CE-015

大規模言語モデルを適用した交渉自動対話システムへのマネージャーの導入
Introduction of Managers to Negotiation Dialogue System with Large Language Models

渡邉 賢[1)]　藤田 桂英[2)]
Ken Watanabe　Katsuhide Fujita

1　はじめに

交渉は，日常生活から国際政治まで，社会生活の様々な場面で必要となる重要な活動である．交渉の場面において，通常，交渉者は利益を最大化しようとする．しかし，互いの利益を最大化できるような合理的な合意を得ることは難しいと考えられている．理由としては，交渉問題が複雑であること，交渉者の感情や社会背景が交渉に影響を及ぼす可能性があることなどが挙げられる．交渉者が合理的な合意を得るために，人間に変わって自動交渉を行うマルチエージェントシステムの研究が行われている．人間と自動交渉エージェントの間で交渉を行う場合には，人間がエージェントの提案内容を理解する必要があるため，提案内容の決定とともに，適切なテキスト生成を行う必要がある．この自然言語で交渉を行う自動交渉エージェントを交渉自動対話システムと呼ぶ．近年では，ニューラルネットワークベースの自然言語処理の発展により，テキスト生成と推論の両方を制御できるエンドツーエンド交渉エージェントモデルが開発されている [1, 2, 3]．これらのモデルは，場合によっては人間よりも合理的な合意形成を行っている．

また，最近の自然言語処理の分野に目を向けると，2022 年に OpenAI が ChatGPT を発表 [4] すると，それに続くように，Google が Bard[5] を，Meta が LLaMa[6] をそれぞれ発表した．これらは，大規模言語モデル (Large Language Model, LLM) と呼ばれる [7]．LLM は，大規模なパラメータ数を持つモデルを大量のテキストデータと大量の計算機リソースを用いて学習を行うことで，以前よりも大幅に高い性能を出す自然言語処理モデルである．LLM は，様々なタスクに対して，これまでのモデルを超える性能を出すことが報告されている [7]．

交渉についても，LLM を応用することは考えられているが，何も手を加えずに LLM に交渉させるのでは，十分な交渉結果を得られないことが報告されている [8]．また，交渉を含む様々なタスク指向自動対話システムに対して，LLM を応用するような研究は既に存在している [9]．しかし，その性能は人間同士の交渉と比較して劣る結果となっている．このことから，LLM を交渉自動対話システムに適用して十分な性能を得るには，交渉に特化した手法が必要であると考えられる．

本研究では，交渉自動対話システムに LLM を適用し，交渉における行動選択を制御するマネージャーを導入する．本研究では，マネージャーは，相手の入力した文をもとに，両者にとって利益があるような案をヒューリスティックにより導出し，その案を提案するよう LLM に指示する．LLM に対してファインチューニングなどを実施せずに交渉を行わせた場合に，システムとして自然な発話を行えることを示すとともに，マネージャーを導入することにより，導入したシステムが導入していないシステムよりも利益の最大化と合意形成のしやすさの点で優位であることを示す．

また，LLM に対して交渉対話データセットを用いてファインチューニングを実施する．ファインチューニングを実施することにより，交渉の状況に沿った適切な発話ができるようになることを示す．

本論文の構成を以下に示す．まず，第 2 章では，交渉支援の関連研究として，自動交渉マルチエージェント，交渉対話システム，交渉対話データセットなどについて述べる．また，使用する自然言語処理の手法について述べる．第 3 章では，交渉自動対話システムが目的とするタスクの定義を行うとともに，提案手法について述べる．第 4 章では，実験に使用する交渉自動対話システムや実験方法，そして実験結果と考察について述べる．第 5 章では，本研究のまとめと今後の課題を示す．

2　関連研究

2.1　自動交渉

2.1.1　交渉のための自動対話システム

かつては，交渉対話システムの構築には戦略的側面に焦点が当てられていた [10]．それに対して，近年では，エンドツーエンドのニューラルモデルによりテキスト生成と推論の両方を制御する交渉自動対話システムが提案された．Lewis ら [1] は，sequence-to-sequence の尤度モデルを用いて，相手のエージェントの入力を受けて完全な対話を生成するようにした．また，効用を最大化するために，教師あり学習での事前学習に加えて，強化学習で評価指標に照らし合わせて微調整を行った．さらに，発話を行う際に，その発話の報酬を推定するようにした．これらにより，人間と同様またはそれ以上の効用を獲得できた．

He ら [2] は，戦略と生成を切り離したモジュール化された対話行為ベースのアプローチを提案した．この手法により，教師あり学習，強化学習，ドメイン固有知識を用いて，柔軟に戦略を設定できるようになったと共に，検索ベースの文章生成により，文脈を考慮した多様な発話を生成できることが示された．Cheng ら [3] は，敵対的なエージェントを用いた攻撃により，対話エージェントのロバスト性を評価するアルゴリズムを開発した．さらに，敵対的なエージェントの攻撃を用いた敵対的学習により，Lewis らが提案したような目標指向型対話システムのロバスト性が大幅に改善されることが示された．

2.1.2　交渉対話データセット

2.1.1 節で示した Lewis らの研究 [1] では，DealOrNoDeal という，戦略的対話をモデル化するための交渉対話コーパスも提案された．このコーパスは，二者間で三種のアイテム (本，帽子，ボール) を山分けする Multi-issue bargaining タスクにおけるダイアログデータセットである．データセットの詳細なデータは表 1 の通りである．

1）　東京農工大学大学院工学府知能情報システム工学専攻
2）　東京農工大学グローバルイノベーション研究院

表 1　DealOrNoDeal の数的データ (文献 [1] より引用)

指標	数
ダイアログ数	5808
1 ダイアログあたりのターン数平均	6.6
1 ターンあたりの単語数平均	7.6
合意に至ったダイアログの割合 (%)	80.1
平均スコア	6.0
パレート最適解に至ったダイアログの割合 (%)	76.9

その他の交渉対話データセットとして，He ら [2] によって提案された，価格交渉対話コーパス，山口ら [11] によって提案された，転職エージェントと被雇用者による二者間転職面談の交渉対話コーパス，Konovalov ら [12] が提案した雇用条件に関連した人間とエージェントの間の二者間交渉コーパス，Petukhova ら [13] が提案した政治議論対話コーパス，Asher ら [14] が提案したゲーム「Settlers of Catan」における多国間交渉対話コーパスなどがある．

2.2　最近の自然言語処理

2.2.1　言語モデルと大規模言語モデル

言語モデルとは，次に来る単語やマスクされた単語の確率を予測するために，与えられた文章を単語の連続と捉え，その尤度を計算することで人間が使用する言語 (自然言語) をモデル化したものである．

OpenAI は，事前学習済み言語モデル (Pre-trained Language Models, PLM) のパラメータ数や，事前学習に用いるデータセットのサイズ，学習に使用される計算量が増えるに連れて，損失関数 (cross-entropy loss) がべき乗則に従って減ること，つまりモデルの性能が指数的に向上することを発見した [15]．これをスケーリング則と呼ぶ．スケーリング則に従って大規模化された PLM を，大規模言語モデル (Large Language Models, LLM) と呼び [7]，GPT-3(OpenAI, 1750 億パラメータ)[16] や PaLM(Google, 5400 億パラメータ)[17] などが発表された．

Meta は，RMSNorm[18] を使用して事前正規化を適用し，活性化関数 SwiGLU[19] と回転位置埋め込み [20] を使用した LLaMa を 2023 年 2 月に発表 [6] すると，2023 年 7 月にはコンテキスト長の増加とグループ化クエリアテンション (GQA) の採用という変更を加えた Llama 2 を発表した [21]．Llama 2 は，一番大きい 700 億パラメータのモデルでも GPT-3 の半分以下のパラメータ数であるにもかかわらず，後継モデルである GPT-3.5 に並ぶ性能を示している (GPT-3.5 のパラメータ数は非公開)．ただし，PaLM の後継モデルである PaLM 2[22] や，GPT-3.5 の後継モデルである GPT-4[23] には性能に差をつけられている．Llama 2 のその他の特徴として，オープンソースで公開された LLM であることが挙げられ，Meta が公開したモデルに対して GPTQ[24] という技術を適用することでモデルのサイズを小さくしたモデルが有志により公開 [25] されている．

2.2.2　ファインチューニングと Low Rank Adaptation

ファインチューニングとは，学習済みのモデルを用いて，モデルの内部状態を微調整して新しいタスクに対応させることである [26]．ファインチューニングにより，モデルのパラメータがある程度近い状態から学習できるため，未学習のモデルを学習させるよりも学習時間を短縮できるほか，新しいタスク用のデータ件数が少ないときでもより高い精度を出すことができるといった特徴がある．

LLM のような，パラメータ数がとても多いモデルは，全てのパラメータに対してファインチューニングを実施するのは現実的ではない．そこで，事前学習されたパラメータの内部状態は固定したまま，Transformer アーキテクチャの各層に学習可能なランク分解行列を注入することで，下流タスクのために学習可能なパラメータの数を大幅に削減する手法として，Low-Rank Adaptation(LoRA) が Hu らにより提案された [27].

3　LLM を適用した交渉自動対話システム

3.1　タスク設定

交渉自動対話システムのタスクとして，Lewis らの研究 [1] で提案された採用されている Multi-issue bargaining タスクを採用した．二者間での交渉を通して，三種のアイテム (本，帽子，ボール) を山分けするというタスクである．本タスクは，交渉問題の解析や理論の構築のための多くの既存研究 [2, 8, 28, 29] で取り上げられている．以下，タスクの詳細を示す．

二人のエージェントに同じアイテムの集合を見せ，各アイテムがどちらかのエージェントに割り当てられるように山分けするように指示する．各エージェントには，各アイテムに非負の値を与える，異なるランダムに生成された価値関数が与えられる．価値関数は，以下のように制約されている．

1. すべてのアイテムの価値の合計は 10 である
2. 各アイテムは少なくとも一人のエージェントにとって 0 でない価値を持つ
3. いくつかのアイテムは両方のエージェントにとって 0 でない価値を持つ

これらの制約により，両方のエージェントが獲得する価値を最大化することは不可能である．また，両方のエージェントにとって価値が 0 であるアイテムは存在しない．10 ターン後には，エージェントに「合意なし」で交渉を終えるという選択肢を与える．それを選択した場合は，両方のエージェントの得点は 0 点になる．

3.2　LLM を適用した交渉自動対話システム

本研究では，交渉自動対話システムに LLM を適用する．また，交渉を行う LLM に対してとるべき行動を指示するマネージャーを導入する．提案するシステムは，相手の入力からマネージャーに必要な情報を取り出すパーサー，とるべき行動の指示を行うマネージャー，指示を受けて応答を生成するジェネレーターからなり，ジェネレーターに LLM を適用する．ベースとする LLM は，Meta が公開した Llama 2[6] のパラメータ数が 130 億のモデルを対話用に微調整したモデルに対して，GPTQ を用いて 4bit 量子化したモデルである Llama-2-13B-chat-GPTQ[25] を使用する．また，マネージャーは，その効果を確認するために，ヒューリスティックに基づき行動を指示する単純なものとした．提案する交渉自動対話システムを用いた交渉の流れを図 1 に示す．ユーザーの入力をシステムが受け取ると，パーサーに入力が送信される (図 1 中の橙色の矢印線)．パーサーはユーザーの入力から正規表現でユーザーの要求を取り出し，その情報をマネージャーに送信する (図 1

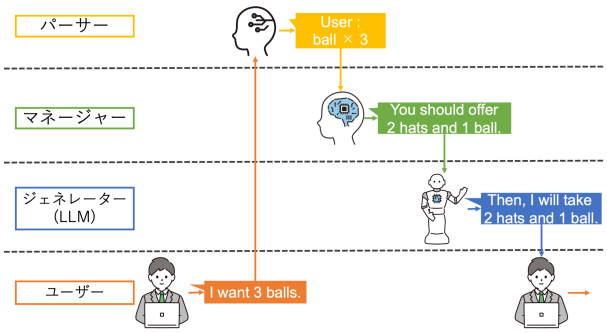

図 1　交渉自動対話システムによる交渉の流れ．ユーザーの入力をシステムが受け取ると，パーサーに入力が送信される (橙色の矢印線)．パーサーはユーザーの入力から正規表現でユーザーの要求を取り出し，その情報をマネージャーに送信する (図中の黄色の線)．マネージャーが入力を受け取ると，それを受けて行動の指示をジェネレーターに送信する (青色の矢印線)．ジェネレーターはユーザーの入力とマネージャーの指示を受け取り，応答を生成してユーザーに送信する (緑色の矢印線)．

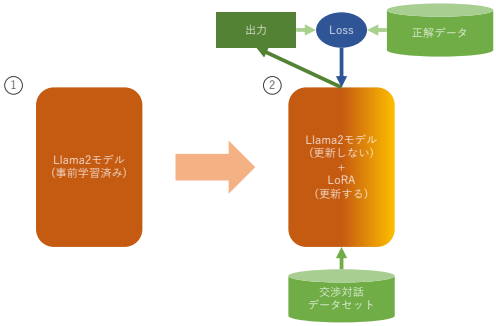

図 2　Llama 2 をファインチューニングする際の流れ．まず，事前学習済みの Llama 2 モデルを用意する (①の部分)．そこに，LoRA を注入する．そして，交渉対話データセット (本研究では DealOrNoDeal) を入力して，その出力とデータセットの応答を元に損失関数の値を計算し，それを逆伝播することでモデルの内部状態を更新する (②の部分)．ただし，内部状態の更新は LoRA のみ行い，Llama 2 本体は行わない．

中の黄色の線)．マネージャーが入力を受け取ると，それを受けて行動の指示をジェネレーター (本研究では Llama 2) に送信する (図 1 中の青色の矢印線)．ジェネレーターはユーザーの入力とマネージャーの指示を受け取り，応答を生成してユーザーに送信する (図 1 中の緑色の矢印線)．

また，交渉を行う LLM にファインチューニングを実施することにより，システムの発話が状況に沿ったものになるかどうかを確認する．ファインチューニングには，Lewis らが提案した交渉対話データセットである DealOrNoDeal[1] を使用する．量子化モデルは全てのパラメーターをファインチューニングする手法は使用できないため，LoRA[27] を用いてのファインチューニングを実施する．ファインチューニングの流れは，図 2 の通りである．まず，事前学習済みの Llama 2 モデルを用意する (①の部分)．そこに，LoRA を注入する．続いて，交渉対話データセット (本研究では DealOrNoDeal) を入力して，その出力とデータセットの応答を元に損失関数の値を計算し，それを逆伝播することでモデルの内部状態を更新する (②の部分)．ただし，内部状態の更新は LoRA のみ行い，Llama 2 本体は行わない．そして，学習が完了すると，LoRA を更新した Llama 2 モデルを得ることができる．

本研究で実装したマネージャーの行動

ここで，本研究で実装したマネージャーがどのような行動を取るかについて説明する．マネージャーの行動は以下の通りである．

- アイテムのポイントと個数が与えられたら，アイテムの組み合わせとその組み合わせで獲得できるポイントを網羅し，それを獲得できるポイントが多い順に並べて保持する．
- 自分が先に発言する場合には，保持している情報から，最も多くのポイントを獲得できる組み合わせを提案するように指示する．
- 相手の要求から，相手にとってアイテムのポイントを以下の手順で推定する．
  1. 相手が要求してきたアイテムの合計数を計算する．ここで，提示されている数よりも相手が少なく要求しているアイテムは，そのことを記録した上で，提示されている数を合計数に足し合わせる．
  2. 相手が要求してきたアイテムは獲得可能な最大ポイントである 10 ポイントを合計数で割った値を，要求していないアイテムは 0 を，それぞれ仮の予測ポイントとする．
  3. 提示されている数よりも相手が少なく要求しているアイテムは，仮の予測ポイントから 1 ポイント引く．これは，全てを要求しないアイテムは，相手にとって重要度が低いと予想できると考えたためである．
  4. 最後に，現在の仮の予想ポイントをもとにした獲得可能な最大ポイントを計算し，それと 10 の差分を，相手が全数を要求しているアイテムの数で割ったうえで，全数を要求しているアイテムの予測ポイントに足し合わせる．
- 相手がしてきた要求について，自分の獲得ポイントが 8 ポイントを超える場合には，その要求を受け入れるように指示する．
- そうでない場合には，相手の要求から自身の獲得するアイテム数を増やしながら，自分の獲得できるポイントと相手が獲得できると予想されるポイントを計算し，両者のポイントの差が小さくかつ自身の獲得ポイントが多いような組み合わせを提案するよう指示する．

要求を提示するよう指示する場合には，

“Offer that you want 1 book and 2 balls because you can get 10 points”

といった文章をジェネレーターに送る．また，相手の要求を受け入れる場合には，

“You should accept partner’s offer because you can get 9 points”

といった文章をジェネレーターに送る．なお，例文中のアイテムの数やポイントは状況によって変化する．

4　評価実験

4.1　実験設定

4.1.1　実験に使用するシステム

実験には，表 2 に記載した 4 種類の交渉自動対話システムを用いる．また，比較対象として，Lewis らが提案した GRU ベースの sequence-to-sequence モデル [1] を使用したシステム (GRU-System) も用いた．

なお, マネージャーを導入していない Llama2-System$_{\text{pln}}$ と Llama2-System$_{\text{sft}}$ については，ジェネレーターの入力はユーザーの入力文のみとなるため，マネージャーを導入している図 1 とは異なり，交渉相手の入力文は直接ジェネレーターに送信される．

ジェネレーターの設定: ファインチューニング実施なし

ジェネレーターとして，Llama-2-13B-chat-GPTQ[25] をファインチューニング等を実施せずに用いる．交渉を始める前に，

“You are a negotiator. Negotiate with the other party and divide the items. Here is the items and points you get if you get the item for this negotiation: There are 2 books and each book is worth 4 points. There is 1 hat and the hat is worth 2 points. There are 3 balls and each ball is worth 0 point. The worth of the item is different for the other party. Do not state the worth of the items to you in the negotiation. In the negotiation, show your offer that contains items which you can earn as many points as possible. Be careful not to make a mistake with the number of items.”

という文章を入力することで，モデルにアイテムのポイントと個数や交渉の状況設定，とるべき行動を指示する．なお，上記の例文中に含まれるアイテムのポイントや個数は一例であり，実際の実験では，実験ごとに異なる数字が入る．

応答文の生成の際には，モデルの出力に対してサンプリングを実施した．サンプリング手法として，単語の出現確率のうち上位 k 個からサンプリングする Top-k サンプリングと，単語の出現確率について，累積確率が p になる最小の単語集合の中からサンプリングする Top-p サンプリングを組み合わせ，より対象となる単語集合が小さくなる方を選択する方法 [30] を採用した．今回の実験では，$k = 50, p = 0.95$ とした．

ジェネレーターの設定: ファインチューニング実施あり

Llama-2-13B-chat-GPTQ について，各層に LoRA モデルを注入し，ファインチューニングを実施したモデルを用いる．ファインチューニングの際には，DealOrNoDeal を学習用データセットとして用いた．データセットのうち，80%を訓練データ，10%を検証データ，10%をテストデータとした．また，LoRA モデルのハイパーパラメータについては $r = 16, \alpha = 32, dropout = 0.05$ とした．学習時のハイパーパラメータは，バッチサイズを 4，学習率を 0.00001 とし，5 エポック学習を行なった．各エポックでのモデルを保存し，事前に生成される文章を確認した結果，3 エポック時点でのモデルの生成文が，マネージャーの指示を反映しつつ違和感のない応答をしていたため，評価対象のモデルとした．それぞれのシステム用にファインチューニングを行なった際の損失関数の変化は，Llama2-System$_{\text{sft}}$ 用のものが図 3，Llama2-System$_{\text{mng\&sft}}$ 用のものが図 4 の通りである．図 3 及び図 4 から共通してわかることとして，どちらも 3 エポック学習時点で一度学習時の損失関数の値と検証時の損失関数の値が近づいている．このことからも，3 エポック時点で学習が十分に行われた状態になっていることが考えられる．

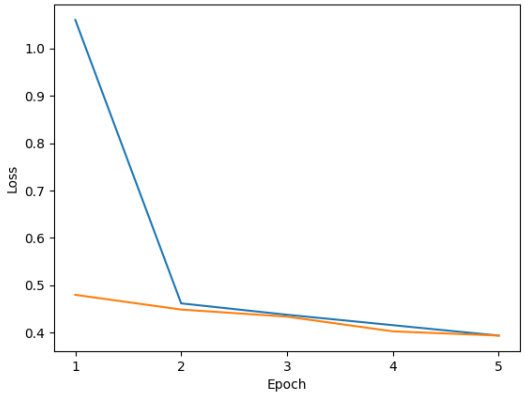

図 3　Llama2-System$_{\text{sft}}$ 用にファインチューニングを実施した際の loss の変化

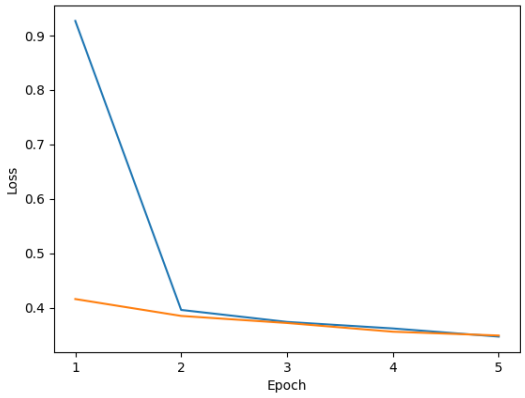

図 4　Llama2-System$_{\text{mng\&sft}}$ 用にファインチューニングを実施した際の loss の変化

また，こちらについても，交渉を始める前の文章や出力のサンプリングについては Llama2-System$_{\text{pln}}$ 及び Llama2-System$_{\text{mng}}$ と同様のものを用いた．

ジェネレーターの設定: GRU-System

LLM を導入することの効果を確認するために，比較対象として，Lewis ら [1] が提案した GRU ベースの sequence-to-sequence モデルを使用する．発話を行う部分は，GRU 一層の出力に線形層を一層つけたシンプルな構造である．そのほか，アイテムのポイントや個数といった情報を処理する部分などがあり，全体のパラメータ数は，557869 である．

4.1.2　評価指標

定量評価

システムの定量評価の指標として，交渉によって獲得したポイントを用いる．被験者と交渉を行い，その全ての結果からシステムとユーザー (被験者) の平均獲得ポイント (Score) を計算し，それをシステムの評価とする．また，獲得ポイントが両者とも 0 になった割合，つまり合意に至らなかった割合から，合意率 (合意に至った割

表 2　実験に使用したシステム

システム名	マネージャー導入	ファインチューニング実施
Llama2-System$_{pln}$	×	×
Llama2-System$_{mng}$	○	×
Llama2-System$_{sft}$	×	○
Llama2-System$_{mng\&sft}$	○	○

合，Agr.) や，合意に至った交渉のみを対象とした各平均獲得ポイント (Agr. Score) も計算し，これらもシステムの評価とする．

定性評価

機械翻訳の評価指標であれば BLEU，自動要約の評価指標であれば ROUGE などがあるが，対話システムの定量評価指標として一般的に用いられているものはまだ存在しない．これは，対話の応答に絶対的な正解が存在しないことが理由である．そのため，本研究では，対話システムとしての評価は，人手によって定性的な評価を行う．自然な応答を行えてかつ交渉に用いることができるシステムであることに対する評価として，

- Fluency(Flu, 流暢さ):
  システムの応答は流暢であったか
  (ex. 反復，重複などが発生していないか)
- Consistency(Cns, 一貫性):
  システムの応答は全体を通して一貫していたか
  (ex. 1 つ前の応答との内容の矛盾がないか)
- Correctness(Crr, 正確性):
  システムの応答は状況を正しく反映していたか
  (ex. アイテムの数に過不足はないか)

の三つの観点について．5 点満点で点数をつけてもらう．そして，点数の平均をそのシステムのスコアとする．被験者は 7 人で，それぞれのシステムについて 3 回ずつ交渉を行ってもらった．そのため，各システムの評価数は 21 件となる．なお，交渉するシステムの順番は被験者ごとにランダムとしている．

4.1.3　実験環境

モデルの記述には PyTorch[31] を用いた．なお，バージョンは 2.1.0，CUDA バージョンは 12.1，cuDNN バージョンは 8.9.4 である加えて，Transformer ベースのモデルである Llama 2 は，Transformers[32] を用いてプログラムに組み込んだ．なお．バージョンは 4.35.2 である．また，交渉を行うインタフェースの作成には，Gradio[33] を用いた．なお，バージョンは 4.4.0 である．プログラムの動作環境は，

- CPU: AMD 7950x zen4 16 cores 32 threads
- GPU: RTX 4090 24GB
- Memory: Non-ECC 32GB×4

である．図 5 は，被験者がシステムと交渉を行うインタフェースの画像である．画面左側に表示されたアイテムのポイントと個数をもとに交渉を行い，その履歴が画面右側に表示される．

4.2　実験結果と考察

実験結果は表 3 の通りである．全交渉についてのシステムの平均獲得ポイントは，Llama2-System$_{mng}$ が最も高く，Llama2-System$_{mng\&sft}$ が二番目に高い．また，合意率については，Llama2-System$_{mng\&sft}$ が最も高く，Llama2-System$_{mng}$ が二番目に高い．これらの結果から，

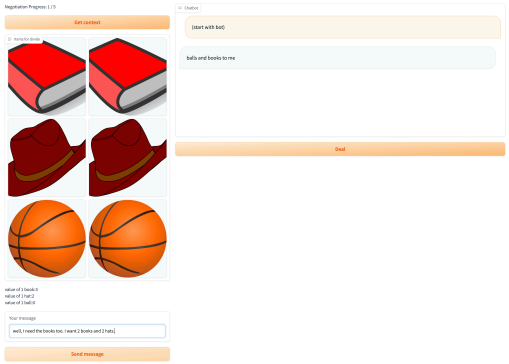

図 5　被験者に提供した交渉を行うインタフェース

マネージャーを導入することにより，導入していないシステムと比べての，獲得ポイントや，合意率の向上に効果があると言える．まず，Llama2-System$_{pln}$ と Llama2-System$_{mng}$ とを比較する．LLM だけでは自分の獲得ポイントを多くするという目的やそのためにとるべき行動について理解できない傾向にあったために，マネージャーを導入してその部分を補強したことの効果が現れたと考えられる．続いて，Llama2-System$_{sft}$ と Llama2-System$_{mng\&sft}$ を比較する．ファインチューニングに用いた学習データセットが GRU-System と同一であるため，マネージャーを導入していない Llama2-System$_{sft}$ と GRU-System は相手の入力に対しての応答が近いものになったと考えられる．その結果，あまり歩み寄りをしなかったために，ユーザーが納得できる合意案を形成できなかったことが多く発生したために合意率が低くなり，合意に至らないと両者の獲得ポイントが 0 ポイントになることから，平均獲得ポイントも少なくなったと考えられる．その点，マネージャーを導入することで，双方が納得できる合意案を探って提案するようにしたことで，Llama2-System$_{mng\&sft}$ は合意率を高く，獲得ポイントも多くすることができたと考えられる．

一方で，合意に至った交渉のみを対象とした平均獲得ポイントは，先行手法である GRU-System が最もシステムの平均獲得ポイントが高い結果となった．ただし，GRU-System の合意に至った割合は，5 つのシステムの中で最も低い．この結果から，GRU-System は，合意に至れば先行手法の方が獲得ポイントが多くなるが，合意に至らない確率が比較的高いため不安定であると言える．このような結果については，マネージャーを導入したシステムが，マネージャーの戦略の一環として，ユーザーに歩み寄って提案を行なったことが，GRU-System と比べて合意に至った交渉のみを対象とした平均獲得ポイントが少なくなった原因と考えられる．獲得ポイントの向上は，合意率の向上とはある種トレードオフの関係でもあるため，この結果が一概に悪いことであるとは言えない．とはいえ，マネージャーが相手の入力を受けてどのような行動をするかを調整することで，合意率を維

表 3　実験結果．System がシステム名，Score がユーザー (Usr，左) とシステム (Sys，右) の平均獲得ポイント，Agr. Score が合意に至った交渉でのユーザー (Usr，左) とシステム (Sys，右) の平均獲得ポイント，Agr. が合意に至った交渉の割合 (この項目のみ%表記)，Flu が Fluency，Cns が Consistency，Crr が Correctness をそれぞれ表している．太字に下線がついているところが，その項目で最も良い成績であることを，下線のみがついているところが，その項目で二番目に良い成績であることを，それぞれ表している．

System	Score	Agr. Score	Agr.	Flu	Cns	Crr
	Usr vs. Sys	Usr vs. Sys	(%)			
Llama2-System$_{\text{pln}}$	7.05 vs. 4.19	7.50 vs. 4.89	85.71	**4.67**	**4.05**	3.33
Llama2-System$_{\text{mng}}$	6.76 vs. **5.33**	7.21 vs. 5.68	**90.48**	4.05	3.43	2.81
Llama2-System$_{\text{sft}}$	4.71 vs. 4.29	7.07 vs. 5.71	66.67	3.57	3.38	3.71
Llama2-System$_{\text{mng\&sft}}$	7.71 vs. 4.67	7.68 vs. 5.16	**90.48**	4.38	3.67	**3.90**
GRU-System	4.10 vs. 4.05	6.62 vs. **6.54**	61.90	3.43	2.76	3.48

持しつつ平均獲得ポイントを向上させることが可能であるかを検証する余地は残されている．

また，Fluency と Consistency について最も高い結果となったシステムは，Llama2-System$_{\text{pln}}$ であった．そして，GRU-System は 5 つのシステムの中で Fluency と Consistency の両方で最も低い結果となった．この結果から，Llama2 をジェネレーターとして導入することによって，システムが行う発話がより流暢に，例えば反復や重複がないものになり，かつ一貫したもの，例えば前後の文との矛盾がないものになると言える．この結果は，GRU-System に使用したモデルと Llama 2 では，パラメータ数が大幅に異なることや，GRU-System の学習データが DealOrNoDeal のみであるのに対して，Llama 2 は膨大なテキストデータで事前学習を行なっていることから，当然の結果であると言える．また，Llama2-System$_{\text{mng}}$ と Llama2-System$_{\text{sft}}$ が，Llama2-System$_{\text{pln}}$ を下回るものの，Llama2-System$_{\text{mng\&sft}}$ は Llama2-System$_{\text{pln}}$ に近づき，二番目に高い値になっている．このことから，マネージャーの導入やファインチューニングの実施は，それぞれ片方のみを実施すると，LLM が本来持つ発話の流暢さや一貫性を損なわせてしまうが，二つを組み合わせることにより，流暢さや一貫性を維持することができると考えられる．この結果について，まず Llama2-System$_{\text{mng}}$ については，マネージャーの指示とそれを受けて生成するべき文章の対応関係が十分に把握できず，生成文の流暢さや一貫性に影響が出てしまったと考えられる．Llama2-System$_{\text{sft}}$ については，学習データから交渉の進め方をうまく学習できず，結果として強引な応答になってしまったため，流暢さや一貫性の評価が下がってしまったと考えられる．その点，Llama2-System$_{\text{mng\&sft}}$ は，マネージャーの指示とそれを受けて生成するべき文章の対応関係が十分に把握でき，かつ交渉の進行方向については指示を受けているため強引な応答にならず，生成文の流暢さや一貫性が改善したと考えられる．

Correctness については，Llama2-System$_{\text{mng\&sft}}$ が最も高く，Llama2-System$_{\text{sft}}$ が二番目に高い結果となった．この結果から，ファインチューニングを実施することにより，システムが置かれている状況をより正確に把握した応答を行えるようになると言える．このような結果になった理由として，アイテムのポイントや個数といった情報を与える文と実際にその情報をもとに生成する文の対応関係について，ファインチューニングを行なっていない LLM では正確に把握できないことが考えられる．LLM は，学習データがない (zero-shot) か少ない (few-shot) 状態でも様々なタスクを実施できることが報告されている [7]．しかし，数量データを扱うタスクはあまり見られず，zero-shot や few-shot で数量データを正確に扱うことについて困難があることが考えられる．Llama2-System$_{\text{pln}}$ と Llama2-System$_{\text{mng}}$ についても，zero-shot であったため，同様の困難により正確性が失われたと考えられる．その点，ファインチューニングを実施した Llama2-System$_{\text{sft}}$ と Llama2-System$_{\text{mng\&sft}}$ では，十分な量の学習データを用いたことにより，入力の数量データと応答の数量データの対応づけを学習できたと考えられる．

それぞれのシステムの応答

Llama2-System$_{\text{pln}}$ の応答は，自身に提示されているポイントを応答に含めながら相手の獲得ポイントも自身に提示されたポイントをもとに計算する傾向にあった．この傾向から，システム側が交渉相手と自分とでアイテムの持つポイントが違うと言うことを理解できていない，ということがわかった．交渉を始める前の指示文の中に交渉相手と自分とでアイテムの持つポイントが違うということは記載したが，実際の交渉の応答文には反映されなかった．また，被験者の入力に対しても，その内容を正確に把握した応答ができていない傾向も確認できた．このことから，Llama2-System$_{\text{pln}}$ は，指示文に記載した前提条件や交渉相手の入力内容を正確に把握できず，交渉において自分の利益を最大化するための適切な行動を取れなかったことが考えられる．

Llama2-System$_{\text{mng}}$ の応答の傾向は，交渉相手の提案内容を正確に理解できていないという Llama2-System$_{\text{pln}}$ と同様の傾向が見られるほか，マネージャーの指示も正確に理解できていないと考えられるものであった．このことから，Llama2-System$_{\text{mng}}$ は，マネージャーの指示について，要求を提示する・相手の要求を受け入れるといった部分については正確に反映できているものの，要求のうち何をいくつ要求するのかといった数値の正確性が不十分である傾向があったために，正確性の部分で評価が低くなったと考えられる．

Llama2-System$_{\text{sft}}$ の応答からは，交渉が合意に至り獲得アイテムを入力する段階に移行することを示す ⟨selection⟩ というトークンを交渉の途中で出力しやすい傾向が見られた．このことから，学習データに多く含まれた ⟨selection⟩ を出力しやすい傾向になってしまったために，流暢さについて評価が低くなったとともに，合意に至ったことを両者が確認する前に ⟨selection⟩ を出力し

てしまったために，合意に至らない交渉が増え，合意率が下がってしまったと考えられる．

Llama2-System$_{mng\&sft}$ は，同じくマネージャーを導入した Llama2-System$_{mng}$ と比較して，相手の入力やマネージャーの指示を正確に理解できていると考えられる応答をしている傾向にあった．このことから，正確性の部分で高い評価が得られたと考えられる．また，Llama2-System$_{sft}$ と比較して，対話の往復数が増え，交渉相手にシステムの意図が伝わりやすい傾向にあった．このことから，流暢さや一貫性について評価が比較的高くなったと考えられる．

GRU-System の応答の傾向として，相手がシステムの要求を受け入れる入力をしているにもかかわらず，再度同じ要求をしたり，要求が相手の要求とは関係なしに突然変化したりしている．このことから，GRU-System は相手の要求や入力そのものについて，その内容を正確に把握できなかったため，発話についての定性評価指標で評価が低くなり，交渉相手の意図を汲んだ交渉できなかった結果，合意率が下がる結果になったと考えられる．

5 まとめと今後の課題

5.1 まとめ

本研究では，LLM を交渉自動対話システムに適用し，行動の選択を行うマネージャーを導入する手法を提案した．LLM は様々なタスクに応用されているが，タスク指向の自動対話システムとして LLM を用いるには，タスクの目的を達成するための工夫が必要であると考えられてる．交渉についても，同様であり，相手の提案を受けて自分がどのように応答するかを制御するための工夫が必要と考えられる．本研究では，その工夫としてマネージャーを導入し，その効果を確認した．また，交渉対話データセットによるファインチューニングも実施し，その効果を確認した．同時に，LLM を適用した効果を確認するために先行手法との比較も行なった．

実験結果より，マネージャーを導入することは，システムが獲得する利益の向上や，合意する割合の向上に対して効果があることが確認できた．そして，LLM をジェネレーターとして導入することで，システムの発話が人間にとって自然に感じられるものになることが確認できた．また，ファインチューニングを LLM に対して実施することは，LLM がジェネレーターとして，自身が置かれている状況を正確に把握した応答をすることに対して効果があることが確認できた．

5.2 今後の課題

本研究では，マネージャーとしてヒューリスティックに要求の指示を出すものを採用したが，強化学習を実施したマネージャーを導入する効果を検証する必要がある．現在，自動交渉の分野では，強化学習を用いた自動交渉エージェントが一般的である，そのため，交渉対話システムのマネージャーとしても，ヒューリスティックを用いたマネージャーよりも強化学習を行なったマネージャーの方が，交渉において高い成果をあげられることが予想できる．

類似の検討事項として，システム同士の交渉による強化学習が挙げられる．先行研究である Lewis らの研究 [1] で実施されていた手法であるが，これにより，マネージャーを導入せずにジェネレーターが交渉でとるべき行動を学習できる可能性が考えられる．ヒューリスティックなマネージャーは，学習コストがかかっていないが，マネージャーの強化学習をする場合には，学習コストがかかるため，マネージャーを強化学習するのとシステムを直接強化学習するので，コストや交渉結果についてどちらが良いのか検証の必要がある．

また，本研究では，交渉対話データセットを用いたファインチューニングのみを行なったが，Chat-GPT で採用されたような，人手での評価による強化学習 (RLHF) を導入することも検討の価値がある．交渉の状況に応じた人手での評価を与えることでも，交渉内容に則した応答を生成できるのかは検証するべきであると考えられる．

参考文献

[1] Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2443–2453, 2017.

[2] He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2333–2343, 2018.

[3] Minhao Cheng, Wei Wei, and Cho-Jui Hsieh. Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3325–3335, 2019.

[4] OpenAI. Introducing chatgpt. `https://openai.com/blog/chatgpt`, 11 2022. (Accessed on 1/30/2024).

[5] Eli Collins Sissie Hsiao. Sign up to try bard from google. `https://blog.google/technology/ai/try-bard/`, 3 2023. (Accessed on 1/30/2024).

[6] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[7] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv e-prints*, pp. arXiv–2303, 2023.

[8] Johannes Schneider, Steffi Haag, and Leona Chandra Kruse. Negotiating with llms: Prompt hacks, skill gaps, and reasoning deficits. *arXiv e-prints*, pp. arXiv–2312, 2023.

[9] Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. *arXiv e-prints*, pp. arXiv–2311, 2023.

[10] Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. Strategic dialogue management via deep reinforcement learning. *CoRR*, Vol. abs/1511.08099, pp. 1–10, 2015.

[11] Atsuki Yamaguchi, Kosui Iwasa, and Katsuhide Fujita. Dialogue act-based breakdown detection in negotiation dialogues. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 745–757, 2021.

[12] Vasily Konovalov, Ron Artstein, Oren Melamud, and Ido Dagan. The negochat corpus of human-agent negotiation dialogues. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3141–3145, 2016.

[13] Volha Petukhova, Christopher Stevens, Harmen de Weerd, Niels Taatgen, Fokie Cnossen, and Andrei Malchanau. Modelling

multi-issue bargaining dialogues: Data collection, annotation design and corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3133–3140, 2016.

[14] Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 2721–2727, 2016.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, Vol. abs/2001.08361, pp. 1–30, 2020.

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, Vol. 24, No. 240, pp. 1–113, 2023.

[18] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., 2019.

[19] Noam Shazeer. GLU variants improve transformer. *CoRR*, Vol. abs/2002.05202, pp. 1–5, 2020.

[20] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, Vol. 568, p. 127063, 2024.

[21] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[22] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv e-prints*, pp. arXiv–2305, 2023.

[23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[24] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[25] TheBloke. Thebloke/llama-2-13b-chat-gptq · hugging face. `https://huggingface.co/TheBloke/Llama-2-13B-chat-GPTQ`, 7 2023. (Accessed on 1/30/2024).

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Proceedings, Part IV*, Vol. 9908 of *Lecture Notes in Computer Science*, pp. 614–629, 2016.

[27] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, Vol. abs/2106.09685, pp. 1–26, 2021.

[28] Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1208–1218, 2019.

[29] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[30] Patrick von Platen. How to generate text: using different decoding methods for language generation with transformers. `https://huggingface.co/blog/how-to-generate`, 3 2020. (Accessed on 01/30/2024).

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pp. 8024–8035, 2019.

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics.

[33] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019.