the agent.

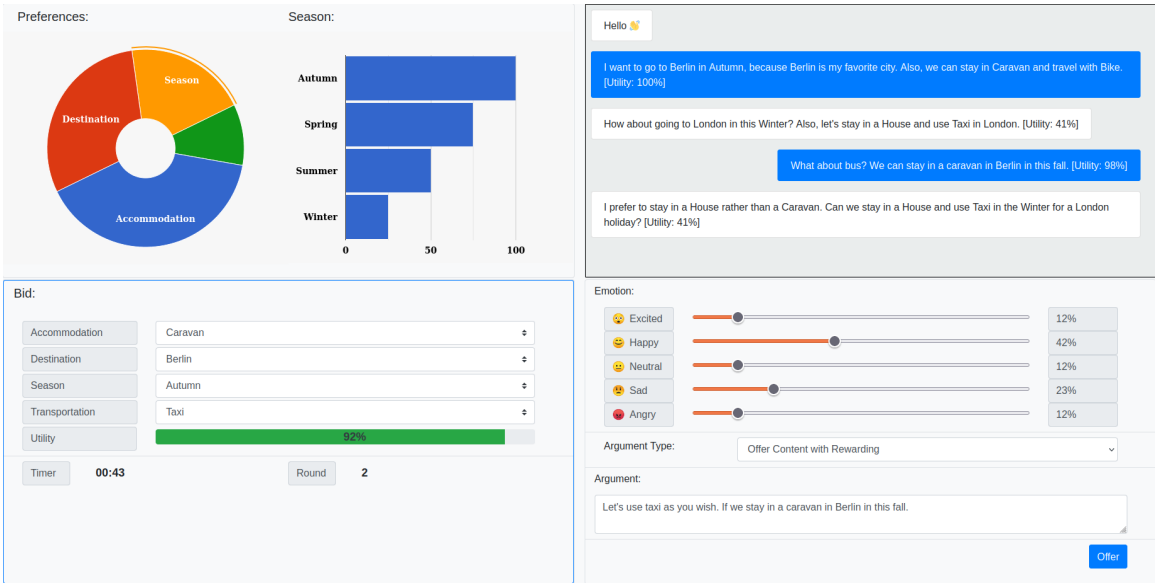


Figure 11: The Data Collection Experiment Interface

As seen in Figure 11, the participants can see their preference profiles in the upper

left-hand corner. The participants can see the chat history with their utility scores in the upper right-hand corner. They can make an offer in the lower left-hand corner by choosing the values of each issue from the given drop-down lists. It is worth noting that the issues and corresponding values are sorted according to the participants' preference profiles to decrease the participants' cognitive load. Here, they can also see the current time, round, and utility of the offer they constructed. Note that to minimize the time pressure on participants so they can provide an enriched set of arguments, we avoid displaying the remaining time; thus, we only show the current time. On the lower right-hand side, the participants can type an offer content and the corresponding arguments as text in English and also express their emotional state via emojis. To accept the opponent's offer, the participant should select the argument type as Acceptance, write an agreement sentence and express the emotional state.

While entering the arguments, the participants need to choose one of the following actions: (i) *"Only Offer Content"*, (ii) *"Offer Content with Explanatory"*, (iii) *"Offer Content with Rewarding"*, (iv) *"Offer Content with Threatening"* and (v) *"Acceptance"* and enter the corresponding text. This categorization simplified the labeling process. The human annotators label each text entered by the participants according to the categories in our taxonomy.

To collect the data, we conducted human experiments where the participants were asked to negotiate twice on the holiday domain (Section 7.2.1) with our negotiating agent employing a different strategy in each session. In each negotiation session, participants have 30 minutes to complete their negotiation. It is worth noting that participants know only their scores, and they were informed that agents do not know their scores too. At the beginning of the experiment, the interaction protocol was explained to the participants via a demo video. Besides, they have a training session where they negotiate a more straightforward problem for 5 minutes to get familiar with the negotiation process. Before starting the experiments, participants signed

the consent form [1].

During their negotiation, the participants communicate with the agent via text, combo boxes, and buttons on the web page. They negotiated on the holiday domain in both sessions, but there were slight differences, such as different destination locations in each setting. In total, 64 participants attended our experiments, resulting in 3013 text exchanges. Afterward, those texts are manually labeled according to the categories in our taxonomy. It is worth noting that some texts have multiple class labels. For instance, some sentences involve both comparisons and rewarding arguments.

[1]The experiment protocol in this study was approved by the Ethics Committee of Özyeğin University on 21 December 2021.

CHAPTER VI

PROPOSED APPROACH

A human negotiator can make arguments to achieve an agreement sooner and achieve a better negotiation outcome. This means that understanding human arguments is crucial for a sophisticated agent. For this purpose, how to extract the human arguments from a given text is explained since the arguments and the offer content are provided in natural language by a human negotiator (Section 6.1). Moreover, this chapter explicates the argument-based opponent modeling (Section 6.3) and the argument-based bidding strategy (Section 6.4). The proposed opponent modeling enriches the windowed frequency opponent model [2], while the proposed bidding strategy is an extended version of Solver Agent [9].

6.1 Classification of Human Opponent's Arguments

In order to process and utilize the meaning of the sentences given in a text, the agent first needs to detect what type of arguments they are. With this purpose, a classifier is built to predict which argument types in the taxonomy correspond to the given text. Recall that human annotators labeled the dataset collected during human-agent negotiations (Chapter 5). According to the proposed taxonomy and dataset, this classification problem can be defined as a multi-label classification (MLC) task and hierarchical classification task, as illustrated in Figure 12.

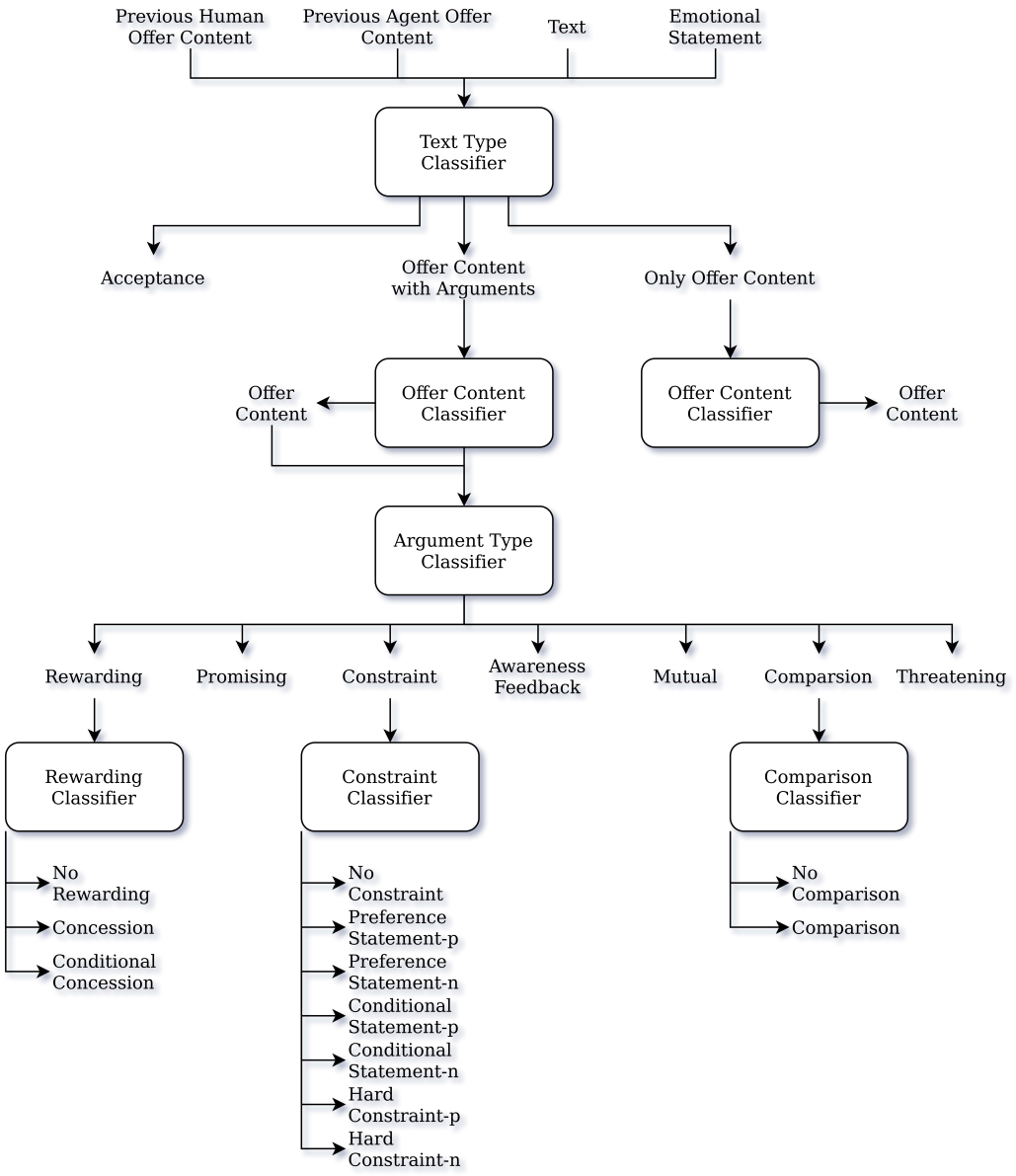


Figure 12: The Proposed Hierarchy of the Classifiers

As seen in Figure 14, the text and emotional statement is given by the human negotiator. However, the text type, the offer content and the arguments must be extracted. In an addition to them, some argument types require estimating the corresponding values to utilize them. However, the size of the collected data is not enough to handle this kind of complex classification task. Besides, the dataset is unbalanced which means that the overfitting problem can be observed. To get better

classification performance and to prevent the overfitting problem, two kinds of data augmentation techniques are applied before training the models. The first one is template-based sentences. With this approach, all combinations of the values in the domain are used to generate new sentences from the template sentences. Secondly, new sentences from the existing ones are generated by changing the values in the sentence. As a result, nearly 51k data instances are obtained.

The proposed approach relies on having multiple classifiers where each classifier aims to learn a particular task. For example, we have a text-type classifier that detects whether the given sentence is an acceptance statement, offers content with arguments, or only involves the offer content. This classifier utilizes the available inputs, such as the underlying text to be classified, the agent's and human opponent's previous offer contents, and the emotional statement expressed by emojis and a scroll bar for their percentages. If the text involves the content of the offer, an offer content classifier is used to detect the issue values per each issue. For example, the classifier predicts the value of the destination from the given text, where the outputs could be "London", "Tokyo", and other possible destinations defined in the negotiation domain. If the text involves arguments, a classifier predicts whether the given argument is a type of rewarding, promising, constraint, awareness feedback, mutual explanation, comparison, or threatening argument. As seen from the classifier hierarchy, there are more fine-grained classifiers if the arguments fit their category. For example, if the argument type is detected as constraint, a constraint classifier aims to detect what type of constraint the argument involves (e.g., preference statement or hard constraints).

Besides predicting predefined categories, rewarding, constraint, and comparison classifiers should also extract the corresponding value. Therefore, these classifiers also take the candidate value as an input and classify it depending on the given inputs and the candidate parameters. In other words, they iterate on each candidate value to extract the proper argument type with the corresponding value. The inputs of

these classifiers are shown in Figure 13. For instance, the same data instance may contain a positive preference statement on the "Istanbul" value. Therefore, the constraint classifier should extract this value-argument pair from the given data instance. While iterating over the whole values in that domain (e.g., "Tokyo", "Milan", "Istanbul"), the classifier should predict "Preference Statement-p" when "Istanbul" is the candidate value. For the other values in that domain (e.g., "Milan" and "Tokyo"), it should output as "NoConstraint".

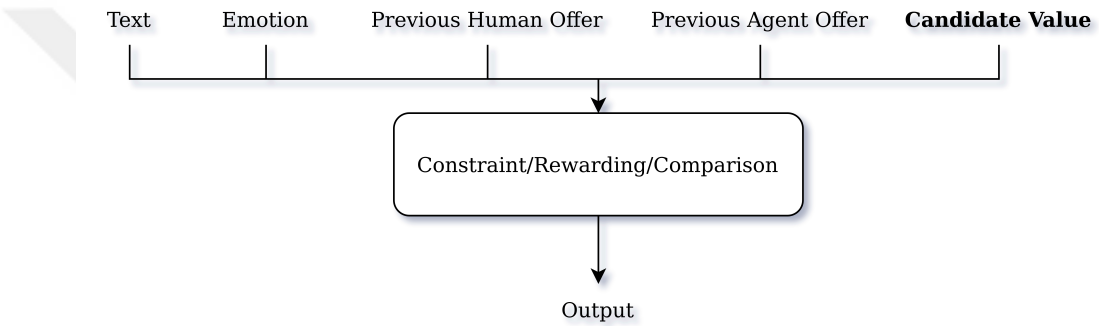


Figure 13: The Inputs of Constraint/Rewarding/Comparison Classifiers

- **Text:** *"I have no free time in Winter. I want to go to London in the Summer and stay in a Hotel. Also, I prefer to use the Subway more than Taxi. This is your last chance!"*
- **Text Type:** Offer Content with arguments.
- **Offer Content:**
  - **Accommodation:** Hotel
  - **Destination:** London
  - **Transportation:** Subway
  - **Season:** Summer
- **Arguments:**
  1. **Label**: Threatening — **Value**: None
  2. **Label**: Hard Constraint Negative — **Value**: Winter
  3. **Label**: Comparison — **Value**: Subway > Taxi
- **Emotional Statement:**
  - **Excited:** 0%
  - **Happy:** 0%
  - **Neutral:** 0%
  - **Sad:** 15%
  - **Angry:** 85%

Figure 14: An Example Text Data in the Corpus

To illustrate the complexity of this classification task, an example text data from the dataset is examined in Figure 14. In this example, the given text involves both

the content of the offer and the arguments. From the given text, we need to extract the content of the offers in a structured way (i.e., extracting the offer in terms of issue-value pairs). We need to also pull out all specific arguments from the text, such as comparison and threat arguments. In this case, the size of the collected data might be insufficient to train such complex models. Furthermore, the collected dataset was unbalanced. Two kinds of data augmentation techniques are applied before training the models to increase the classification prediction performance. The first one is template-based sentences. This approach uses all combinations of the values in the domain to generate new sentences from the template sentences. Secondly, new sentences from the existing ones are generated by changing the values in the sentence. As a result, nearly 51k data instances were obtained.

Classical machine learning approaches for training models such as Naive Bayes, K-nearest Neighbor Classifier, Logistic Regression, Multilayer Perceptron (MLP), decision trees, and Random Forests are adopted on the collected dataset. For the representation of text, the Bag-Of-Words approach is applied with the NGram method ($n \in [1-3]$) [55]. On the other hand, LSTM which is a sequence-based deep learning method is adopted. The evaluation results for each classification task are indicated in Chapter 7.

6.2 Emotional State Analyze via Emojis

A human negotiator can express its emotional state via emojis during a human-agent negotiation. For a more inclusive representation, the negotiator can denote their emotional states with the percentages of emojis (Chapter 5). Thus, more detailed information about the emotional state can be obtained. This section explains how to analyze the opponent's emotional state with emojis during a negotiation.

Two main components of an emotional state are arousal and valence [59]. The arousal represents the physiological activity of an emotional state, while the valence

indicates whether the emotional state is positive or negative. Therefore, the valence value can be taken as the direction, while the arousal can be considered the magnitude of an emotional state. A human negotiator expresses their emotional state with predefined emojis (i.e., angry, sad, neutral, happy, excited) during a human-agent negotiation. Kutsuzawa *et al.* introduce a dataset that consists of the emojis with the corresponding valence and arousal values [60]. In our work, those valence values and arousal values are scaled between -1 and 1, and between 0 and 1, respectively. By multiplying those values, we can reduce those values into a single value, which we call *emotional effect*. Figure 15 displays the normalized values of the predefined emojis where the x-axis and y-axis denote the valence and arousal values, respectively. Table 3 demonstrates the emotional effect of each emoji (E) where Equation 20 shows the calculation of the emotional effect (E^t) of an emotional state (O^t).

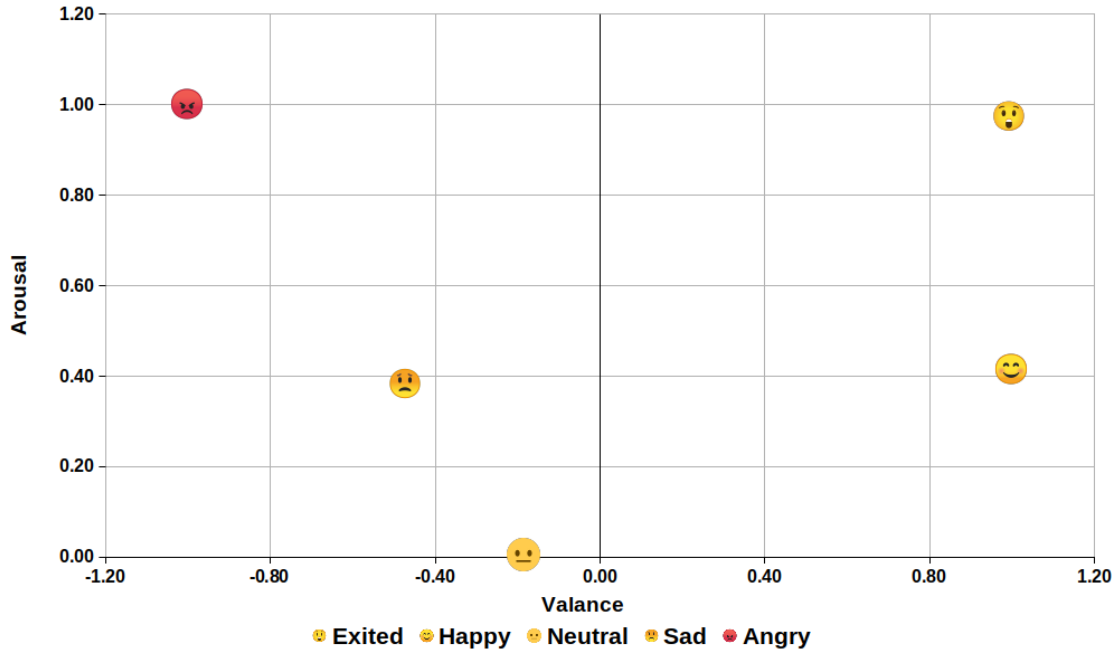


Figure 15: The Normalized Valence and Arousal Values of the Emojis

Table 3: The Emotional Effect Values of Each Emoji

Emoji	**Angry**	**Sad**	**Neutral**	**Happy**	**Excited**
Emotion Effect	-1.00	-0.17	0.02	0.44	1.0

$$E^t = \sum_i V_i * O_i^t \tag{20}$$

Solver Agent has P_E value which is calculated according to the emotional state of the opponent. This parameter can be calculated via emojis by following Equation 21 where the coefficient P_4 regulates the amount of the emotional state on the target utility[1].

$$P_E = P_4 * E^t \tag{21}$$

6.3 Opponent Modelling

Frequency-based opponent models take the value observation frequency and the issue value change rate into consideration [41, 2, 12]. Tunalı et al. presented a state-of-art approach called a windowed frequency-based opponent model [2, 12]. In this study, the windowed frequency-based opponent model is extended with arguments. For this purpose, some issue and value update rules for some argument types (rewarding, preference statement, hard constraint, and comparison) are set on the windowed frequency-based opponent model. The update rules when the classifier detects the corresponding argument type are described below.

6.3.1 Preference Statement

Preference statements express the likeness or dislikeness of a value for an issue. If the statement is positive, the corresponding evaluation value of the corresponding

[1]In our study, P_4 is taken as 0.66

value should be increased. Otherwise, it should be decreased. Therefore, the value score will be increased when a positive preference statement is received (Lines 5–7 in Algorithm 1). When a negative preference statement is detected, the value scores of the other values for the underlying issue are increased to avoid negative scores (Lines 8–10 in Algorithm 1). Regarding the importance of the issues, the human negotiator tends to make preference statements on important issues. Therefore, we update the importance of the issue positively considering the current negotiation time when a preference statement argument type is made. The issue importance update is shown in Equation 22.

6.3.2 Rewarding

A human negotiator can make rewarding argument types to indicate a concession. In this case, it is assumed that negotiators reward by conceding on their most minor preferred issues and values more than the most preferred ones. Hence, the value/weight scores of the other issues/values are increased. The weight score update is indicated in Equation 22 while the value score update is shown in Algorithm 1 (Lines 5–7). Note that the issue values appeared in the offer, but the human negotiator does not concede on as a reward is updated.

6.3.3 Comparison

A human negotiator can state preferences via comparison argument type. In comparison argument type, there are more preferred and less preferred values for an issue. This relationship should be established as a rule in the value score update process. The estimated value of more preferred values should be enforced to have higher than that of the less preferred ones as seen in Algorithm 1 (Line 12–15). Assuming that the negotiator states comparisons on essential issues; thus, the agent increases the weights of those issues that appeared in the comparison statements. The weight update of an issue is denoted in Equation 22 when a comparison is detected.

6.3.4 Hard Constraint

Human negotiators can set some hard constraints to denote an issue's most desired or undesired values. If the constraint is positive, the corresponding value should always be greater than the other values for the underlying issue. As in comparison argument type, corresponding relationship rules should be established and applied in Algorithm 1 (Lines 12–15). As we increase the importance of the issues that appeared in the comparison or preference statements, we increase the weight of the issues mentioned in the hard constraints as seen in Equation 22.

$$\begin{cases} \hat{W}_i = \hat{W}_i + \alpha_{pref} * (1-t)^{\beta_{pref}}, & \text{if } i \text{ has any preference statement} \\ \hat{W}_{others} = \hat{W}_{others} + \alpha_{rew} * (1-t)^{\beta_{rew}}, & \text{if } i \text{ has any rewarding} \\ \hat{W}_i = \hat{W}_i + \alpha_{comp} * (1-t)^{\beta_{comp}}, & \text{if } i \text{ has any comparison} \\ \hat{W}_i = \hat{W}_i + \alpha_{hard} * (1-t)^{\beta_{hard}}, & \text{if } i \text{ has any hard constraint} \end{cases} \tag{22}$$

The relationship rules established by comparison and hard constraint argument types may cause inconsistency. A cycle detection method is applied to find the inconsistent relationship rules to avoid it. If a cycle is detected, the oldest relationship rule in that cycle should be removed to maintain consistency.

A Genetic Algorithm approach is implemented to optimize the hyperparameters of the proposed opponent model. The previous human-agent negotiation data is used to maximize Kendall's Tau metric (i.e., the higher the value, the more desired it is). Note that Kendall's Tau is the most reliable evaluation metric for the performance analysis of an opponent model, which is described in Section 7.2.2. The best parameter values concerning the Genetic Algorithm are given in Table 4. It is worth mentioning that the collected data (Chapter 5) is used for the parameter tuning of the opponent model.

Table 4: The Optimized Parameter Values by Genetic Algorithm

Parameter	α	β	α_{pref}	β_{pref}	α_{hard}	β_{hard}	α_{comp}	β_{comp}	α_{rew}	β_{rew}	k	γ
Value	10	5	2	5	2	5	1	3	1	1	2	0.15

The issue weight estimation is almost the same with the distribution-based frequentist opponent model [12, 2]. The difference is that the issue scores are also updated with proper parameters when the corresponding argument type is detected, as seen in Equation 22.

Algorithm 1 Value Estimation Update

procedure VALUE ESTIMATION UPDATE($\hat{V}$,$Pref$,$Offer$,$concession$,γ)
  for each $i \in I$ **do**
    for each $j \in J_i$ **do**
      update_greater $\leftarrow False$
      if $(j \in Pref_{positive})$ **or** $(j \in Offer$ **and** $j \notin concession)$ **then**
        $\hat{V}_j^i \leftarrow \hat{V}_j^i + 1$
        update_greater $\leftarrow True$
      else if $j \in Pref_{negative}$ **then**
        $\hat{V}_k^i \leftarrow \hat{V}_k^i + 1 \; \forall \; k \neq j$
        update_greater $\leftarrow True$
      end if
      for each $\hat{V}_{greater}^i \in$ Greater$(\hat{V}_j^i)$ **do**
        if update_greater $= True$ **then**
          $\hat{V}_{greater}^i \leftarrow \hat{V}_{greater}^i + 1$
        end if
      end for
    end for
  end for
  for each $i \in I$ **do**
    for each $j \in J_i$ **do**
      $\hat{V}_j^i \leftarrow (\frac{\hat{V}_j^i}{max_j(\hat{V}^i)})^\gamma$
    end for
  end for
end procedure

6.4 Bidding Strategy

While making an offer, one challenge is generating well-targeted offers against the human negotiator. It requires considering the opponent's bid history, emotional state, and arguments. Keskin et al. introduced an emotional, and opponent-aware agent (called Solver Agent) for human-agent negotiations [9]. On the other hand, this strategy does not utilize human arguments. In this work, we extend Solver Agent to pre-process the human opponent's arguments and revise its opponent model accordingly. Although the target utility estimation is the same as the original algorithm, we apply a window-based bid search approach instead of searching for an offer with a utility closest to the target utility. As it is common, the agent employs the AC_{next} strategy to decide when to accept the opponent's offer.

Algorithm 2 demonstrates the windowed offer search with a constraint satisfaction approach. The windowed offer search approach chooses an offer around the target utility (TU) instead of the closest one. This window is defined by some thresholds (δ). This means that this approach generates an offer pool with utility values between $TU - \delta_{lower}$ and $TU + \delta_{upper}$ in the provided bid space (Line 3). Note that δ_{lower} and δ_{upper} are not equal. Afterward, the bid having the highest estimated product utility will be offered. The estimated product utility is calculated by multiplying the utility of the bid for the agent (i.e., U_{Agent}) and the estimated utility of the bid for the opponent via the opponent model (i.e., U_{Opp}^{Est}) as seen in Equation 23.

$$Product\_Utility^{Est} = U_{Opp}^{Est} * U_{Agent} \tag{23}$$

A human negotiator can specify some constraints during the negotiation. Finding offers that align with the opponent's constraints is essential for beneficial agreements. Thus, the offers conflicting with the hard and conditional constraints should be eliminated from the candidate offer list (Line 4). Our strategy chooses the offer with the highest estimated utility product (Line 16). However, if no candidate offer satisfies

the given constraints, the windowed offer search thresholds (δ) are increased until at least one candidate offer is found (Line 5–12). If the lower bound drops significantly, the agent may make undesired offers; therefore, we limit the decrease in this threshold with an α_{lower} parameter (Line 7–9). If the candidate offers have still been empty where $\delta_{lower} = \alpha_{lower}$ and $\delta_{upper} = 1 - TU$, $offers$ will be generated without constraint satisfaction (Lines 13–15). Furthermore, the defined α_{lower} and the initial values of the thresholds can vary domain by domain [2]. They should be carefully determined by considering the domain specifications. Besides, human negotiators may concede over time; therefore, their hard constraints might turn into soft constraints. The hard constraint rule is removed from the constraint set in such a case.

Algorithm 2 Windowed Offer Search with The Constraint Satisfaction

```
1:  procedure WINDOWED_OFFER_SEARCH(δ, α_lower, C)
2:      TU ← Solver Agent
3:      offers ← The offers in range [TU − δ_lower, TU + δ_upper]
4:      offers ← Eliminate(offers, C)
5:      while (offers = ∅) and (δ_lower < α_lower or δ_upper ≤ 1 − TU) do
6:          δ_upper = δ_upper + 0.01
7:          if δ_lower < α_lower then
8:              δ_lower = δ_lower + 0.01
9:          end if
10:         offers ← The offers in range [TU − δ_lower, TU + δ_upper]
11:         offers ← Eliminate(offers, C)
12:     end while
13:     if offers = ∅ then
14:         offers ← The offers in range [TU − δ_lower, TU + δ_upper]
15:     end if
16:     offer ← The offer with the maximum estimated product utility in offers
17:     return offer
18: end procedure
```

An extended version of the Solver Agent with an argument-based frequentist opponent model is presented above. With the extension, it can use not only the opponent's bid history, the passing time, and the opponent's emotional state but also the

[2] In our domain, we take $\alpha_{lower} = 0.1$, $\delta_{lower} = 0.03$ and $\delta_{upper} = 0.03$.

opponent's arguments. This proposed negotiating agent is assumed to improve the performance (e.g., the outcome and acceptance time) in human-agent negotiations. Note that the proposed agent design applies AC_{next} as the acceptance strategy.

CHAPTER VII

EVALUATION

This chapter reports the performance evaluation of the proposed hierarchical classifier and argument-based agent design. Firstly, the performance evaluation of the trained classifiers is examined (Section 7.1). Afterward, we investigated the performance of the proposed opponent model and NLPSolver Agent (Section 7.2).

7.1 Evaluation of Argument Extraction

To determine the types of the given arguments (i.e., each category in the argument taxonomy in Figure 10), we applied the classification algorithms that are briefly explained in Chapter 3. This section presents the evaluation results for each classification task in the proposed hierarchy.

7.1.1 Evaluation Metrics for Classification

For evaluation of the classifiers, the most widely used metric is accuracy denoting the percentage of the correct predictions over all predictions as shown in Equation 24. However, the accuracy may mislead us when the data is not balanced. Therefore, other metrics, such as precision and recall, are also examined. To calculate the precision and recall, a confusion matrix - a two-dimensional matrix where one of the dimensions ("Real") represents the actual class labels and the other dimension ("Predicted") denotes the predicted labels. The matrix elements show the number of instances for four categories by comparing the actual label of the instances with the predictions as follows:

- **True Positive (TP)**: The number of predicted positive instances which are actually positive.

- **False Positive (FP)**: The number of predicted positive instances which are actually negative.
- **False Negative (FN)**: The number of predicted negative instances which are actually positive.
- **True Negative (TN)**: The number of predicted negative instances which are actually negative.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}} \tag{24}$$

Accordingly, *precision*, the percentage of the instances that the classifier labeled as positive and that are in fact positive (i.e., TP), is calculated as specified in Equation 25. Similarly, Equation 26 denotes the calculation of recall, which is the percentage of instances actually present in the input that were correctly identified by the classifier.

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

$$Recall = \frac{TP}{TP + FN} \tag{26}$$

Lastly, we have another metric called *F-Score*, which is denoted F_β – a harmonic average of precision and recall. Here, β is a parameter controlling to what extent recall is as crucial as the precision, as seen in Equation 27. When β is equal to one, the precision and the recall are equally considered in the calculation.

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{(\beta^2 \times precision) + recall} \tag{27}$$

Since our dataset is unbalanced in terms of class distribution, we use F_1 to measure the performance of the classifiers. The higher F_1 score is, the better the predictions are. In addition, we also reported the $F_{0.5}$ values for the classification of constraint,

rewarding, and comparison statements because a wrong prediction for such categories will restrict the agent's bidding. Moreover, it can harm the opponent models. Therefore, we consider precision more important in such argument classifications. For the generality of the results, we adopted the 5-Fold cross-validation approach to split training and test sets, and the evaluation is made with respect to the average of these sets.

7.1.2 Text Type Classification

Recall that human negotiators exchange text written in English via a chat-based interface to send their offers, arguments, and emojis to express their current emotional state. In this task, the classifier takes the content of both sides' previous offers, the human negotiator's current emotional state (i.e., percentages of each emoji), and the given text as inputs. The aim is to predict the text category in the given taxonomy (e.g., only offer content, offer content with arguments and acceptance). The label distribution for this task is displayed in Figure 16 in the whole dataset. Note that the augmented data is not included.

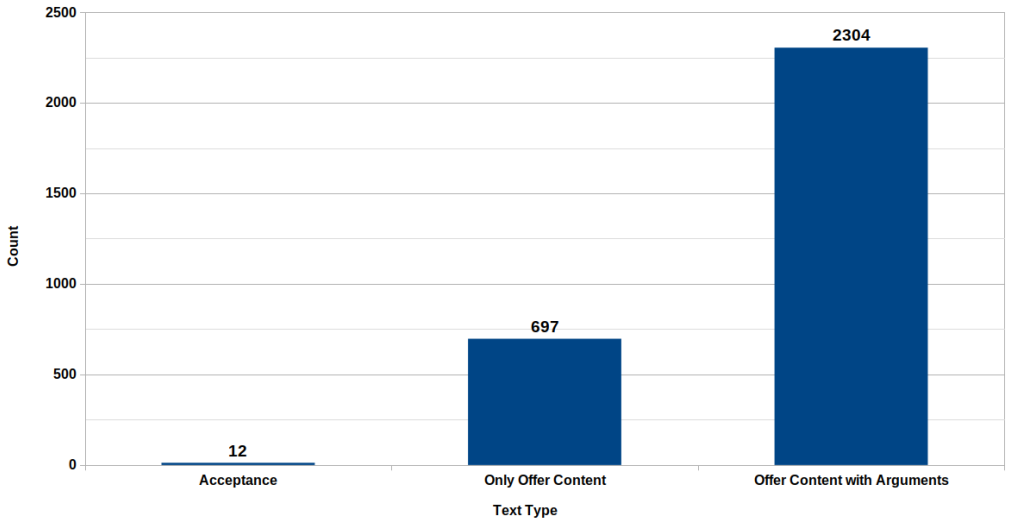


Figure 16: Distribution for Text Type Classification

The classifier may learn the noise or the random fluctuations in the training data

rather than the general concept (i.e., overfitting problem) [61]. As seen in the distribution, the labels are unbalanced and it may cause the overfitting problem. The classification model results are shown in Figure 17. Note that the mixture of experts namely *Mix. Average* and *Mix. Weighted* ensemble the predictions of the MLP and Random Forest model.

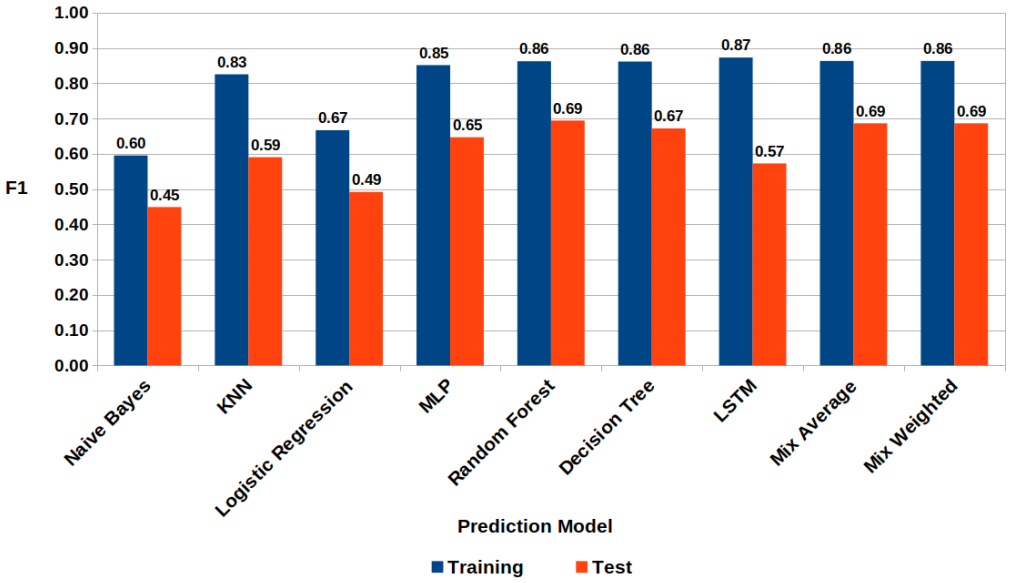


Figure 17: Classifier Results for Text Type Classification

According to those results, it can be observed that machine learning methods such as Naive Bayes, KNN, Logistic Regression, and LMTM performed poorly on the test dataset. In contrast, Random Forest and MLP performed relatively better. We could see that adopting a mixture of experts approaches does not outperform the performance of the Random Forest classifier. It is worth noting that the LSTM model has the highest F_1 score (0.87) in the training set while having a F_1 score of 0.57 in the test set. Consequently, most models can be considered overfitted. Since the performance of the Random Forest and the mixture of experts with the averaging method are almost the same, we pick the mixture of experts with the averaging model for this task. The confusion matrix of *Mix. Average* for the test sets is shown in Table 5 where each cell denotes the average of 5-fold predictions. As seen in the confusion

matrix, the model struggle with "Acceptance" class due to the lack of "Acceptance" instances.

Table 5: Confusion Matrix for the Mix. Average Model on Text Type Classification Task

		Predicted		
		Only Offer	**Acceptance**	**With Argument**
Actual	**Only Offer**	123.6	7.0	8.8
	Offer with Arguments	17.8	0.6	268.8
	Acceptance	0.8	1.4	0.2

7.1.3 Offer Classification

After determining the text type, we train models for detecting the content of the offers; therefore, we exclude the text instances whose types are "Acceptance". That is, we only use the instances involving the content of the offers. In the offer classification task, the classifiers should extract the offered value for each issue from the given text. The classifiers take the previous offer contents, the current emotional state, and the given text as inputs. Recall that we train a classifier for each negotiation issue. Unlike the other classification tasks, the label distribution is almost balanced. The classification model results are depicted in Figure 18. Note that the mixture of experts approach combines MLP, Random Forest, and Decision Tree for this type of classifier in our work.

According to the results, *Mix Average* has the highest F_1 score (0.89), while the Naive Bayes model has the lowest one (0.73) in the test set. Besides, the models, except Naive Bayes, KNN, and Logistic Regression, have a F_1 score higher than 0.90 in the training set. Our expectation was to gain higher prediction performance. The reason why we do not achieve higher prediction accuracy might stem from the fact that the textual data does not involve only offer content but also additional arguments about issues. To investigate the effect of the arguments on the offer classification task, we trained models by only using the textual data involving only content offers. Figure 19 displays the result of those models. According to the results on only offer content,

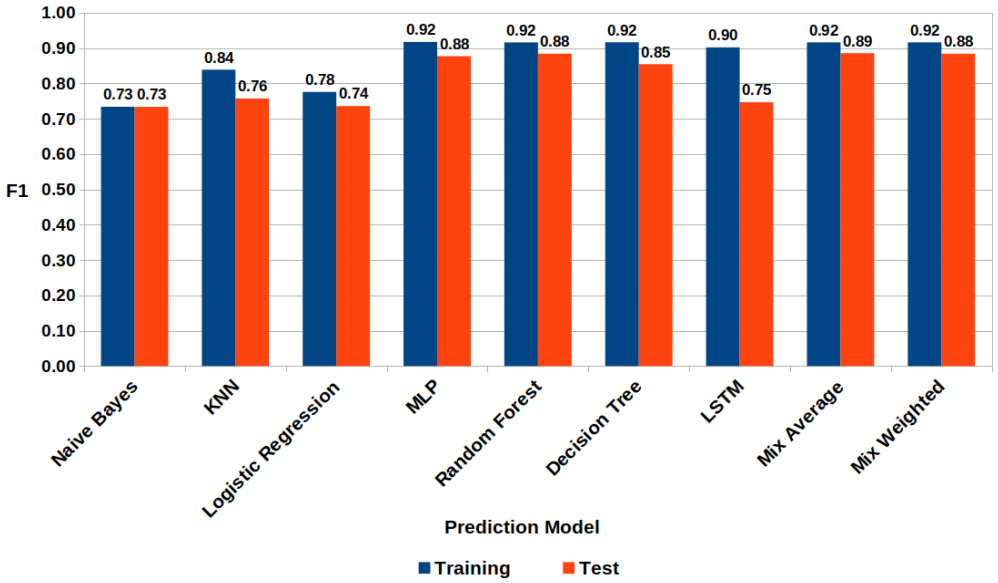


Figure 18: Classifier Results for Offer Classification

most models increase F_1 score slightly in the test set. MLP has the highest F_1 score (from 0.88 to 0.93). This result indicates that arguments in a text are challenging to extract the offer content. In contrast to our expectation, The slight increase in the performance triggers us to examine data elaborately. We found out that some textual data involves only partial offer content.

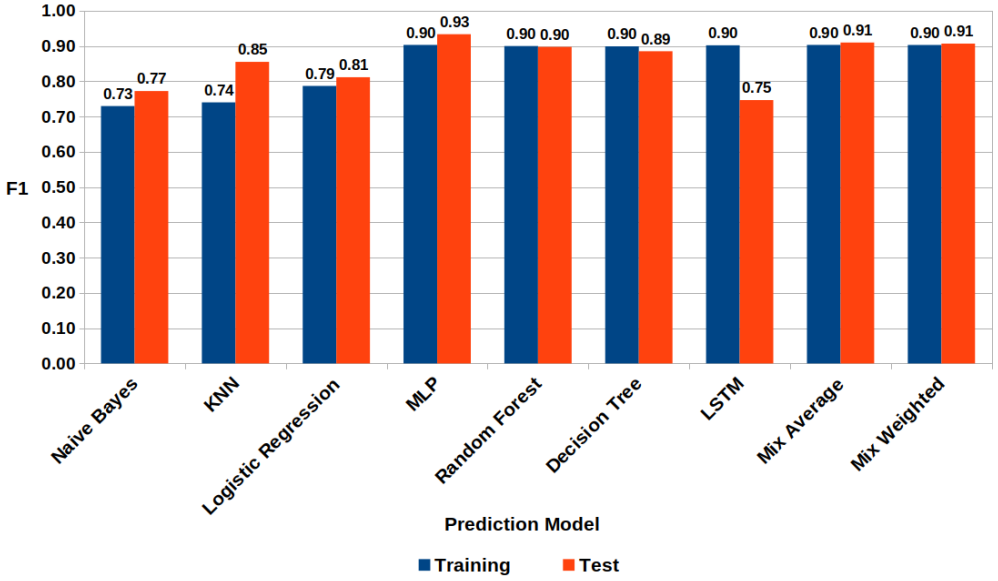


Figure 19: Classifier Results for Offer Classification on Only Offer Content

7.1.4 Argument Type Classification

The argument type classification task is to determine the more-grained category of the given arguments, such as rewarding, awareness feedback, etc. Therefore, we train a binary classifier for each argument type. Unlike text type and offer classification tasks, the classifiers also take the current offer content as input. Figure 20 shows the distribution of each argument type without augmented data. As seen in distribution, some argument type has insufficient positive instances that may cause an overfitting problem.

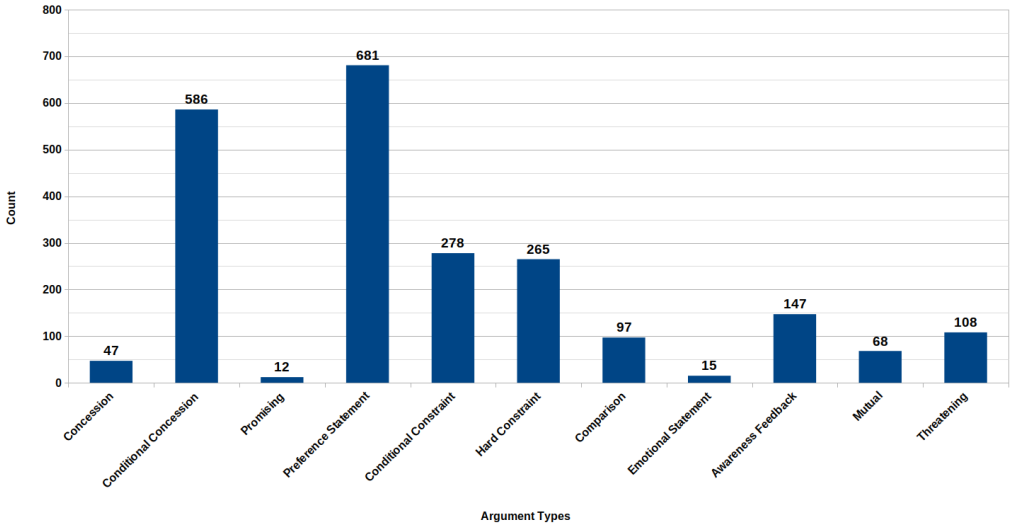


Figure 20: Distribution for Argument Type Classification

According to the results in Figure 21, MLP has the highest F_1 score (0.93), whereas Naive Bayes has the lowest one (0.63) in the test set. Note that these results are the average of each argument type. We can see that the performance of the predictors is reasonably good. Table 6 displays the confusion matrix of MLP on the test set.

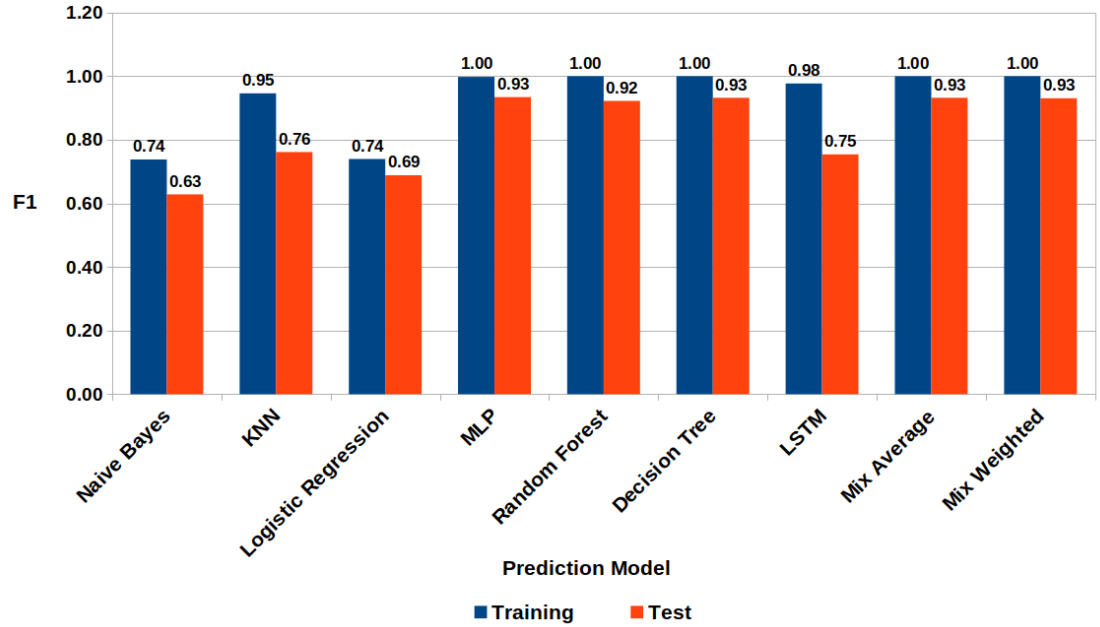


Figure 21: Classifier Results for Argument Type Classification

Table 6: Confusion Matrix of MLP Model on Argument Type Classification Task

		Predicted	
		Negative	**Positive**
Actual	**Negative**	242.4	2.6
	Positive	3.0	39.0

7.1.5 Rewarding Classification

If the given textual instance involves a rewarding argument type, this type of classifier will extract the rewarding type (i.e., concession or conditional concession) with the corresponding issue-value pairs. For this purpose, the models should determine whether a given possible value for the underlying issue was rewarded (e.g., conceded by the opponent) in the argument. That is, we iterate over all possible values for the chosen issue in that domain and ask the predictor whether that value appears in the rewarding part of the text. In more detail, the class labels are *concession*, *conditional concession*, and *no reward*. If there is no reward presented for the issue,

it returns *no reward*. Suppose the reward is presented as a conditional concession (e.g., "London is fine if the season is summer" in the holiday domain). Otherwise, it predicts as *concession* (e.g., "Ok, we can go to London as you wish"). Figure 22 shows the results of these types of classifiers. Since precision is more important than recall for this task, we also report the $F_{0.5}$ scores as seen in Figure 23.

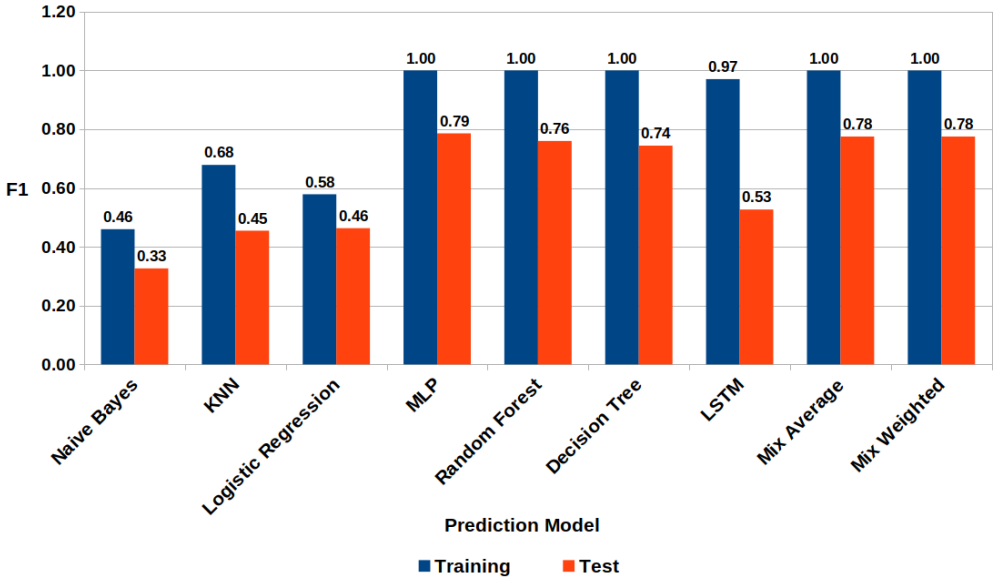


Figure 22: Classifier F_1 Results for Rewarding Classification

According to the results, MLP has the highest F_1 (0.79) and $F_{0.5}$ (0.91) scores, while Naive Bayes has the lowest F_1 (0.33) and $F_{0.5}$ (0.41). However, MLP has 1.00 F_1 and 1.00 $F_{0.5}$ scores on the training set. Accordingly, we can consider that the model may be overfitted. The confusion matrix of MLP is displayed in Table 7. In the table, "Rewarding-c" represents *concession* while "Rewarding-cc" represents *conditional concession*. Note that *no reward* label has the highest instances due to iterating on each possible value.

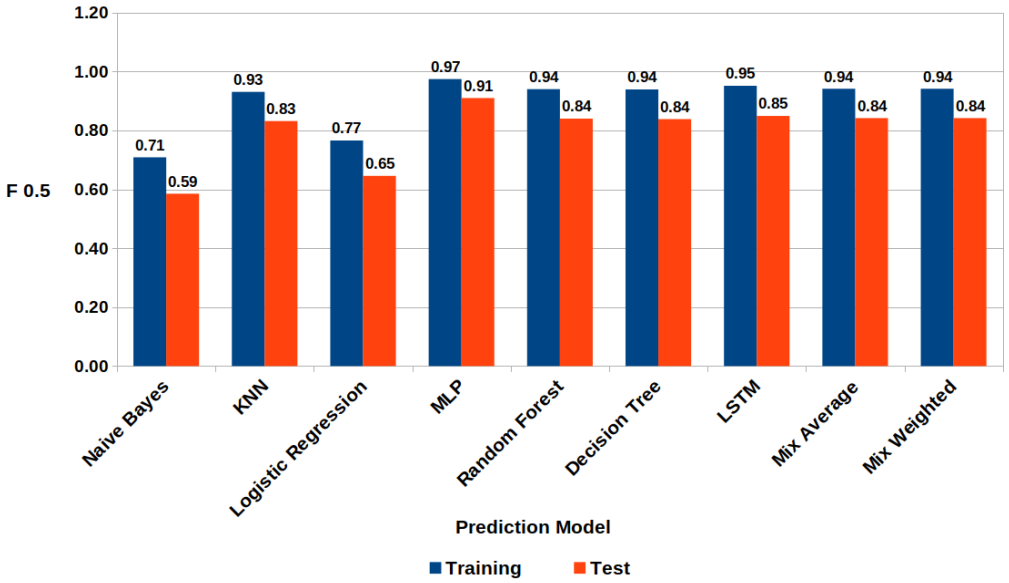


Figure 23: Classifier $F_{0.5}$ Results for Rewarding Classification

Table 7: Confusion Matrix of MLP Model on Rewarding Classification Task

		Predicted		
		No Reward	**Rewarding-c**	**Rewarding-cc**
Actual	**No Reward**	214.6	1.6	25.4
	Rewarding-c	1.2	4.2	3.4
	Rewarding-cc	17.2	1.4	107.8

7.1.6 Constraint Classification

Similar to the rewarding classification task explained above, we aim to extract the proper label with the corresponding issue values from the given argument. That is, we model a classifier that predicts whether the given values for negotiation issues are associated with the given constraints. It outputs the constraint types for the given issue-value assignment. The class labels for the constraint argument type are *positive preference statement*, *negative preference statement*, *positive hard constraint*, *negative hard constraint*, *positive conditional constraint*, *negative conditional constraint* and *no constraint*. The distribution of the labels is shown in Figure 24.

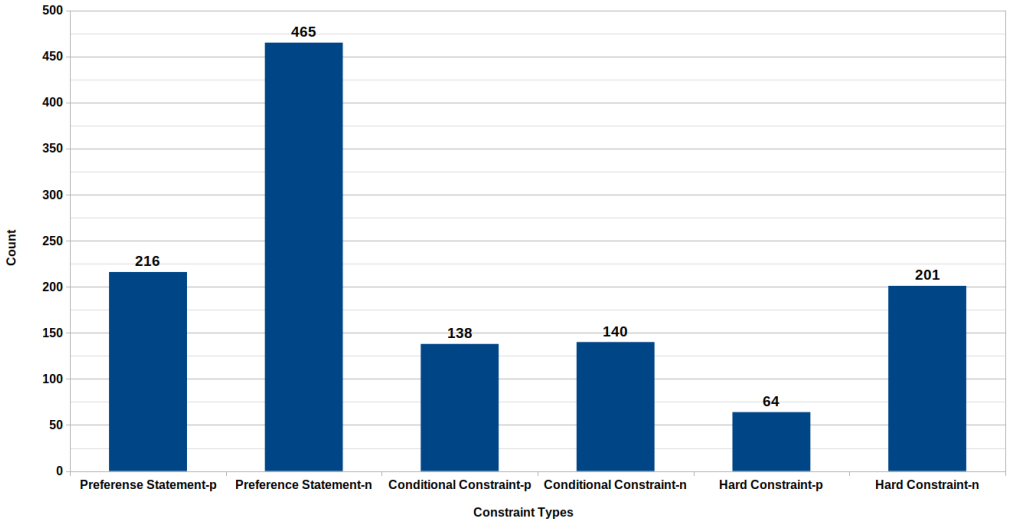


Figure 24: Distribution for Constraint Classification

For evaluation of the classifiers, both F_1 and $F_{0.5}$ scores are reported. Figure 25 and Figure 26 denote the F_1 scores amd the $F_{0.5}$ scores, respectively. According to the results, *Mix. Weighted*, which combines MLP and Random Forest, has the highest F_1 (0.80) and $F_{0.5}$ (0.77) scores on the test set. However, *Mix. Weighted* has F_1 (1.00) and $F_{0.5}$ (1.00) scores on the training set; thus, the model may be overfitted. The confusion matrix of the *Mix. Weighted* is shown in Table 8. Besides, Naive Bayes has the lowest F_1 (0.11) and $F_{0.5}$ (0.19) scores on the test set.

Table 8: Confusion Matrix of Mix. Weighted Model on Constraint Classification Task

		Predicted						
		No Const.	**Con. Const.-p**	**Con. Const.-n**	**Pref.-p**	**Pref.-n**	**Hard Const.-p**	**Hard Const.-n**
Actual	**No Const.**	3,638.8	5.6	11.8	5.4	28.0	1.2	12.0
	Con. Const.-p	2.6	23.4	0.4	0.2	0.6	0.0	0.0
	Con. Const.-n	3.6	0.2	22.4	0.4	0.8	0.0	0.2
	Pref.-p	6.4	0.8	0.4	37.6	0.6	0.2	0.4
	Pref.-n	9.0	0.0	0.4	0.4	77.2	0.2	0.8
	Hard Const.-p	0.8	0.0	0.0	0.4	0.2	5.2	0.0
	Hard Const.-n	3.8	0.0	0.2	0.0	1.0	0.0	36.0

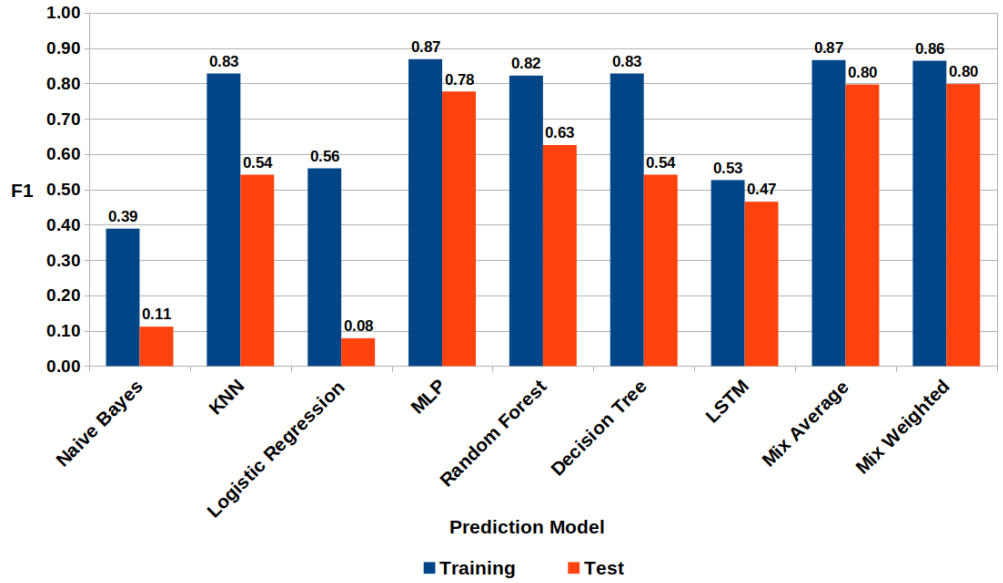


Figure 25: Classifier F_1 Results for Constraint Classification

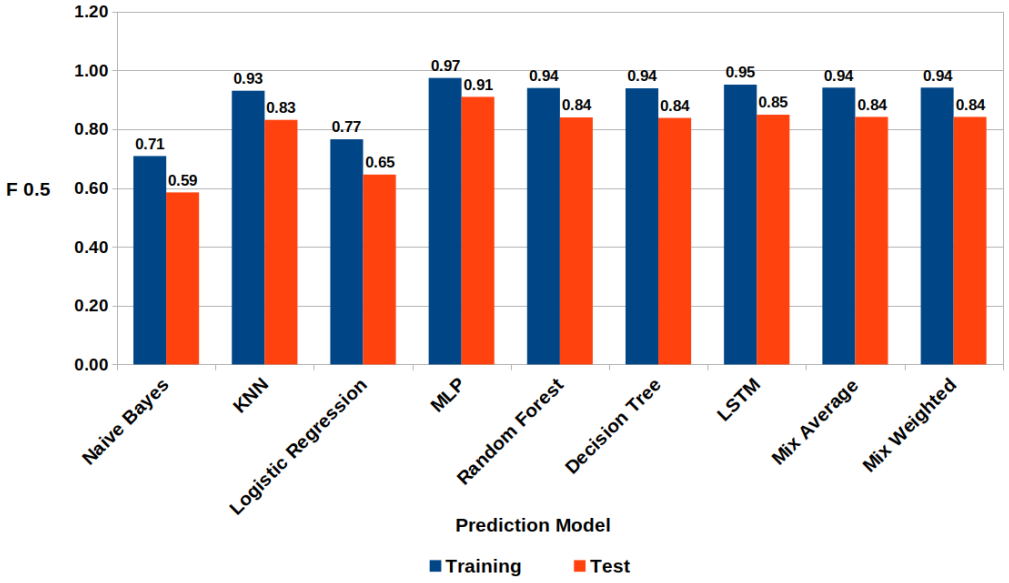


Figure 26: Classifier $F_{0.5}$ Results for Constraint Classification

7.1.7 Comparison Classification

The last classification task is the comparison classification task. Similar to rewarding and constraint classification tasks, the classifiers are used to iterate on each possible

value to detect the associated issue values in a comparison argument. The corresponding labels are *comparison* and *no comparison*. Therefore, binary classifiers can be applied to this task. Figure 27 and Figure 28 show the F_1 scores and the $F_{0.5}$ scores, respectively.

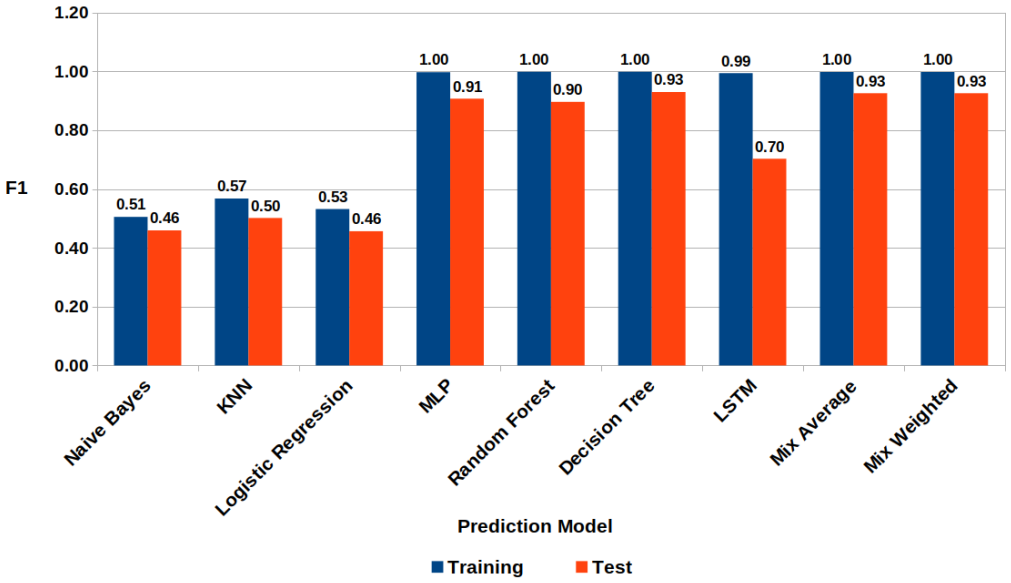


Figure 27: Classifier F_1 Results for Comparison Classification

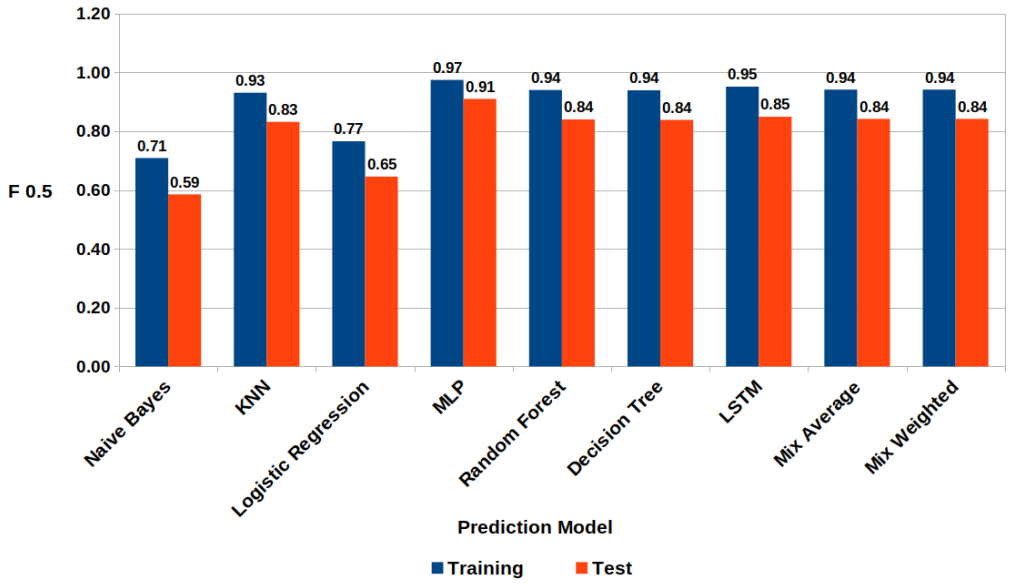


Figure 28: Classifier $F_{0.5}$ Results for Comparison Classification

According to the results, *Mix. Average*, which combines MLP, Random Forest, and Decision Tree, has the highest F_1 (0.93) and $F_{0.5}$ (0.93) scores on the test set. Besides, Naive Bayes and Logistic Regression have the lowest F_1 (0.46) and $F_{0.5}$ (0.50) scores on the test set. The confusion matrix of *Mix. Average* on the test set is shown in Table 9.

Table 9: Confusion Matrix of Mix. Average Model on Comparison Classification Task

		Predicted	
		No Comparison	**Comparison**
Actual	**No Comparison**	411.4	1.6
	Comparison	3.0	16.0

7.2 Evaluation of Agent Design

To evaluate the proposed argument-based agent, we conducted human experiments where the human participants negotiated with the agents in a bilateral fashion over a chat-based interface. Our settings are described in Section 7.2.1. We first evaluate the performance of the proposed opponent model in Section 7.2.2 and then the bidding strategy in Section 7.2.3. For the performance evaluation, the argument-based opponent model is compared with the state-of-art frequency-based opponent models.

7.2.1 Experimental Setup

In order to evaluate the performance of the proposed approach, we created a Holiday domain consisting of four issues. According to the designed scenario, the participants will go on a vacation with another person they do not know each other. They should negotiate where to go (Destination), where to stay (Accommodation), when to go (Season), and how to travel in the city (Transportation). In our setting, participants negotiate separately with two agents, Solver Agent and NLPSolver Agent. We use the same values for both agents as seen in Table 10.

It is a role-playing game; however, it works better when the participants embrace

Table 10: Parameters of Agents

	P_0	P_1	P_2	P_3	P_A	δ_{lower}	δ_{upper}	α
Solver Agent	0.9	0.8	0.5	0.5	0.5	-	-	-
NLPSolver Agent	0.9	0.8	0.5	0.5	0.5	0.03	0.03	0.1

their preferences instinctively. Therefore, before each session, participants are asked to order the issues regarding their importance and possible values per each issue in line with their preferences. These orderings are mapped onto a predefined utility function. Accordingly, a conflicting preference profile is generated for the agent. It is worth noting that each party knows its preferences during the negotiation. The negotiation issues and their possible values are displayed in Table 11, with an example order and scores. To reduce the learning effect between consecutive sessions, we slightly changed the destination for their second negotiation. Accordingly, the utility space for all possible outcomes on the Holiday scenario is displayed in Figure 29. Note that the Nash solution denotes the fairest solution for both sides where both parties can take 0.80 utility.

Table 11: The Weights of Issues and Values of Holiday Domain

Domain: Holiday

Issues	**Accommodation** (0.4)	**Destination** (0.3)	**Transportation** (0.2)	**Season** (0.1)
Values	Caravan (1.0)	London / Amsterdam (1.0)	Subway (1.0)	Winter (1.0)
	Hotel (0.8)	Berlin / Barcelona (0.8)	Bus (0.8)	Summer (0.75)
	House (0.6)	Tokyo / Boston (0.6)	Taxi(0.6)	Spring (0.5)
	Hostel (0.4)	Paris / Milan (0.4)	Car (0.4)	Autumn (0.25)
	Tent (0.2)	Istanbul / Seoul (0.2)	Bike (0.2)	

During the experiments, argument types were recognized by the trained classifiers in the experimental data, in which 64 participants participated. At the end of both experiment sessions, the participants filled out a questionnaire for each agent to compare them. In total, 40 participants participated negotiation experiments, where all of them ended with an agreement. We analyze negotiation logs and survey responses to evaluate the proposed approach in the following session. It is worth noting that participants gave their consent voluntarily to participate in this experiment and

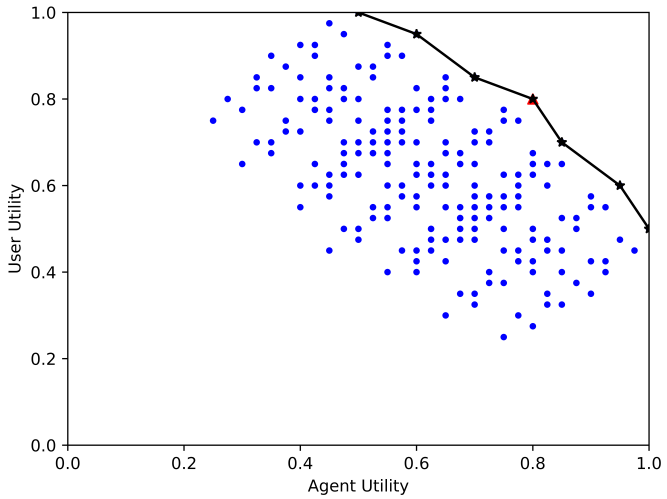


Figure 29: The Utility Space for all Possible Outcomes on the Holiday Scenario

received a supermarket gift for their effort.

7.2.2 Evaluation of Opponent Model

The proposed opponent model is compared with the basic frequency-based [41], and distribution-based frequentist [2] opponent models. For the performance evaluation of an opponent model, the following evaluation metrics are used in the literature:

- **Root Mean-Square Error (RMSE)**: This metric states the difference between the utility of a bid (u_i) and the estimated utility of that bid ($\hat{u}_i$). The opponent model calculates the estimated utility of all bids in the corresponding domain, then the estimated and real utilities are compared. The RMSE always is greater or equal than 0, and a lower RMSE value is desired. Equation 28 displays the calculation of RMSE. Note that N means the number of bids in that domain.

$$RMSE = \sqrt{\frac{\sum_i^N (u_i - \hat{u}_i)^2}{N}} \tag{28}$$

- **Spearman's Rank Correlation Coefficient (Spearman)**: This metric measures the difference between the real and estimated bid orders [62]. The opponent model predicts the estimated utilities, then all bids in the corresponding domain are sorted by the estimated utility. The difference between the estimated and real orders should be observed. For this observation, the difference between the ordering indices is taken into consideration. Spearman calculates a correlation value (ρ) by comparing the estimated ordering indices ($\hat{o}_i$) and the real ordering indices (o_i). This correlation value is in the range between -1 and 1. 1 means the given orders are the same, so higher values are desired. Equation 29 shows the calculation of Spearman, also n indicates the number of the bids in the corresponding domain.

$$\rho = 1 - \frac{6 * \sum_i^n (o_i - \hat{o}_i)^2}{n * (n^2 - 1)} \tag{29}$$

- **Kendall's Rank Correlation Coefficient (Kendall's Tau)**: This metric also measures the difference between the real and estimated bid orders [63]. Kendall's Tau compares the concordant and discordant pairs in the given bid orders. Also, Kendall's Tau has three variants which are Tau-a (τ_a), Tau-b (τ_b), and Tau-c (τ_c). Tau-b is usually preferred to measure the performance of an opponent model. τ_b is always in range between -1 and 1. $\tau_b = 1$ means the orders are the same while $\tau_b = -1$ means the orders are opposite. Therefore, a higher τ_b value is desired. The calculation of τ_b is indicated in Equation 30.

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_c + n_d + n_r) * (n_c + n_d + n_e)}} \tag{30}$$

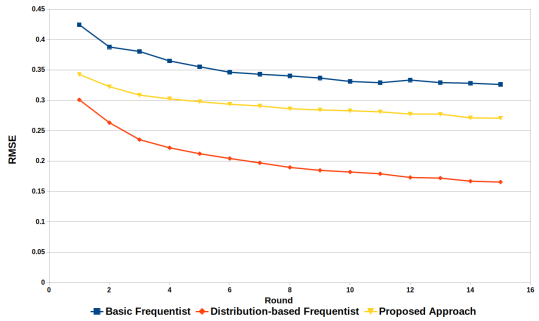
In that calculation, n_c is the number of the concordant pairs while n_d is the number of the discordant pairs. Additionally, n_r is the number of ties only in the real bid ordering while n_e is the number of ties only in the estimated bid

ordering. To calculate the number of concordant pairs (n_c) and the number of the discordant pairs (n_d), two pair instances which are (o_i, $\hat{o}_i$) and (o_j, $\hat{o}_j$) are compared. If $o_i < o_j$ and $\hat{o}_i < \hat{o}_j$, or vice versa, these pairs are concordant pairs. Otherwise, these pairs are discordant pairs. Note that i is always less than j.

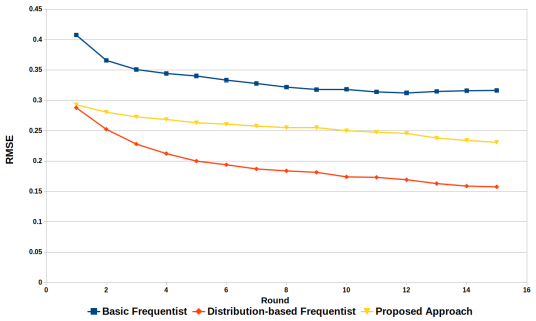
It is worth remarking that we used Kendall's Tau metric as the fitness function of the Genetic Algorithm for fine-tuning the parameters of the proposed opponent model (Section 6.3). Here, the main challenge is that learning the utility of an issue value requires a certain amount of offer exchanges. If the value of a given issue does not appear in the opponent's offer history, making a wise guess on its preference is not trivial. However, human negotiators have a tendency to end their negotiation in a short time, which might be insufficient for an opponent model [13, 18, 16, 17]. This fact exhibits a need to use various information received from the opponent to estimate the opponent's preferences accurately.

All metrics described above were applied for the comparison of the opponent models. For RMSE, the actual and estimated offer utility are compared round by round. For Spearman and Kendall's Tau, the actual offer rankings and estimated offer rankings are compared round by round. The collected data and the evaluation experiment data are used to evaluate the argument-based opponent model with state-of-art approaches. Note that the hyper-parameters of the opponent model were tuned by Genetic Algorithm on only the collected data. Figure 30a denotes the RMSE results of the data collection experiments where actual argument types manually detected by the domain experts are considered to update the opponent modeling. Figure 30b displays the results of the evaluation experiment, where the trained classifiers predict the argument types. It is worth mentioning that the average negotiation round in the data collection experiment is 15, whereas the average round of the evaluation experiment is 10. Remark that a lower RMSE value is desired.

Although the argument-based opponent model achieves a lower RMSE value than



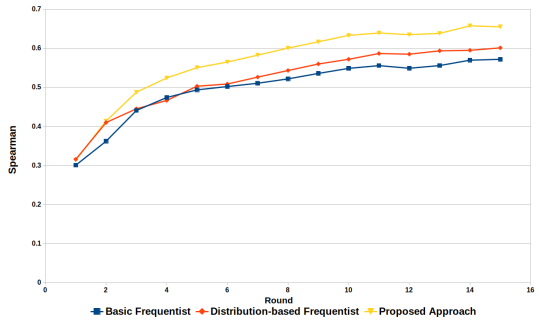
(a) Data Collection Experiment Results



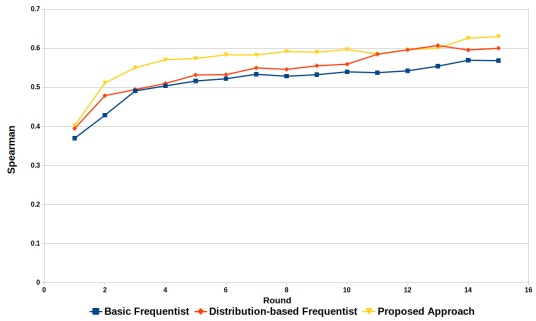
(b) Evaluation Experiment Results

Figure 30: RMSE Results of the Opponent Models

the basic frequency-based opponent model, as seen in Figure 30, it cannot outperform the distribution-based frequentist opponent model in both evaluations. However, the performance of the proposed bidding strategy, which utilizes an opponent model, highly depends on the precision of the estimated offer ranking rather than the estimated offer utility. That means that the ranking correlation metrics (i.e., Spearman and Kendall's Tau) should be taken into more consideration for the evaluation of the opponent models. That is why the fitness function of the Genetic Algorithm is the Kendall's Tau in the parameter tuning of the opponent model. Figure 31a and Figure Figure 32a show the Spearman and the Kendall's Tau results of the data collection experiment, respectively. Similarly, Figure 31b and Figure 32b demonstrate those of the evaluation experiment. Remark that a higher value is desired for both Kendall's Tau and Spearman.

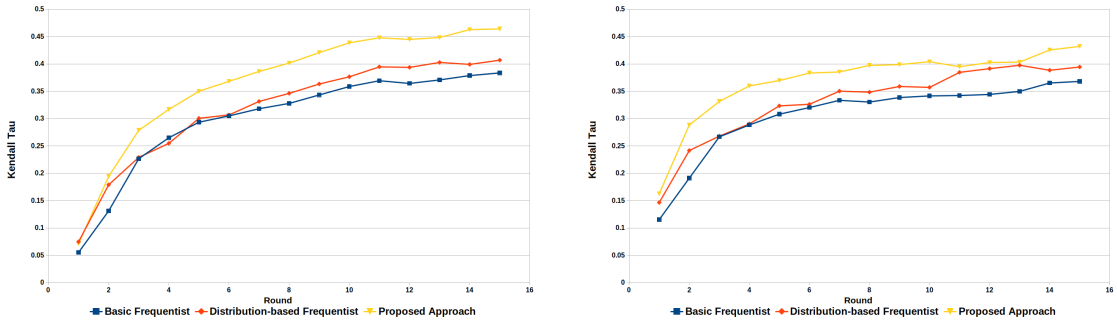


(a) Data Collection Experiment Results



(b) Evaluation Experiment Results

Figure 31: Spearman Results of the Opponent Models



(a) Data Collection Experiment Results

(b) Evaluation Experiment Results

Figure 32: Kendall's Tau Results of the Opponent Models

According to the evaluation results, the proposed opponent model outperforms the state-of-art frequentist opponent models in terms of the estimation of the offer ranking, by considering the human arguments during the negotiations. However, the proposed approach in the evaluation experiment cannot achieve the performance as well as in the data collection experiment due to the insufficient performance of the classifiers. Nevertheless, it still performs better than other models.

7.2.3 Evaluation of the Agent

We analyze the received individual utilities for both users and agents to study whether our extension improves the performance of the negotiating agent. Figure 33 denotes a box-whisker plot on the received utilities for both agents. We applied Kolmogorov–Smirnov normality and homogeneity test to see whether we could directly apply the t-pairwise dependent test for a significant difference in the results. Consequently, we apply the t-pairwise dependent test. The results show a statistically significant difference in the received utility between Solver Agent and NLPSolver Agent ($p = .00256 < 0.05$) with a 95 confidence interval. However, there is no significant difference in the human participants' utilities ($p = .13019$). That shows that without worsening the human negotiator's gain, our agent can improve its received utility thanks to incorporating the arguments into opponent modeling and bidding strategy.

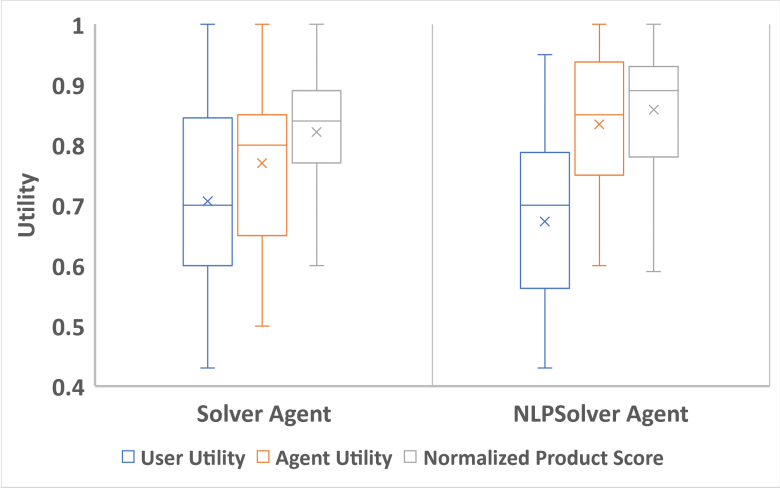


Figure 33: Average Agreement Utility for both Agents and Human Negotiators

Moreover, Figure 34 displays the bidding distribution of the strategies during negotiation. As shown from the figure, on average, NLPSolver Agent made offers closer to Pareto-frontier by increasing both the agent and user utilities compared to the Solver agent. We can say that it is still a competitive agent, but the offers proposed by the NLPSolver Agent dominate the Solver Agent's offers in general.

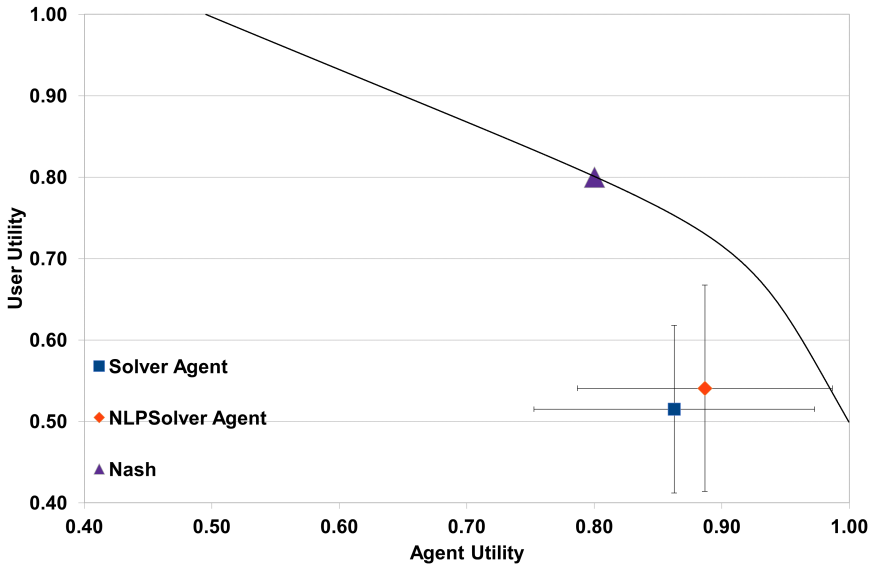
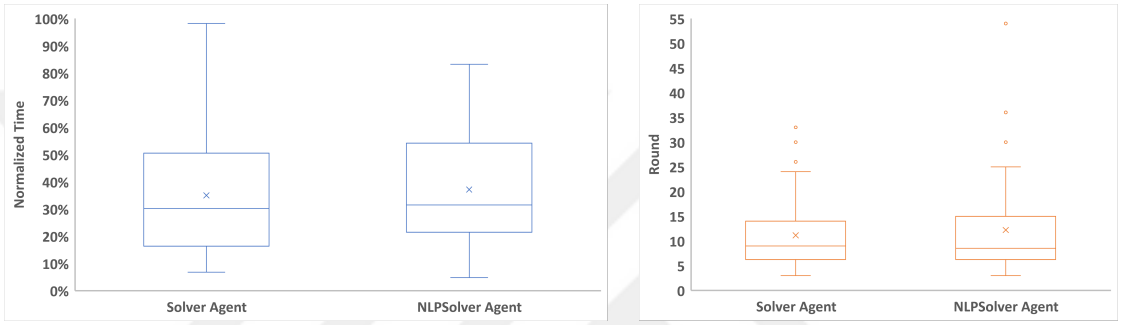


Figure 34: Bidding Distribution of Agents

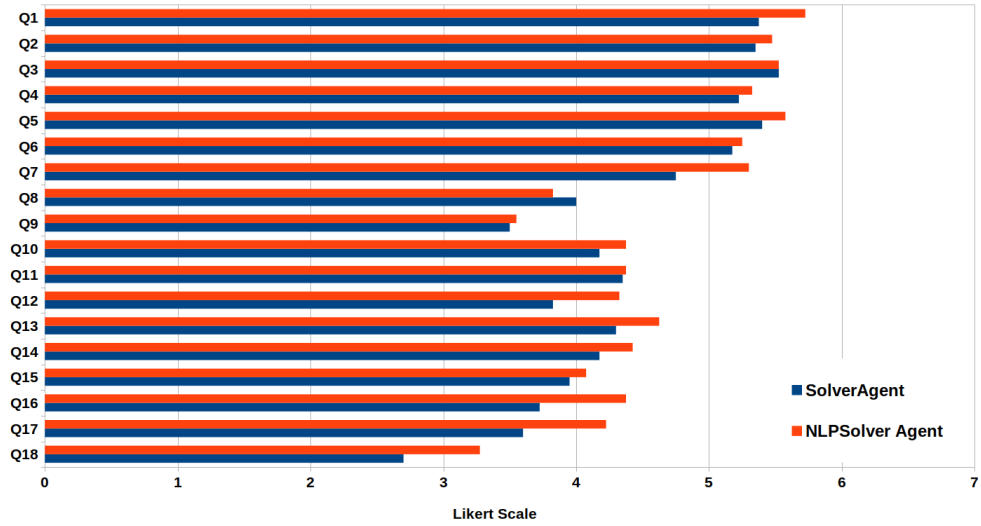
We also analyze the agreement times. Figure 35 shows each agent's average agreement round and time to complete their negotiation with a box-whisker plot. It seems that the Solver Agent slightly reached agreements sooner. However, when we apply the statistical significance tests, there is no statistical difference in the negotiation round and time to reach an agreement ($p = .4725$ and $p = .44401$ for time and round, respectively).



(a) Normalized Times (b) Rounds

Figure 35: Average Agreement Times & Rounds For Agreement

Apart from analyzing the negotiation logs, we analyze the feedback given by the participants during the post-experiment surveys. Figure 36 shows the average ratings of 7-scaled Likert questionnaire responses after each negotiation session. We applied Kolmogorov–Smirnov normality test and homogeneity test. The response data satisfied the precondition for the t-pairwise dependent test. The results of this test show that there is a statistically significant difference in Question 7 ($p = .03074$), Question 16 ($p = .04333$), and Question 17 ($p = .0375$) for the Solver and NLPSolver agent settings. Figure 37 shows the box-whisker plot on the responses for these questions. Recall that four corresponds to neutral; thus, the above 4 indicates their agreement. The participants agree that they observed their opponent consider their preferences (3.7 versus 4.4 out of 7 after negotiating Solver and NLPSolver, respectively) and arguments (3.6 versus 4.2 out of 7 after negotiating Solver and NLPSolver, respectively) while making its offer more when they negotiated with the NLPSolver. That shows they were aware of the agent's attitude change concerning arguments and preferences.



Q-1 I made arguments (e.g., explanatory, rewarding, etc.) during the negotiation.
Q-2 I tried to convince my opponent via some arguments.
Q-3 I shared some of my preferences with my opponent via arguments.
Q-4 I expressed how I feel during the negotiation by means of some arguments.
Q-5 I expressed my emotional state during the negotiation.
Q-6 My emotional state influenced my bidding behavior.
Q-7 **My opponent's attitudes were important for my offer during negotiation.**
Q-8 My opponent negotiated in a fair way.
Q-9 My opponent tried to find the best deal for both of us.
Q-10 My opponent negotiated with me like a human negotiator.
Q-11 My opponent adapted her strategy with respect to my current emotional state.
Q-12 My opponent modified her bidding behavior inline with my offers during the negotiation.
Q-13 I believed that my arguments influenced my opponent"s bidding behavior.
Q-14 While my opponent was making her offers, I observed that s/he considered my negotiation behavior (e.g., selfish, collaborative).
Q-15 While my opponent was making her offers, I observed that s/he considered the remaining time.
Q-16 **While my opponent was making her offers, I observed that s/he considered my preferences.**
Q-17 **While my opponent was making her offers, I observed that s/he considered my arguments.**
Q-18 I observed that my opponent did not offer the options, after I expressed that I did not want.

Figure 36: Average Ratings of Questionnaire Responses

As seen in Figure 36, the participants agree that they made arguments (e.g., explanatory, rewarding) in general in both settings (around 5.5). Similarly, they agree that they express their preferences via arguments and use some arguments to convince their opponents. Figure 38 shows the total number of arguments given by 40 participants. The majority tended to share rewarding and preference statements and some conditional constraints. Although the NLPSolver Agent is more collaborative than the Solver Agent, it is still a competitive agent, as seen in bidding analysis in Figure 34. Responses to our questions, such as Q8 and Q9, show that the participants disagreed that the agents found the best deal for both of them (i.e., fairness).

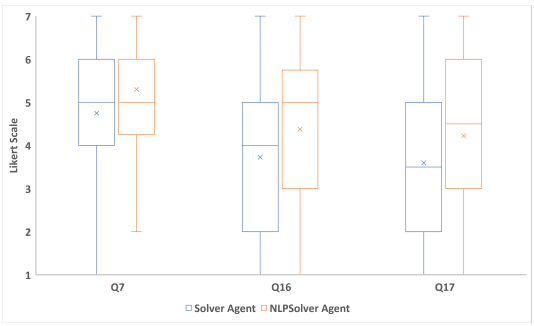


Figure 37: Box Plot of Questions with Significant Difference

Regarding whether our agent negotiated as a human negotiator, the average rate of the participants is around 4-5 and slightly higher in the case of NLPSolver Agent (4.4 versus 4.2). The average rating for Q11 is approximately 4.4, so the participants noticed that the agents adapted their behavior concerning their emotional feedback. However, there is still room for improvement.

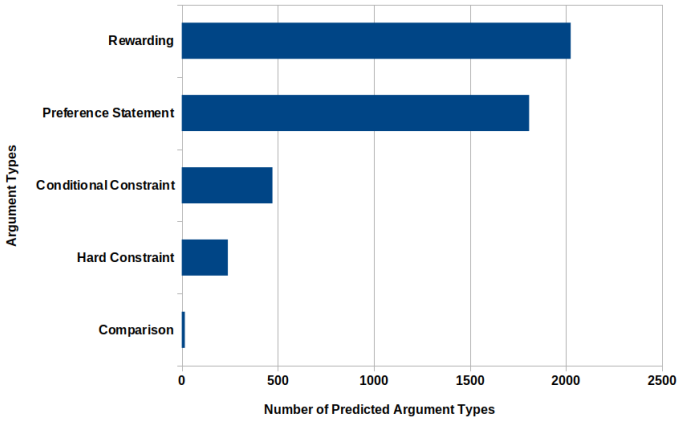


Figure 38: Number of Argument Types Predicted by the Classifiers

Furthermore, while the participants still disagree with the statement that agents did not offer options after they expressed they did not want them, this agreement rating is higher for the NLPSolver Agent (3.3 versus 2.7 for NLPSolver Agent and Solver Agent, respectively). It may stem from our agent considering the soft and hard constraints while offering. That supports our agent's success; however, the human participants are not entirely convinced. Therefore, this part could be revised in the following studies.

CHAPTER VIII

CONCLUSION

Understanding human negotiators' arguments and acting with respect to them is one of the challenges in the human-agent negotiation systems. For this purpose, a taxonomy of argument types for negotiation is presented, and a hierarchical classifier approach is proposed to preprocess and extract meaningful information from the given arguments. Moreover, the extracted preferential information is utilized in opponent modeling and bidding strategies. To evaluate the performance of the proposed approach, human-agent negotiation experiments were conducted. The experimental evaluation shows that the proposed argument-based frequency opponent modeling approach outperformed the state-of-the-art frequentist models in the literature. In addition, the NLPSolver bidding strategy beats the human negotiators and receives higher utility on average than the Solver Agent.

As future work, we plan to enhance the bidding strategy so that human negotiators' preferences are fully captured, and agents can generate automatic arguments specifying that it considers human negotiators' concerns. Consequently, agents can give the impression that they consider their opponent's arguments. Furthermore, the accuracy of the argument prediction has an influence on the opponent model and bidding strategy. We plan to improve the predictors' performance by applying more sophisticated deep learning approaches such as GloVe or BERT-based models [28, 26, 54].

REFERENCES

[1] K. Hindriks, C. Jonker, and D. Tykhonov, "Let's dans! an analytic framework of negotiation dynamics and strategies," *Web Intelligence and Agent Systems*, vol. 9, pp. 319–335, 01 2011.

[2] O. Tunalı, R. Aydoğan, and V. Sanchez-Anguix, "Rethinking frequency opponent modeling in automated negotiation," in *PRIMA: Principles and Practice of Multi-Agent Systems* (B. An, A. Bazzan, J. Leite, S. Villata, and L. van der Torre, eds.), (Cham), pp. 263–279, Springer International Publishing, 2017.

[3] I. Marsa-Maestre, M. Klein, C. M. Jonker, and R. Aydoğan, "From problems to protocols: Towards a negotiation handbook," *Decision Support Systems*, vol. 60, pp. 39–54, 2014.

[4] S. Fatima, S. Kraus, and M. Wooldridge, *Principles of automated negotiation*. Cambridge, England: Cambridge University Press, 2014.

[5] T. Baarslag, K. Hindriks, M. Hendrikx, A. Dirkzwager, and C. Jonker, *Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies*, pp. 61–83. Tokyo: Springer Japan, 2014.

[6] R. Aydoğan, T. Baarslag, K. Fujita, J. Mell, J. Gratch, D. de Jonge, and et al., "Challenges and main results of the automated negotiating agents competition (ANAC) 2019," in *Multi-Agent Systems and Agreement Technologies*, (Macao, China), pp. 366–381, Springer International Publishing, 2019.

[7] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Robotics and Autonomous Systems*, vol. 24, no. 3, pp. 159–182, 1998.

[8] D. d. Jonge and C. Sierra, "NB3: a multilateral negotiation algorithm for large, non-linear agreement spaces with limited time," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 5, pp. 896–942, 2015.

[9] M. O. Keskin, U. Çakan, and R. Aydoğan, "Solver agent: Towards emotional and opponent-aware agent for human-robot negotiation," in *Proceedings of the AAMAS*, AAMAS '21, (Richland, SC), p. 1557–1559, International Foundation for Autonomous Agents and Multiagent Systems, 2021.

[10] V. Sanchez-Anguix, R. Aydoğan, V. Julian, and C. Jonker, "Unanimously acceptable agreements for negotiation teams in unpredictable domains," *Electronic Commerce Research and Applications*, vol. 13, no. 4, pp. 243–265, 2014.

[11] T. Baarslag, K. Hindriks, and C. Jonker, *Acceptance Conditions in Automated Negotiation*, pp. 95–111. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[12] V. Sanchez-Anguix, O. Tunalı, R. Aydoğan, and V. Julian, "Can social agents efficiently perform in automated negotiation?," *Applied Sciences*, vol. 11, no. 13, pp. 1–26, 2021.

[13] R. Aydoğan, O. Keskin, and U. Çakan, "Would you imagine yourself negotiating with a robot, jennifer? why not?," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 1, pp. 41–51, 2022.

[14] B. Türkgeldi, C. Özden, and R. Aydogan, "The effect of appearance of virtual agents in human-agent negotiation," *AI*, vol. 3, pp. 683–701, 08 2022.

[15] J. Hoegen, D. DeVault, and J. Gratch, "Exploring the function of expressions in negotiation: the dynego-woz corpus," *IEEE Transactions on Affective Computing*, pp. 1–12, nov 2022.

[16] M. Lewis, D. Yarats, Y. Dauphin, D. Parikh, and D. Batra, "Deal or no deal? end-to-end learning of negotiation dialogues," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 2443–2453, Association for Computational Linguistics, Sept. 2017.

[17] H. He, D. Chen, A. Balakrishnan, and P. Liang, "Decoupling strategy and generation in negotiation dialogues," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 2333–2343, Association for Computational Linguistics, Oct.-Nov. 2018.

[18] K. Chawla, J. Ramirez, R. Clever, G. M. Lucas, J. May, and J. Gratch, "Casino: A corpus of campsite negotiation dialogues for automatic negotiation systems," *CoRR*, vol. abs/2103.15721, pp. 3167–3185, 2021.

[19] M. A. Bakker, M. J. Chadwick, H. R. Sheahan, M. H. Tessler, L. Campbell-Gillingham, J. Balaguer, N. McAleese, A. Glaese, J. Aslanides, M. M. Botvinick, and C. Summerfield, "Fine-tuning language models to find agreement among humans with diverse preferences," 2022.

[20] A. Rosenfeld and S. Kraus, "Providing arguments in discussions on the basis of the prediction of human argumentative behavior," *ACM Trans. Interact. Intell. Syst.*, vol. 6, dec 2016.

[21] R. Hadfi, J. Haqbeen, S. Sahab, and T. Ito, "Argumentative conversational agents for online discussions," *Journal of Systems Science and Systems Engineering*, vol. 30, pp. 450–464, Aug 2021.

[22] A. Rosenfeld, I. Zuckerman, E. Segal-Halevi, O. Drein, and S. Kraus, "Negochat-a: a chat-based negotiation agent with bounded rationality," *Autonomous Agents and Multi-Agent Systems*, vol. 30, pp. 60–81, Jan 2016.

[23] J. Mell and J. Gratch, "Iago: Interactive arbitration guide online (demonstration)," in *Adaptive Agents and Multi-Agent Systems*, AAMAS '16, (Richland,

SC), p. 1510–1512, International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[24] O. Güngör, U. Çakan, R. Aydoğan, and P. Özturk, "Effect of awareness of other side's gain on negotiation outcome, emotion, argument, and bidding behavior," in *Recent Advances in Agent-based Negotiation* (R. Aydoğan, T. Ito, A. Moustafa, T. Otsuka, and M. Zhang, eds.), (Singapore), pp. 3–20, Springer Singapore, 2021.

[25] C. M. Jonker, R. Aydoğan, T. Baarslag, K. Fujita, T. Ito, and K. Hindriks, "Automated negotiating agents competition (ANAC)," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, (Melbourne), pp. 5070–5072, AAAI Press, 2017.

[26] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, pp. 4171–4186, 2018.

[27] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Berlin, Germany), pp. 225–230, Association for Computational Linguistics, Aug. 2016.

[28] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[29] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, and et al., "Training compute-optimal large language models," 2022.

[30] R. Yang, J. Chen, and K. Narasimhan, "Improving dialog systems for negotiation with personality modeling," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, (Online), pp. 681–693, Association for Computational Linguistics, Aug. 2021.

[31] Y. Zhou, H. He, A. W. Black, and Y. Tsvetkov, "A dynamic strategy coach for effective negotiation," *CoRR*, vol. abs/1909.13426, pp. 367—378, 2019.

[32] K. Chawla, R. Clever, J. Ramirez, G. Lucas, and J. Gratch, "Towards emotion-aware agents for negotiation dialogues," *International Conference on Affective Computing and Intelligent Interaction*, vol. abs/2107.13165, pp. 1–8, oct 2021.

[33] K. Chawla, G. Lucas, J. May, and J. Gratch, "Opponent modeling in negotiation dialogues by related data adaptation," in *Findings of the Association for Computational Linguistics: NAACL*, (Seattle, United States), pp. 661–674, Association for Computational Linguistics, July 2022.

[34] L. Amgoud and H. Prade, "Generation and evaluation of different types of arguments in negotiation," in *10th International Workshop on Non-Monotonic Reasoning (NMR)*, (Whistler, BC, Canada), pp. 10–15, Proceedings of International Workshop on Non-Monotonic Reasoning, June 2004.

[35] S. Kraus, K. Sycara, and A. Evenchik, "Reaching agreements through argumentation: a logical model and implementation," *Artificial Intelligence*, vol. 104, no. 1-2, pp. 1–69, 1998.

[36] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, "A framework for argumentation-based negotiation," in *Intelligent Agents IV Agent Theories, Architectures, and Languages* (M. P. Singh, A. Rao, and M. J. Wooldridge, eds.), (Berlin, Heidelberg), pp. 177–192, Springer Berlin Heidelberg, 1998.

[37] R. Aydoğan, D. Festen, K. V. Hindriks, and C. M. Jonker, *Alternating Offers Protocols for Multilateral Negotiation*, pp. 153–167. Cham: Springer International Publishing, 2017.

[38] R. Aydoğan, K. V. Hindriks, and C. M. Jonker, "Multilateral mediated negotiation protocols with feedback," in *Novel Insights in Agent-based Complex Automated Negotiation*, pp. 43–59, Tokyo: Springer Japan, 2014.

[39] T. Baarslag, K. Hindriks, and C. Jonker, *A Tit for Tat Negotiation Strategy for Real-Time Bilateral Negotiations*, pp. 229–233. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[40] R. M. Vahidov, G. E. Kersten, and B. Yu, "Human-agent negotiations: The impact agents' concession schedule and task complexity on agreements," in *50th Hawaii International Conference on System Sciences* (T. Bui, ed.), (Hawaii, USA), pp. 1–9, 2017.

[41] T. van Krimpen, D. Looije, and S. Hajizadeh, *HardHeaded*, pp. 223–227. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[42] N. Indurkhya and F. J. Damerau, *Handbook of Natural Language Processing*. Chapman & Hall/CRC, 2nd ed., 2010.

[43] E. Alpaydin, *Introduction to Machine Learning*. London, England: The MIT Press, 2nd ed., 2010.

[44] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, no. 1/2, pp. 167–179, 1967.

[45] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," tech. rep., Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[46] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[47] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, vol. 3, pp. 41–46, IBM New York, 2001.

[48] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Taylor & Francis, 1984.

[49] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[51] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 606–615, Association for Computational Linguistics, Nov. 2016.

[52] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[53] N. Zainuddin and A. Selamat, "Sentiment analysis using support vector machine," in *International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 333–337, 09 2014.

[54] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, pp. 1–5, 2019.

[55] Z. S. Harris, "Distributional structure," *WORD*, vol. 10, no. 2-3, pp. 146–162, 1954.

[56] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[57] R. Dehkharghani, Y. Saygin, B. Yanikoglu, and K. Oflazer, "Sentiturknet: a turkish polarity lexicon for sentiment analysis," *Language Resources and Evaluation*, vol. 50, pp. 667–685, Sep 2016.

[58] K. Hindriks and D. Tykhonov, "Opponent modelling in automated multi-issue negotiation using bayesian learning," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, (Richland, SC), p. 331–338, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[59] J. Russell and L. Barrett, "Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant," *Journal of Personality and Social Psychology*, vol. 76, pp. 805–19, 06 1999.

[60] G. Kutsuzawa, H. Umemura, K. Eto, and Y. Kobayashi, "Classification of 74 facial emoji's emotional states on the valence-arousal axes," *Scientific Reports*, vol. 12, p. 398, 01 2022.

[61] L. Yu, S. Wang, and K. K. Lai, "An integrated data preparation scheme for neural network data analysis.," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, pp. 217– 230, 03 2006.

[62] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 100, no. 3/4, pp. 441–471, 1987.

[63] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.

VITA

Anıl Doğru is a member of OzU Interactive Intelligent Systems Lab. He received his bachelor's degree from the department of Computer Science at Özyeğin University in 2019. After his graduation, he started his master's studies at the same institution. He has worked as a research and teaching assistant under the supervision of Asst. Prof. Reyhan Aydoğan. During his M.Sc. thesis, he has studied on various topics such as the Multi-Agent Negotiation Systems, Machine Learning, Natural Language Processing, Computer Vision, and so on.