欺瞞対話コーパスの構築に向けた 人狼ゲームプラットフォームLiCOSの開発

阪本 浩太郎 †1†2　渋木 英潔 †1　森 辰則 †1

†1横浜国立大学　†2国立情報学研究所
E-mail: {sakamoto,shib,mori}@forest.eis.ynu.ac.jp

1 はじめに

人狼ゲームとは，対話を通して「村人」の中に潜伏した「人狼」を見つけ出す対戦型の多人数ゲームであり，近年，人狼知能プロジェクト[1]など研究テーマとしても注目されている.「人狼」となったプレイヤーは，他のプレイヤーに正体を悟られないよう，普通の「村人」のふりをして「偽の推理」を披露したり,「占い師（人狼の正体を知ることができる）」を騙って他のプレイヤーを扇動したりすることで勝利を目指す．また，「村人」であっても「人狼」を焙り出すために，敢えて「村人」と認識しているプレイヤーを「人狼」と疑うような発話をする場合もある．こういった認識している事柄（以降，**本心**と呼ぶ）と異なる認識を他者に与えようとする発話を本研究では**欺瞞**と定義する．我々は，これからの対話システムには，ユーザの命令に唯々諾々と従うのではなく，自発的に対話を進めていくことも必要であると考えており，システムが欺瞞的な発話をすることはその一つに該当すると考えている．本研究における本心は欺瞞によってすり替えられる認識があるもののみに限定していることに注意されたい[2].

欺瞞を扱った対話の従来研究として，文献 [1] などがあげられるが，対話に存在する欺瞞の検出に焦点をあてており，いかに相手を騙すかという話し手の視点からのものではない．検出などの聞き手の視点からは，相手の発話が事実でないことが分かれば十分であるが，話し手の視点からは，自分にとって都合の良い内容を相手に信じてもらうために最適な発話を選択しなければならない．そのような対話を行うための方法論は確立されておらず，欺瞞を含む対話を分析するところから始める必要がある．従来研究で構築された対話コーパス [2, 3] では，発話者は基本的に誠実であり，相手を騙すことを目的とした発話は収録されていない．人狼ゲームに関する従来研究 [4, 5] では，人狼 BBS[3]の発話ログを利用したものが多い．人狼 BBS はオンラインで行う掲示板型人狼ゲームであるが，発話ログには本心が明記されていないため，発話者の役割や投票などの行動から推測するしかない．それゆえ，発話と本心が対応付けられた欺瞞対話コーパスが望まれる．

欺瞞対話コーパスを構築するために，欺瞞を発話する際にその時の本心を記録してもらうという方法も考えられるが，自然な対話環境とは言い難い上に構築コストも高い．低コストで大規模に知識を獲得する方法として，ゲームを通した知識獲得 [6, 7] がある．もしも，人狼ゲームにおける発話の際の本心に関するデータを対話を妨げることなく自動的に収集できれば，多くの人々に遊んでもらいながら容易に欺瞞対話コーパスの構築が可能になる．以上の背景から，我々は，人狼ゲームにおける欺瞞対話観測システム LiCOS(**Li**ar's **C**onversation **O**bserving **S**ystem) の開発を行っている．本稿では，人狼ゲームにおける欺瞞と本心について考察した後，LiCOS のアーキテクチャについて説明する．

2 人狼ゲームにおける欺瞞と本心

人狼ゲームは『村人』[4]と『人狼』の２つの陣営に分かれて戦うチーム戦である.『村人』側は，正体を隠している全ての「人狼」を見つけて処刑する（ゲームから脱落させる）ことで勝利となり,『人狼』側は，毎晩「人狼」以外のプレイヤーを襲撃して（ゲームから脱落させて）いき,「人狼」の数が「人狼」以外の数以上になることで勝利となる．必然，ゲーム中の対話も上記の勝利条件を達成するためということが前提に行われる[5]．そのため，人狼ゲームにおける欺瞞の例としては，

- 正体を隠したい「狩人（自分以外に対する人狼の

[1]http://aiwolf.org/
[2]単純な「ゲームに勝ちたい」や「処刑されたくない」などの思いは本研究で扱う本心ではない．
[3]http://ninjinix.com/
[4]プレイヤーとしての「村人」や「人狼」と区別するために，陣営の場合は『』で表すこととする．
[5]ゲーム中の雑談は禁止されていないが，プレイヤーの中には「彼は雑談ばかりで推理しようとする気が感じられない．だから人狼に違いない」や「彼女は全く雑談に応じない．きっとボロを出さないようにしている．だから人狼に違いない」のように雑談を手がかりに考えるものも存在する．

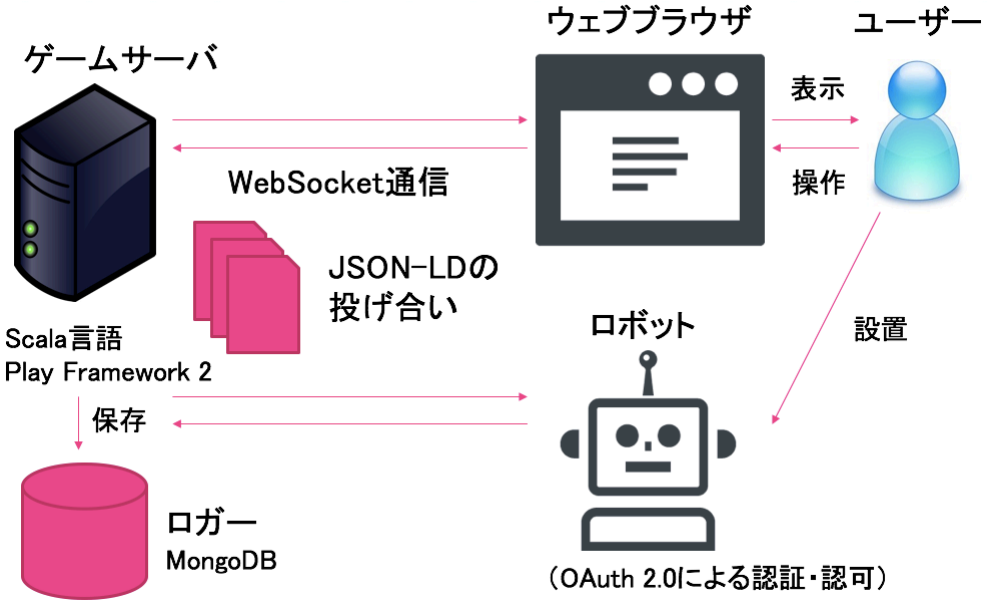

図 1: システム構成

襲撃を防ぐことができる）」が「自分は村人だ」と答える

- 他のプレイヤーが「占い師」だと名乗っている状況で,「狂人（人狼ではないが『人狼』陣営に属する）」が「自分が本当の占い師だ」と名乗る
- 「人狼」が,「人狼」ではないと分かっている[6]プレイヤーを指して「彼は人狼だと思う」と述べる

といったものがあげられる．これらの発話に共通するのは,「（自他問わずに）あるプレイヤーに対して，自分が認識している正体とは違う正体を述べている」ということであり，発話時の「認識している正体」を観測できれば，発話と本心が対応付けられたデータを収集することができる．

さて，人狼 BBS の発話ログを観察すると，

> 異老｜％五宿｜M修旅虎兵宿妙†羊孫娘面
> 真偽｜霊白白｜灰白灰灰灰灰灰†白白狼狼
> 偽真｜霊白白｜白灰灰灰灰灰灰†白白狼狼

のように，表形式で推理や確定事実を述べるプレイヤーが一定数存在する．推理の際にプレイヤーと正体とを対応付ける表（以降，**対応表**）を利用することは，直観的に全体を把握でき，矛盾などを認識するのに役立つ．もしも，ゲームのシステム上で対応表が実装されていれば，プレイヤーが推理，認識した正体を記述する可能性は高く，発話時にシステムが対応表の情報を取得することで，プレイヤーの本心に関するデータを自然に収集できるようになると考えられる．また，独り言や「人狼」同士の会話，誰を処刑するかの投票などにも本心が表れるため，それらの情報を発話と共に取得することで，発話と本心との対応がついたデータセットを構築することができる．次節では，LiCOS がこれらの情報をどのように取得しているかを説明する．

[6]「人狼」は，誰が他の「人狼」であるか分かる．

3 LiCOS

図 1 に，LiCOS のシステム構成を示す．ユーザーはウェブブラウザを通して人間のプレイヤーとしてゲームに直接参加するか，あるいはロボットを設置してロボットにゲームに参加させることができる．ゲームサーバが複数のプレイヤーのウェブブラウザやロボットとそれぞれ非同期通信を WebSocket を用いて行う．ゲームサーバとプレイヤーは相互に，ソースコード 1 のような JSON-LD フォーマットで構造化されたデータを WebSocket を通して送り合う．データの規約や例については，https://werewolf.world/を参照されたい．プレイヤーとゲームサーバでやりとりしたデータは，ログとしてタイムスタンプや送信先の情報を更新した上で，全てをデータベース MongoDB に保存する．ゲームサーバは，ウェブアプリケーションフレームワークの Play Framework 2 を用いて Scala 言語で開発した．

図 2 で，人間のプレイヤーが使用するユーザインターフェース画面の例を示す．左上に，ゲーム内の日にちとフェーズ，そのフェーズ終了までの残り時間を表示する．右上に，ゲーム中にプレイヤーが演じるエージェントの見た目の画像や名前（例では「モーリッツ」）と，正体（例では「人狼」）を表示する．左中央にプレイヤーの発話のタイムラインを表示する．下部に発話の

図 2: ユーザインターフェース画面

ための発話入力フォームを表示する．右中央に対応表を表示する．なお，ロボットのプレイヤーは，このユーザインターフェースを使用せず，直接 WebSocket を通してゲームサーバと通信する．

3.1　タイムライン

タイムラインには，プレイヤーが閲覧可能な発話が時間順序で表示される．新しい発話はタイムラインの下部に追加される．発話数が増えて画面に入りきらない場合は，スクロールすることで過去のタイムラインが閲覧できる．

3.2　発話入力フォーム

発話入力フォームは，図 2 の下部の左・中央・右の 3 つのフォームである．左のフォームは公開用であり，ここから投稿された発話は全てのプレイヤーが閲覧できる．基本的に左のフォームを使用して対話が行われる．中央のフォームは非公開用であり，ここから投稿された発話は投稿したプレイヤーにのみ閲覧ができる．プレイヤーが後で見返すためのメモを自由記述で残すことができるため，正体予想の根拠や前提条件など，対応表では表せない本心がこのフォームに投稿されるかもしれない．右のフォームは人狼用であり，人狼のみが発話を投稿でき，ここから投稿された発話は人狼のみ閲覧できる．それぞれの入力フォームには，一回の発話の文字数制限や，1 日当たりの発言回数制限が表示され，制限に達した場合は投稿ボタンがクリックできなくなる．さらに，発話直後も一定時間はゲームサーバが次の発話を受け付けないため，その時間も投稿ボタンはクリックできなくなる．

3.3　対応表

対応表の各カラムはそれぞれ正体を示しており，左から順に村人，占い師，霊媒師，狩人，フリーメイソン，狂人，人狼，ハムスター人間である．各カラムのヘッダーには，正体の種類を表す画像と，その正体であるプレイヤーの数が書かれている．各行はそれぞれプレイヤーを示しており，各行のヘッダーには，各プレイヤーが演じるエージェントの画像と名前が書かれている．各セルは，プレイヤーと正体の対応を示しており，もしプレイヤー本人がその対応が正しい（もしくは，間違っている）と考えれば，セルをクリックすることでセル内の表示を初期値の「?」から「△」,「○」,「×」に変更できる．図 2 では，モーリッツを演じる人狼プレイヤーが，ヴァルターが占い師であり，ニコラスが村人であると考え，対応表の該当箇所をどちらも「○」に変更している．変更情報は，逐次ゲームサーバに送信されログとして保存される．さらに，システムからの情報で正体が明かされた場合は，該当するセルを赤縁にし「○」を表示する．また，正体が異なることが明かされた情報についてもセルを黒塗りにし，明かされた日にちを白字で表示する．このようにシステムから変更されたセルの情報は，プレイヤーが変更

できない．システムから明かされる情報で，上記のようにセルが変更されるのは次のいずれかの場合のみである．

- 最初に明かされる本人の正体
- プレイヤーが占い師だった時，占った翌日に明かされる占い先が人狼か否か
- プレイヤーが霊媒師だった時，霊媒した翌日に明かされる処刑先が人狼か否か

3.4 発話と本心の対応づけ

プレイヤーは，タイムラインを通して他のプレイヤーと対話し，他プレイヤーの正体の予想を対応表で容易にまとめることが可能である．プレイヤーの発言と対応表の更新はタイムスタンプとともに全てロガーに保存されるため，対応表の入力を本心と仮定することで，プレイヤーの発話と本心の対応づけが得られる．加えて，非公開の発話入力フォームに入力された発話にも，本心が現れる可能性がある．これもタイムスタンプとともに全てロガーに保存されるため，ここからもプレイヤーの発話と本心の対応づけが得られるかもしれない．

ソースコード 1: プレイヤーが送信するデータの例

```
{
  "@context": [
      "https://werewolf.world/context/0.1/base.jsonld",
      "https://werewolf.world/context/0.1/chat.jsonld"
  ],
  "@id": "https://werewolf.world/resource/0.1/playerMessage",
  "villageId": 3,
  "villageName": "横国の森の小さな村",
  "totalNumberOfAgents": 15,
  "token": "eFVr3O93oLhmnE8OqTMl5VSVGIV",
  "phase": "day␣conversation",
  "date": 1,
  "phaseTimeLimit": 600,
  "phaseStartTime": "2017-11-17T12:50:12.568+09:00",
  "serverTimestamp": null,
  "clientTimestamp": "2017-11-17T12:51:24.123+09:00",
  "directionality": "client␣to␣server",
  "intensionalDisclosureRange": "public",
  "extensionalDisclosureRange": [],
  "agent": {
      "@id": "https://werewolf.world/resource/0.1/Moritz",
      "id": 1,
      "name": {"ja": "モーリッツ"},
      "image": "https://werewolf.world/image/0.1/Moritz.jpg"
  },
  "isMine": true,
  "id": 3,
  "counter": 3,
  "limit": 10,
  "interval": "5s",
  "text": ">>1␣そんなはずはない.ワシが占い師じゃ.",
  "characterLimit": 140,
  "language": "ja",
  "isOver": false
}
```

4 まとめ

本稿では，人狼ゲームにおける欺瞞と本心の関係を考察し，欺瞞対話観測システム LiCOS の説明を行った．今後，LiCOS を利用して欺瞞対話コーパスを構築する予定である．

謝辞

本研究を進めるにあたり，サーバマシンを提供してくださった rakumo 株式会社様に感謝します．開発に協力してくださった横浜国立大学のサークル YNU WAI-WAI の宇田川悠大氏，清水彰馬氏，そして森研究室の阿部穣太郎氏に感謝します．

参考文献

[1] Yuiko Tsunomori, Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura. An Analysis Towards Dialogue-Based Deception Detection. International Workshop on Spoken Dialog System Technology, 2015.

[2] 東中竜一郎, 船越孝太郎, 荒木雅弘, 塚原裕史, 小林優佳, 水上雅博. テキストチャットを用いた雑談対話コーパスの構築と対話破綻の分析. 自然言語処理, 23(1), 2015.

[3] 塚原裕史, 内海慶. オープンプラットフォームとクラウドソーシングを活用した対話コーパス構築方法. 言語処理学会第 21 回年次大会発表論文集, 2015.

[4] 林友超, 馬場瑞穂, 宇津呂武仁. 役職確定情報に着目した人狼ログ・ダイジェストの作成. 第 30 回人工知能学会全国大会論文集, 2016.

[5] 稲葉通将, 鳥海不二夫, 高橋健一. 人狼ゲームデータの統計的分析. ゲームプログラミングワークショップ 2012 論文集, 2012.

[6] 大谷直樹, 河原大輔, 黒橋禎夫, 鍛治伸裕, 颯々野学. 連想ゲームによるコモンセンス知識の獲得. 言語処理学会第 22 回年次大会発表論文集, 2016.

[7] 加藤義清, 草刈嗣人, 平田健成, 庄司裕子. 研究者キャリアゲームで獲得される戦略知識. 2006 年度人工知能学会全国大会 (第 20 回) 論文集, 2006.