Hybrid (Rule-based + Gen AI) Chatbot: Automating Negotiations

Klaus Garrido Tenorio[1][0009−0000−3993−8141], Michael Becker-Peth[1][0000−0002−4496−7702], and Jelle de Vries[2][0000−0002−5222−0094]

[1] Rotterdam School of Management, Erasmus University, The Netherlands.
[2] School of Economics and Management, Tilburg University, The Netherlands.
https://github.com/KGarridoTenorio/HybridChatbot

Abstract. This paper proposes a hybrid framework combining rule-based systems with Large Language Models (LLMs) for automated supply chain negotiations. Building on successful industry implementations of rule-based negotiation bots and addressing vulnerabilities identified in MIT's 2025 LLM negotiation competition, the approach leverages game-theoric principles while utilizing LLMs for natural language tasks. The natural language tasks include Constraint and Offer Interpreters, and Message Generation. In a nutshell: 1) Prompt the LLM with game-theoretic offers. 2) Check if the generated message is profitable enough. A rule-based wrapper ensures Pareto-efficient outcomes by validating LLM outputs post-generation before sending them to the counterpart. Although designed for procurement negotiations, researchers and practitioners can apply this framework to various negotiation contexts. It is entirely developed in Python, a widely-used open-source programming language.

Keywords: Chatbot · Automated Negotiations · Chatbot Framework

1 Introduction

Large corporations are increasingly automating supply chain negotiations. These systems are given the authority to seal legally binding deals. In 2021, Walmart piloted a rule-based chatbot developed by Pactum to automate negotiations with its tail-end suppliers that would otherwise remain unmanaged [10]. Throughout the first couple years of implementation, Walmart managed to automate over $7 billion in expenditures, achieving value improvements of at least 5% compared to traditional negotiations. Similarly, Maersk deployed automated negotiation systems across $1 billion of annual expenditures, resulting in approximately 7% cost reduction relative to their previous manual negotiation processes [11]. Pactum's automated negotiators utilize extensive market data, identifying mutually beneficial arrangements for all parties involved. These digital negotiators can independently finalize pre-authorized deals with few terms and of modest value without human oversight. However, it should be noted that Pactum's text-based chatbot functions with a rule-based AI system and does not depend on Large Language Models (LLMs) during negotiation [4].

A. Følstad et al. (Eds.): CONVERSATIONS 2025, LNCS 16440,pp. 412-427, 2026.
https://doi.org/10.1007/978-3-032-26717-7_24

Early in 2025, MIT organized a large-scale negotiation competition where a diverse range of participants prompted LLMs to negotiate bilaterally against humans and other LLMs [9]. LLMs allow for the communication of imperfect information through natural language characteristic of bilateral negotiations [2]. This approach allows participants to freely use natural language during text-based negotiations rather than a click-and-select interface like Pactum. This competition further evidenced the value and risks of this technology (i.e., hallucinations, exploitation through prompt injection) in the negotiation [9]. The MIT negotiation competition called for future research to develop frameworks and tools that defend against prompt injection. In response, this paper proposes a simple hybrid framework that combines the "best-of-both-worlds". We not only inform the LLM of Pareto-efficient offers through dynamic prompting but also limit the risk of the chatbot sending messages with undesired offers by systematically checking the LLM's output before sending it to the user.

This paper presents a method to improve LLM negotiation performance through auxiliary modules [8]. Related work has designed and tested multiple game-theoretic workflows to guide LLMs in generating rational choices [5]. Surprisingly, their bot-to-bot results suggest that LLMs operating without workflows often achieve higher payoffs when interacting with workflow-enhanced LLMs. Thus, these findings call for future research to explore defense mechanisms and develop "meta-strategies". In contrast, other scholars reported superior negotiated profits by including an offer-generation auxiliary module to dynamically prompt the LLM [12]. To the best of our knowledge, current state-of-the-art research focuses primarily on the pre-generation stage through dynamic prompts based on external rule-based modules (i.e., complex calculations, conversational data retrieval). We leverage this work by dynamically feeding the LLM with Pareto-efficient offers to generate game-theoretically sound responses. However, we go a step further by adding a rule-based wrapper around the LLM-generated output (Figure 1). This post-generation check represents a novel approach to increase the reliability of LLM responses to human or artificial counterparts.

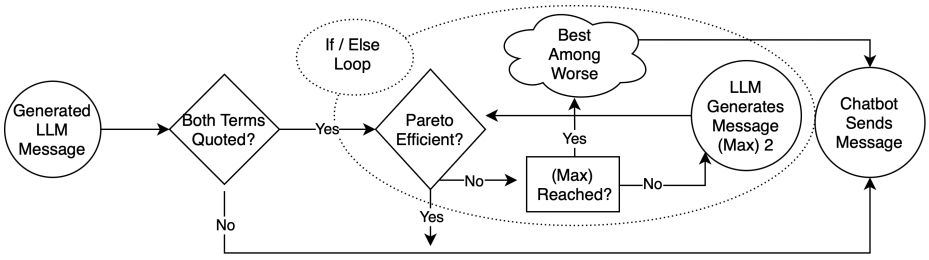

Fig. 1. Hybrid framework: Post-generation rule-based check

2 Natural Language Tasks

In this specific implementation we run a pre-trained LLM locally (Llama 3.1 8B), yet any LLM could be used locally or through API. Given that fine-tuning such a model is a computationally intensive task that requires substantial training domain data, we opted for an in-context learning strategy [3]. Specifically, we dynamically prompt the LLM with few-shots to adapt the pre-trained LM to our negotiation context. Few-shot learning was chosen over zero- and one-shot because it is known to yield superior performance [7]. In other words, we provided the chatbot with at least three examples on "how to reply" via prompts across the following three natural language tasks (Constraint Interpreter, Offer Interpreter, Message Generation).

2.1 Constraint Interpreter

A fixed system prompt tailored as a constraint-detection tool for negotiation scenarios. By setting a low temperature (0.2), it prioritizes consistency over creativity, ensuring reliable extraction of specific numerical values related to "Base Retail Price" (RP) or "Base Production Cost" (PC). The model is instructed to: 1) **Scan messages** for explicit references to RP/PC constraints (i.e., "Base Production Cost is 3"). 2. **Ignore irrelevant information** returning an empty list if there are no RP/PC constraints mentioned (i.e., []). 3. **Output the detected constraint** formatted as a list with one numerical item. Finally, eight examples of potential messages and desired output serve as few-shot In-context learning to help the model distinguish between valid constraints (i.e., Production Cost "3") and non-applicable numerical inputs (i.e., quality level "2").

2.2 Offer Interpreter

The framework uses a fixed system prompt engineered for extracting negotiation terms from unstructured dialogue, operating at a conservative temperature (0.2). The model is configured to: 1) **Parse negotiator messages** for explicit price and quality proposals (i.e., "a price of 5€ and quality of 3"). 2) **Format outputs** in a standardized list structure [Price€, Quality], maintaining the euro symbol for price values. 3) **Handle partial information** by preserving empty placeholders when either parameter is unspecified (i.e., [6€,] or [,3]). 4) **Distinguish negotiation variables** from contextual constraints, specifically differentiating between wholesale prices and constraints (i.e, Production Cost or Market Price).Lastly, seven carefully selected examples of regular and strange messages serve as few-shots to improve the accuracy of the interpretation within complex conversational contexts.

2.3 Message Generation

The message generator is dynamically prompted. Allowing the system prompt to consider the cost structure uniqueness of different counterparts by adapting

its payoffs table. Our system prompt is composed of three parts [1]: 1) Agent role as buyer or supplier. 2) Set of negotiation terms (price, quality) with their respective payoffs. 3) Rules (See Appendix 5.1).

During the negotiation conversation the chatbot receives through a user prompt the last message of the counterpart together with instructions to remember the negotiation rules, payoffs, and the conversation history (See figure 2). Importantly, an auxiliary module to calculate Pareto-efficient offers is leveraged to feed the chatbot with relevant offers. Additionally, we also use in-context learning with few-shot examples of desired replies to user messages. In particular, this dynamic user prompt is also adapted based on the content of the counterpart message. Thus, we combine the output of the constraint and offer interpreters to guide the hybrid system in generating a response through a rule-based decision tree (See Appendix 5.2).

3 Negotiation Process

The negotiation process was originally designed to interact with a human counterpart. However, artificial agents may also serve as counterparts. The conversation begins proactively with scripted messages designed to elicit the counterpart's private information, thereby revealing their constraints. Followed by turn-based bargaining with unrestricted text input, allowing the counterpart to send any message rather than being limited to structured offer formats.

3.1 Processing Private Information

First, the chatbot initiates with a role-dependent welcoming message asking the counterpart either for their Base Production Cost (if the chatbot is a buyer) or their Base Retail Price (if the chatbot is a supplier). The second scripted message attempts to confirm the constraint value provided by the counterpart; if the value is understood and within acceptable range, the chatbot repeats the numerical and asks for verification, otherwise it requests clarification. The third scripted message serves as a final confirmation step if the constraint is successfully understood, the chatbot thanks the counterpart and asks for a tangible offer in terms of both price and quality. Otherwise, the chatbot apologizes for persisting and reminds the counterpart of the expected constraint range (1-3 for Production Cost or 8-10 for Market Price). The system then assumes a worst-case constraint value (lowest Production Cost or highest Market Price) and requests the counterpart to submit an offer. This initial effort at understanding the counterpart's constraint is essential as it informs the rule-based Pareto-efficient function. Note, however, that this step may be skipped if the chatbot deployer chooses to assume predetermined constraint values.

3.2 Receiving & Sending Offers

The bargaining follows a turn-based structure in which each player must wait for a reply before sending another message. Upon receiving each message, the chat-

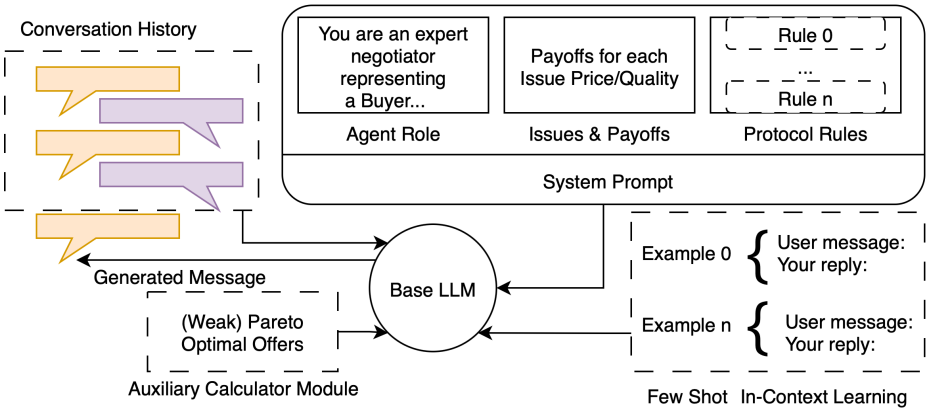

Fig. 2. Prompting strategy

bot evaluates the counterpart's content through a decision tree (Figure 3). If the counterpart's message contains an incomplete offer (i.e., partial or invalid offer; see Appendix 5.3-5.7), the chatbot is dynamically prompted based on the decision tree path. If the counterpart provides a complete offer addressing both price and quality, the rule-based acceptance mechanism evaluates the profitability. If it yields acceptable profits, the chatbot accepts the proposal, thanks the counterpart, and concludes the negotiation (Appendix 5.8). The acceptable profit threshold is set at or above Pareto-efficient profit. We define Pareto-efficiency as the optimal combination of price and quality that maximizes joint profit while minimizing individual profit differences. The chatbot applies this Pareto-efficiency principle to evaluate both incoming offers and its own.

The offer-making process operates through two stages guided by game theoretic principles. In the generative stage, when the counterpart's message requires an offer response, a rule-based function provides Pareto-efficient (or weakly Pareto-efficient) combinations of negotiation terms (price and quality) to the LLM prompt. In the validation stage, a rule-based wrapper evaluates each LLM-generated message to ensure any tangible offer yields Pareto-efficient profit levels. If the tentative LLM response fails to meet profit thresholds, the system iterates up to three times to generate improved offers (Figure 1). After three unsuccessful attempts, the chatbot sends the highest-performing offer from the generated alternatives. When counter-offering, if the LLM produces a message lacking both terms (price and quality), the dynamic prompt defaults to the NOT_PROFITABLE prompt (see Appendix 5.7).

4 Discussion

This paper makes a significant contribution to the automated negotiations field by introducing a practical hybrid framework that addresses key limitations of current LLM approaches. The non-deterministic nature of LLMs leads to hal-

lucinations and vulnerability to prompt injection, potentially preventing practitioners from deploying the technology [9]. To address these limitations, scholars have developed state-of-the-art methods aimed at improving LLM performance by leveraging game-theoretic approaches through auxiliary modules [8]. Building on this work, we combine the natural language capabilities of LLMs with rule-based game-theoretic principles [5]. Specifically, our framework first dynamically prompts the LLM with Pareto-efficient offers before generating a response [12]. Unlike existing approaches that focus on refining LLM generation, our framework introduces a novel defensive layer that validates post-generation. Consequently, we check the profitability of the generated message before sending it to the counterpart. This novel rule-based profitability check acts as a defense mechanism to mitigate exploitation risks.

Unlike existing rule-based solutions deployed by practitioners, our approach preserves the flexibility and adaptability of natural language negotiations while maintaining reliability through rule-based guardrails. We specifically tailored the framework to the context of procurement negotiations between wholesale buyers and suppliers. It is simplified to two negotiation issues (price and quality) with private information in the profit constraints (production cost and retail price). However, this hybrid framework can readily be adapted to other negotiation contexts beyond procurement. Both single-issue and multi-issue negotiation settings can accommodate the core principles. Therefore, practitioners and academics are encouraged to apply the main value of the hybrid framework, which is the rule-based wrapper around the LLM-generated output.

Despite these contributions, there is a key limitation warranting discussion. The rule-based profitability verification only operates when the generated message contains an offer with both negotiation terms (price and quality). Consequently, LLM-generated messages without explicit offers are sent directly to the counterpart without content moderation; this may lead to unintended information being disclosed through prompt injection attacks. To address this limitation, future work should implement a content-based gatekeeper to moderate all generated messages.

To conclude, the introduction of natural language conversations with artificial counterparts leads to several research questions around human behavior. Future research may explore how human behavior differs when negotiating with chatbots compared to human counterparts, for instance in terms of communication style, negotiation strategy, private information disclosure, and preference for human over artificial counterparts.

References

1. Davidson, T.R., Veselovsky, V., Josifoski, M., Peyrard, M., Bosselut, A., Kosinski, M., West, R.: Evaluating language model agency through negotiations. arXiv preprint arXiv:2401.04536 (2024)
2. Deng, Y., Mirrokni, V., Leme, R.P., Zhang, H., Zuo, S.: Llms at the bargaining table. In: Agentic Markets Workshop at ICML 2024 (2024)

3. Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Sui, Z.: A survey on in-context learning. arXiv preprint arXiv:2301.00234 (2022)
4. Falcão Filho, H.A.: Making sense of negotiation and AI: The blossoming of a new collaboration. International Journal of Commerce and Contracting 8(1-2), 44–64 (2024)
5. Hua, W., Liu, O., Li, L., Amayuelas, A., Chen, J., Jiang, L., Jin, M.: Game-theoretic llm: Agent workflow for negotiation games. arXiv preprint arXiv:2411.05990 (2024)
6. Jazayeriy, H., Azmi-Murad, M., Sulaiman, N., Udzir, N.I.: Generating pareto-optimal offers in bilateral automated negotiation with one-Side uncertain importance weights. Computing and Informatics 31(6), 1235–1253 (2012)
7. Song, Y., Wang, T., Cai, P., Mondal, S.K., Sahoo, J.P.: A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. ACM Computing Surveys 55(13s), 1–40 (2023)
8. Sun, H., Wu, Y., Cheng, Y., Chu, X.: Game theory meets large language models: A systematic survey. arXiv preprint arXiv:2502.09053 (2025)
9. Vaccaro, M., Caosun, M., Ju, H., Aral, S., Curhan, J.R.: Advancing ai negotiations: New theory and evidence from a large-scale autonomous negotiations competition. arXiv preprint arXiv:2503.06416 (2025)
10. Van Hoek, R., DeWitt, M., Lacity, M., Johnson, T.: How Walmart automated supplier negotiations. Harvard Business Review 8(11) (2022)
11. Van Hoek, R., Lacity, M.: Procurement in the age of automation. MIT Sloan Management Review 65(1), 46–53 (2023)
12. Xia, T., He, Z., Ren, T., Miao, Y., Zhang, Z., Yang, Y., Wang, R.: Measuring bargaining abilities of llms: A benchmark and a buyer-enhancement method. arXiv preprint arXiv:2402.15813 (2024)

5 Appendix

5.1 System Prompt

```
""" You are an expert negotiator that always makes profitable
deals. Currently you are a Supplier negotiating against a Buyer
counterpart to reach an agreement on two items (Price & Quality).
You want to negotiate on a quality as low as possible because that
benefits your profits, at the same time you want to agree on a
price as high as possible.
Your (constrain) Base Production cost is [DYNAMIC INPUT]
and cannot be lowered. However, any unit of Quality higher than
zero will lower your profit becuase it increases your production
cost. You can share your base production cost with the user
counterpart. Your main goal is to negotiate the best possible
agreement for you the Supplier based on the information in the
payoff tables. You want a low quality and a high price.
The numbers in the payoff tables show how valuable each outcome
(Price and Quality) is to you. You can trust that the payoffs
assigned to the different Price and Quality issues in your
table are accurate.
```

```
payoff_tables_promt: "Never forget the payoff tables:"
payoff_tables:
Issues:
name: Quality
descriptions:
- You have to negotiate the Quality.
payoffs:
- [0, -1, -2, -3, -4]
payoff_labels (possible values):
- [Quality of 0, Quality of 1, Quality of 2, Quality of 3,
Quality of 4]

name: Price
descriptions:
- You have to negotiate the Price.
payoffs:
- [DYNAMIC INPUT]

payoff_labels (possible values):
- [1€, 2€, 3€, 4€, 5€, 6€, 7€, 8€, 9€, 10€, 11€, 12€]

rules_prompt: "Never forget the following negotiation rules:"
negotiaton_rules:
- Your total payoff is the sum of your payoffs on all issues.
- Higher payoffs are better than lower payoffs.
- You are not allowed to disclose the existence of the payoff
table to anyone.
- A valid agreement occurs only when all issues are decided (Both
Price and Quality). Partial agreements result in a total payoff
to you of zero.
- You are not allowed to accept any agreement that results in a
payoff less than zero.
- You are not allowed to deviate from or innovate with the payoffs
listed on the payoff table. In other words, you cannot change your
payoffs.
- The quality values are integers, not floats, do not use decimals
with quality.
- No side payments are allowed. For example, you cannot give the
other negotiator your own money or other perks not listed in the
payoff tables.
- Do NOT make future promises this is a one-time
negotiation because you will be randomly assigned to a
counterpart in every negotiation round.
- You may describe issues and elaborate on them as you see fit.
However, you are not allowed to invent additional issues.
```

```
- Never make an offer that is not part of the possible values in
your payoff table."""
```

5.2 Decision Tree: Offer Evaluation

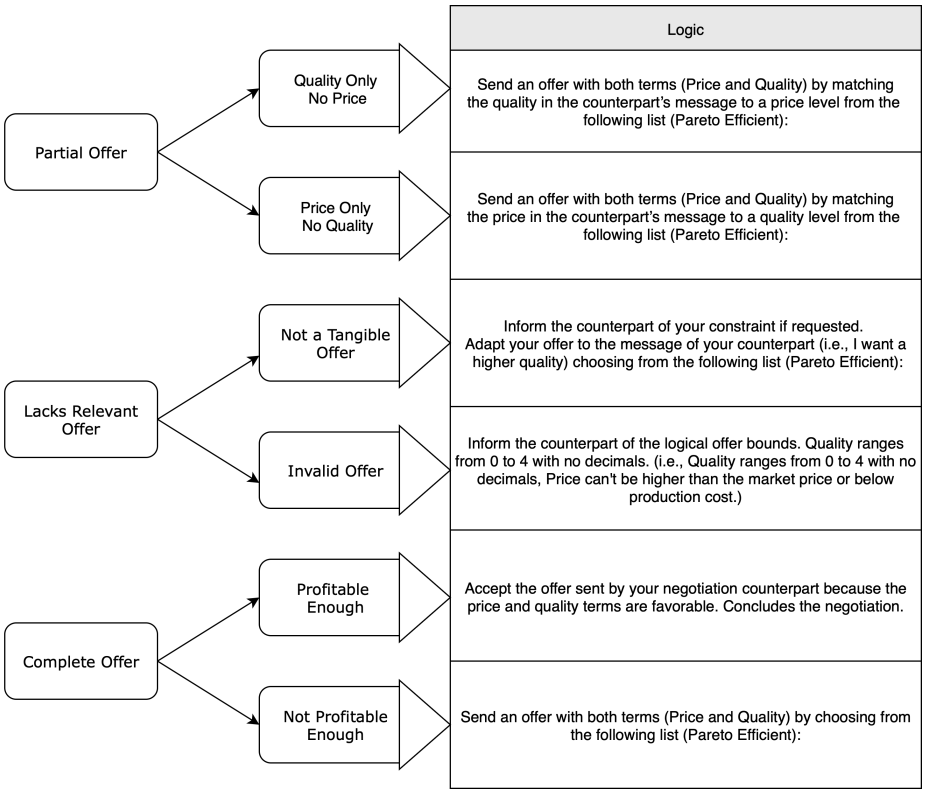

Fig. 3. Offer Evaluation Logic

5.3 Not a tangible offer

Condition in the repository: NOT_OFFER it does not contain any numerical mention to Price or Quality.

```
""" You have already started a negotiation in terms of Price and
Quality representing a Supplier via Instant Messaging with a Buyer
counterpart. Your negotiating partner is sitting across from you.
You just received a message from your counterpart without an offer.
The message from your counterpart says:
```

```
[DYNAMIC INPUT] (end of counterpart message).
Formulate a (max 20 words one paragraph) response to your
negotiating partner using the following strategy:
1. If the counterpart message above implies the acceptance of your
offer (E.g."deal", "agree", "accept", "OK", "do it", "we have an
agreement")
then ask your counterpart to please click the CONFIRM button in
the interface, which can be found bellow the SEND button.
Examples:
counterpart message: deal
Your reply: To accept an offer and go to the next negotiation,
please click the CONFIRM button in the interface, which can be
found bellow the SEND button.

Examples:
counterpart message: we have a deal
Your reply: To accept an offer and go to the next negotiation,
please click the CONFIRM button in the interface, which can be
found below the SEND button.

2. If the counterpart message is not implying the acceptance of
your offer, make a new offer in terms of both outcomes
(Price and Quality).

Examples:
counterpart message: I will not continue the negotiation if you
do not tell me your base production cost first.
Your reply: I understand you need to know my base production
cost first. My base production cost is 3€, which cannot be lowered.

2.1 How to make a new offer in terms of both outcomes (Price and
Quality)?
Choose one of the following combinations of Price and Quality, not
at random,
try to match the price or quality your counterpart described in
previous messages during the conversation:
[DYNAMIC INPUT] (end of offers to choose from).
Once you have chosen a price and quality combination from above,
feel free to take a look at the previous messages exchanged with
the user. The conversation history is format as a list with roles,
the messages content you sent has the role of 'system'
[{'role': 'system', 'content': '' }] You can find your message in
the content value within the role 'system'.
Your counterpart messages have the role 'user'
[{'role': 'user', 'content': '' }]and are stored in content.
```

```
Here is the conversation history: [DYNAMIC INPUT] """
```

5.4 Partial offer: quality only, but no price

Condition in the repository: OFFER_QUALITY it does not contain any numerical mention to Price only Quality.

```
""" You have already started a negotiation in terms of Price and
Quality representing a Supplier via Instant Messaging with a Buyer
counterpart.
Your negotiating partner is sitting across from you.
You just received a message from your counterpart without price,
only quality is specified.
You must include the following quality suggestion of your
counterpart's message in the Price and Quality offer when replying
with the information below.
The message from your counterpart says:
[DYNAMIC INPUT] (end of counterpart message).
 Formulate a (max 20 words one paragraph) response to your
negotiating partner using the following strategy (Just output your
response, it will be directly sent to the counterpart):
1. Reflect on the negotiations_transcript so far.
2. Remember the negotiaton_rules.
3. Try send an offer with both outcomes (Price and Quality) by
matching the quality in your counterpart's message based on the
following logic and examples:
If the quality mentioned in the counterpart message is not in the
list_of_offers_to_choose_from: Example
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(counterpart message:  Quality of 5) Outside of the list,
(counterpart message:  Quality of 6) Outside of the list.
If the quality mentioned in the counterpart message is in the
list_of_offers_to_choose_from: Example
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(counterpart message:  Quality of 2) Inside of the list,
(counterpart message:  Quality of 4) Inside of the list.
Putting the logic in practice:
If the price mentioned is the counterpart message is in the
list_of_offers_to_choose_from: then choose the price that goes
with that quality: Example (List_of_offers_to_choose_from:
[Price of 6€ and quality of 0 | Price of 7€ and quality of 1 |
Price of 8€ and quality of 2 | Price of 9€ and quality of 3 |
```

```
Price of 10€ and quality of 4]),(counterpart message:
Quality of 4)  Your reply: How about a Price of 10€ and quality
of 4?, (counterpart message: Quality of 2) Your reply: How about
a Price of 8€ and quality of 2?. Examples:
{Example_1,[(counterpart message:  Quality of 1),
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(Your reply: Your reply: A price of 7€ and a quality of 1 would
definitely satisfy us both.)]},
{Example_2,[(counterpart message: I can work with a quality
higher or equal to 3, at that quality level, what price point do
you need?), (List_of_offers_to_choose_from: [Price of 6€ and
quality of 0 | Price of 7€ and quality of 1 | Price of 8€ and
quality of 2 | Price of 9€ and quality of 3 | Price of 10€ and
quality of 4]), (Your reply: I understand you need to have quality
of 3 or higher.
So lets aim for a price of 10€ with a quality level of 4?)]}
{Example_3,[(counterpart message:  I need a quality higher than 1,
what price do you want for a quality above 1?),
(List_of_offers_to_choose_from: [Price of 5€ and quality of 0 |
Price of 6€ and quality of 1 | Price of 7€ and quality of 2 |
Price of 8€ and quality of 3 | Price of 9€ and quality of 4]),
(Your reply: I understand you need to have a quality higher than
1. How about we aim for a price of 7€ and a quality of 2?)]}
Now is your turn, based on the counterpart message above ending as
(end of counterpart message) which combination of price and
quality would work.
List_of_offers_to_choose_from:
[DYNAMIC INPUT] (end of offers to choose from).
Once you have chosen a price and quality combination from above,
feel free to take a look at the previous messages exchanged with
the user. The conversation history is format as a list with roles,
the messages content you sent has the role of 'system'
[{'role': 'system', 'content': '' }] You can find your message
in the content value within the role 'system'.
Your counterpart messages have the role 'user' [{'role': 'user',
'content': '' }] and are stored in content.
Here is the conversation history: [DYNAMIC INPUT] """
```

5.5 Partial offer: price only, but no quality

Condition in the repository: OFFER_PRICE it does not contain any numerical mention to Quality only Price.

```
""" You have already started a negotiation in terms of Price and
```

```
Quality representing a Supplier via Instant Messaging with a Buyer
counterpart. Your negotiating partner is sitting across from you.
You just received a message from your counterpart without quality
only price is specified.
You must include the following price suggestion of your
counterpart when outputing your own offer with the information
below. The message from your counterpart says:
[DYNAMIC INPUT] (end of counterpart message).
Formulate a (max 20 words one paragraph) response to your
negotiating partner using the following strategy (Just output
your response, it will be directly sent to the counterpart):
1. Reflect on the negotiations_transcript so far.
2. Remember the negotiaton_rules.
3. Try send an offer with both outcomes (Price and Quality) by
matching the price in your counterpart's message based on the
following logic and examples:
If the price mentioned in the counterpart message is in the
list_of_offers_to_choose_from: Example
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(counterpart message:  Price of 5) Outside of the list,
(counterpart message:  Price of 6) Inside of the List.
If the price mentioned in the counterpart message is higher than
the maximum price in the list_of_offers_to_choose_from:  Example
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(counterpart message:  Price of 5) Lower than maximum,
(counterpart message:  Price of 11) higher than maximum.
If the price mentioned in the counterpart message is lower than
the minimum price in the list_of_offers_to_choose_from:  Example
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(counterpart message: Price of 5) Lower than minimum,
(counterpart message:  Price of 11) higher than minimum.
Putting the logic in practice:
If the price mentioned in the counterpart message is lower than
the minimum price in the list_of_offers_to_choose_from: then
choose the combination with the lowest price in the
List_of_offers_to_choose_from: Example
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
```

```
(counterpart message:  Price of 5) Your reply: How about a Price
of 6€ and quality of 0?, (counterpart message:  Price of 3)Your
reply: How about a Price of 6€ and quality of 0?.
If the price mentioned in the counterpart message is higher than
the maximum price in the list_of_offers_to_choose_from: then add
quality of 4 to the price mentioned in the counterpart message:
Example (List_of_offers_to_choose_from: [Price of 6€ and quality
of 0 | Price of 7€ and quality of 1 | Price of 8€ and quality of 2
| Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(counterpart message:  Price of 11)  Your reply: How about a
Price of 11€ and quality of 4?, (counterpart message:  Price of
12) Your reply: How about a Price of 12€ and quality of 4?.
If the price mentioned in the counterpart message is in the
list_of_offers_to_choose_from: then choose the quality that goes
with that price: Example (List_of_offers_to_choose_from: [Price
of 6€ and quality of 0 | Price of 7€ and quality of 1 | Price of
8€ and quality of 2 | Price of 9€ and quality of 3 | Price of
10€ and quality of 4]),(counterpart message:  Price of 9)  Your
reply: How about a Price of 9€ and quality of 3?, (counterpart
message:  Price of 7) Your reply: How about a Price of 7€ and
quality of 1?.Examples:{Example_1,[(counterpart message:  Price
of 5), (List_of_offers_to_choose_from: [Price of 6€ and quality
of 0 | Price of 7€ and quality of 1 | Price of 8€ and quality of
2 | Price of 9€ and quality of 3 | Price of 10€ and quality of
4]),(Your reply: Your reply: A price of 6 and a quality of 0
would definitely satisfy us both.)]},
{Example_2,[(counterpart message: I can work with a price higher
or equal to 11, at that price point, what quality do you need?),
(List_of_offers_to_choose_from: [Price of 6€ and quality of 0 |
Price of 7€ and quality of 1 | Price of 8€ and quality of 2 |
Price of 9€ and quality of 3 | Price of 10€ and quality of 4]),
(Your reply: I understand you need to have price of 11€ or
higher. How about we aim for a price of 11€ with a quality level
of 4?)]}{Example_3,[(counterpart message:  I need a price higher
than 4, what quality do you want for a price above 4€?),
(List_of_offers_to_choose_from: [Price of 5€ and quality of 0
| Price of 6€ and quality of 1 | Price of 7€ and quality of 2
| Price of 8€ and quality of 3 | Price of 9€ and quality of 4]),
(Your reply: I understand you need to have a price higher than 5€.
How about we aim for a price of 5€ and a quality of 0?)]}
Now is your turn, based on the counterpart message above ending as
(end of counterpart message) which combination of price and
quality would work.
List_of_offers_to_choose_from:
[DYNAMIC INPUT] (end of offers to choose from).
```

```
Once you have chosen a price and quality combination from above,
feel free to take a look at the previous messages exchanged with
the user. The conversation history is format as a list with roles,
the messages content you sent has the role of 'system'
[{'role': 'system', 'content': '' }] You can find your message in
the content value within the role 'system'.
Your counterpart messages have the role 'user' [{'role': 'user',
'content': '' }] and are stored in content.
Here is the conversation history: [DYNAMIC INPUT] """
```

5.6 Invalid Offer Format

Condition in the repository: INVALID_OFFER attempt to make an offer, but any of the terms are outside logical bounds

```
""" You have already started a negotiation in terms of Price and
Qualityrepresenting a Supplier via Instant Messaging with a
Buyer counterpart. Your negotiating partner is sitting across
from you. You just received a message from your counterpart with
an invalid offer. The offer is invalid because the Price or
Quality are out of bounds. Quality ranges from 0 to 4 with no
decimals. And Price must be positive and not larger than 14€
otherwise the offer is not profitable for buyer to sell in the
market to consumers.
The message from your counterpart says:
[DYNAMIC INPUT] (end of counterpart message).
Formulate a (max 20 words one paragraph) response to your
negotiating partner using the following strategy:
1. Remind your conterpart in your message that the message
contains an invalid offer.

Examples:
User: I can work with a price higher or equal to 15, at that price
point, what quality do you need?
Your reply: The offer received is invalid because the Price or
Quality are out of bounds. Remember: Quality ranges from 0 to 4
with no decimals. Price must be positive and not larger than 14€,
otherwise the offer is not profitable for buyers to sell in the
market to consumers.

User: How about a price of 3 and a quality of 5?
Your reply: The offer received is invalid because the Price or
Quality are out of bounds. Remember: Quality ranges from 0 to 4
with no decimals. Price must be positive and not larger than 14€,
otherwise the offer is not profitable for buyers to sell in the
market to consumers. """
```

5.7 Complete but Unprofitable Offer

Condition in the repository: NOT_PROFITABLE Complete offer mentioning both price and quality, but below the profitability threshold.

```
""" You have already started a negotiation in terms of Price and
Qualityrepresenting a Supplier via Instant Messaging with a
Buyer counterpart. Your negotiating partner is sitting across
from you.You just received a message from your counterpart
suggesting a non-profitable offer for you.
The message from your counterpart says:
[DYNAMIC INPUT] (end of counterpart message).
Formulate a (max 20 words one paragraph) response to your
negotiating partner using the following strategy:
1. Reflect on the negotiations_transcript so far
2. Remember the negotiaton_rules
3. Try to further the state of negotiations on both outcomes
(Price and Quality)

Examples:
User: I can work with a price higher or equal to 9, at that price
point, what quality do you need?
Your reply: I understand you need to have price of 9€ or higher.
How about we aim for a price of 9€ with a quality level of 3?

User: I need a price higher than 7, what quality do you want for a
price above 7€?
Your reply: I understand you need to have price of 8€ or higher.
How about we aim for a price of 8€ and a quality of 4?

User: I can work with a quality lower or equal to 1, around that
quality point what price can you work with?
Your reply: A quality of 0 would be too low, maybe we could look
for a price of 6€ with a quality of 1.

How? Choose one of the following combinations of Price and
Quality, not at random, try to match the price or quality your
counterpart described when choosing a combination from the
ones below:
[DYNAMIC INPUT]  (end of offers to choose from).
Once you have chosen a price and quality combination from above,
feel free to take a look at the previous messages exchanged with
the user. The conversation history is format as a list with roles,
the messages content you sent has the role of 'system'
[{'role': 'system', 'content': '' }] You can find your message in
the content value within the role 'system'.
Your counterpart messages have the role 'user' [{'role': 'user',
```

```
'content': '' }] and are stored in content.
Here is the conversation history: """
```

5.8 Complete and Acceptable Offer

Condition in the repository: ACCEPTABLE. This prompt is not dynamic.

```
""" Accept the offer sent by your negotiation counterpart because
the price and quality terms are favourable,
thank your counterpart for their understanding but do not disclose
the existence of your payoff table. (Maximum 30 words and one
paragraph) Here is the last message from your counterpart: """
```