

シャーロックホームズ短編

～ナレッジグラフ推論チャレンジ2019～

2019年11月30日

株式会社野村総合研究所

田村光太郎 角田充弘 外園康智

〒100-0004

東京都千代田区大手町1-9-2 大手町フィナンシャルシティ グランキューブ

NRI

Share the Next Values!



応募者概要

■ 応募者の氏名、所属、連絡先

● 田村光太郎

- 株式会社 野村総合研究所
 - (出向) NRIデジタル株式会社
- E-mail : k4-tamura@nri-digital.jp
- TEL : 070-4560-6025
- Web : <https://scholar.google.co.jp/citations?user=KHc7j4EAAAAJ&hl=ja>

● 外園康智

- 株式会社 野村総合研究所
- E-mail : y-hokazono@nri.co.jp
- TEL : 080-1353-6360

● 角田充弘

- 株式会社 野村総合研究所
- E-mail : m1-tsunoda@nri.co.jp
- TEL : 090-5768-6676

第2回チャレンジで目指したこと

チャレンジ① シャーロックの鋭い観察による推理（= **仮説推論**）を、真似できるか？

- 「密室殺人」について、観察事実から、どんな犯行だったか推理する。
- 「花婿失踪事件」は、「花婿＝継父」を見抜けるか？

チャレンジ② 短編の“不完全さ”を補うことができるか？ **犯行解を一意にする条件の提示**

各短編は犯人自白か死亡のため、証拠は提示できていない。

- 「背中の曲がった男」 死亡原因は卒中
- 「悪魔の足」 第一事件の犯人は死亡、第二事件は自白
- 「まだらの紐」 犯人死亡

さらに、コナン・ドイルの意図した解以外の**面白い新解釈**を与える。

チャレンジ③ 登場人物の発言の嘘つき問題を考慮して、犯行が解けるか？

チャレンジ④ “背景知識”＋日本語文のみ（ナレッジグラフを介さず）で推論できるか？

チャレンジ⑤ 踊る人形の画像認識は、事前学習なしで挑み、**原著の手書き人形図の暗号文解読**も実施

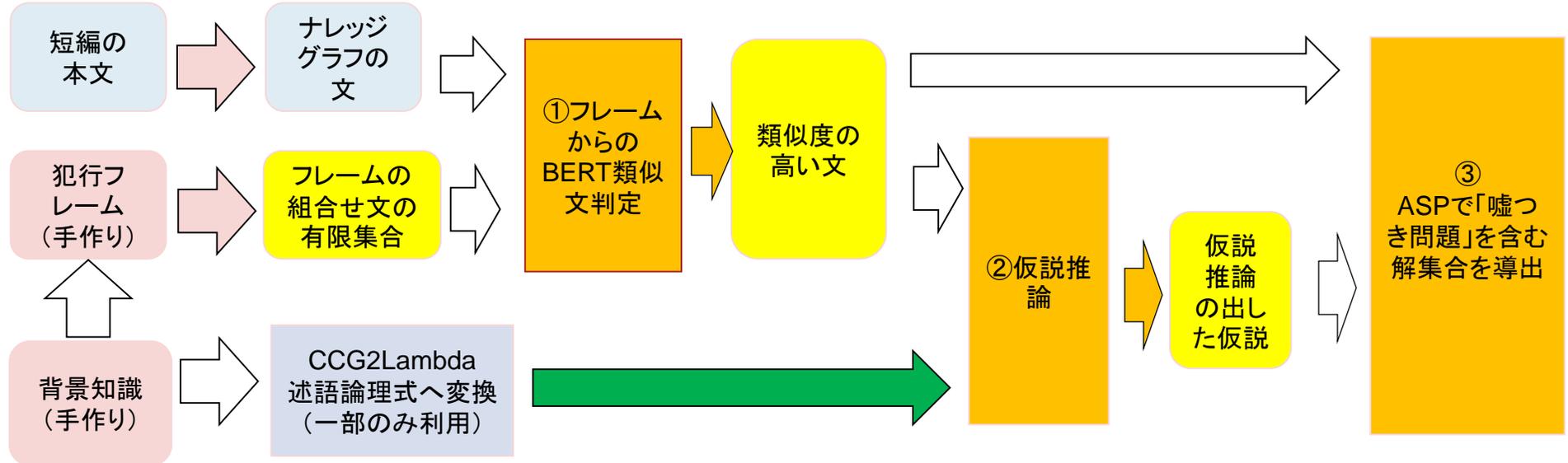
犯行解特定へのステップ

ステップ① 短編の本文から“犯行フレーム”に合致した文を、BERTの類似文判定 →ファクトの抽出

ステップ② 仮説推論にて、“殺人事件の背景知識”と”①のファクトから、犯行の”犯行解の仮説“を導出。

ステップ③ 仮説推論が導出した“仮説”も含めて、“嘘つき問題”を含みながら、充足可能解をASP（解集合プログラミング）で導出。犯行解が一意になる条件も特定

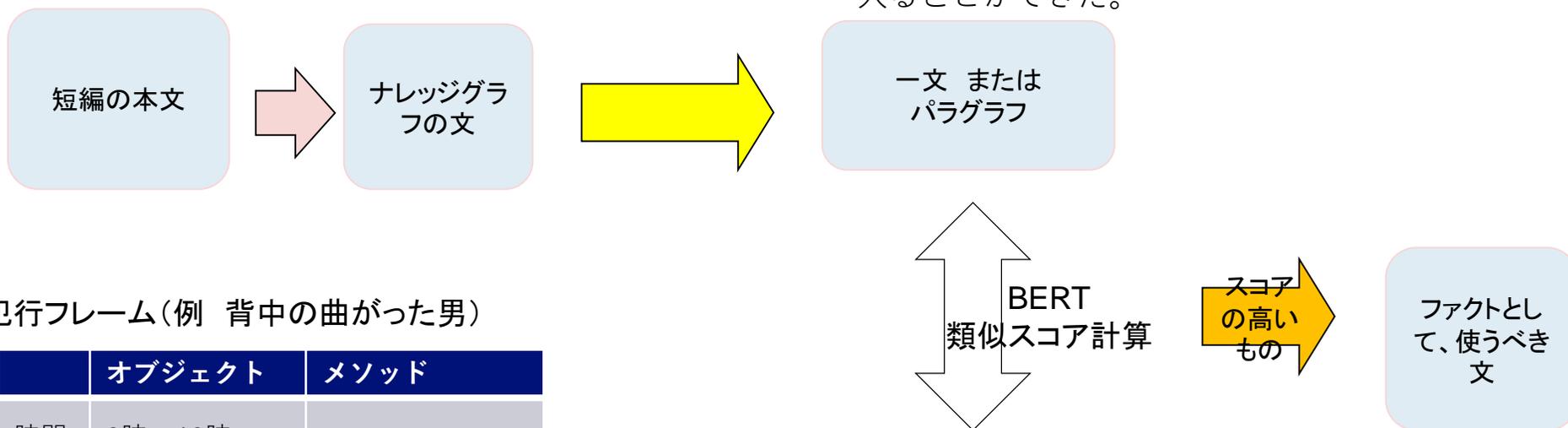
（ステップ④ 事前の学習モデルなしで、各人形図の同一性を画像認識技術で判別する。）



ステップ①手法説明 フレームから類似文判定 ファクトに含める文の選別

本文の一文と、“フレームから組合せ的に作られる文”を、BERTモデルを使って、類似度を計算する。
 “類似度”の高い文をファクトとして利用。

例) 御者は簡単に部屋の中に入ることができた。



犯行フレーム(例 背中の曲がった男)

	オブジェクト	メソッド
時間	9時、10時	
空間	居間 廊下 庭	窓が開いている 鍵がかかる 繋がる
人物	バークレ ナンシ、 ジェーン、 御者、コック	話す 入る 殺す 死ぬ



例) 御者は居間に入る
居間は窓が開いている

ステップ①手法説明 短編本文から仮説推論やASPに必要なファクト抽出

手法 1) ナレッジグラフ利用

←ナレッジグラフ構築の労力が高い

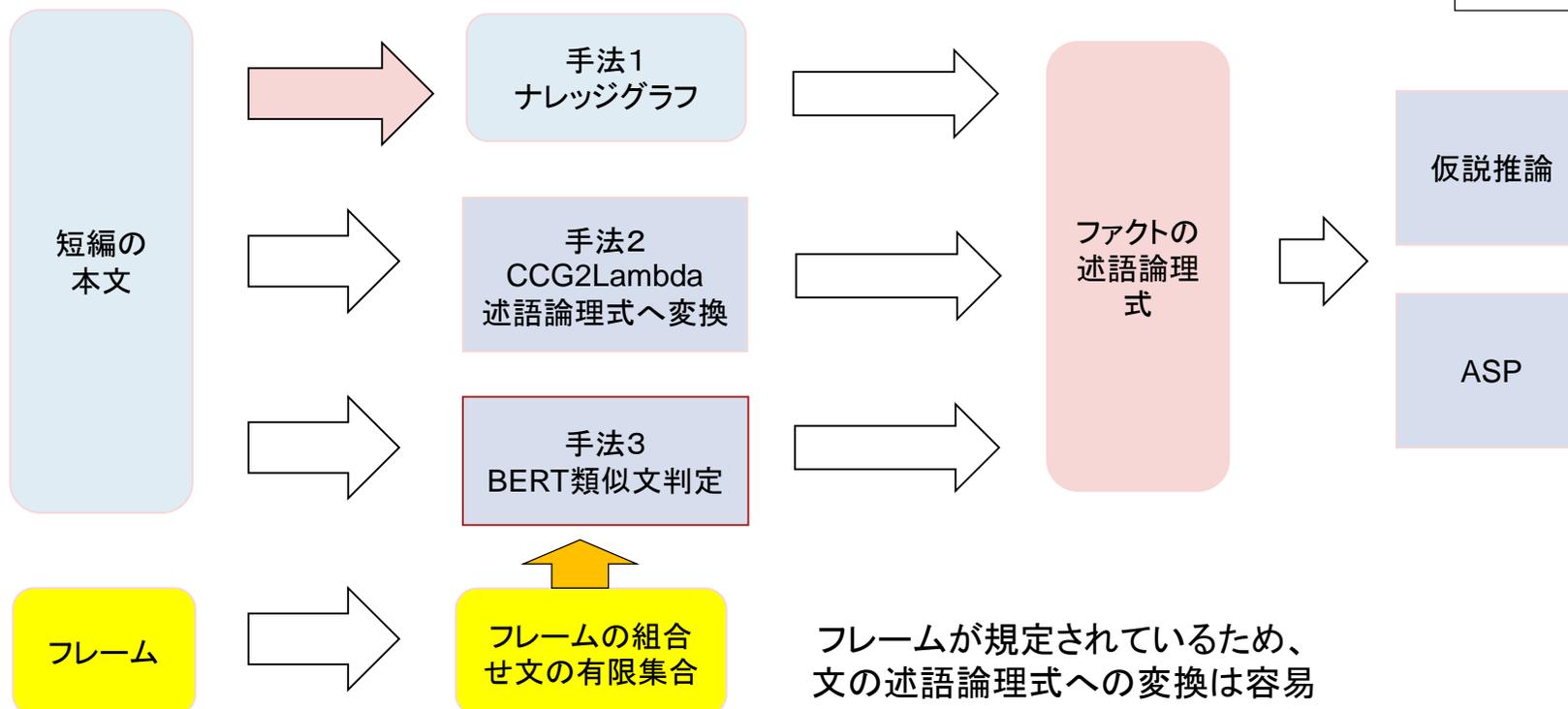
手法 2) CCG2Lambda利用

←口語文や未知語の精度が低い。

手法 3) BERTの類似文判定で抽出

←フレームの設定が必要。口語文の類似判定が難しい。

今回は手法 3
を採用



フレームが規定されているため、
文の述語論理式への変換は容易

ステップ②手法説明 仮説推論＝観測事実から犯行の仮説を挙げる

背景知識を用いて、観測事実を説明する“仮説”を求める推論

<背景知識>

- 庭から部屋に侵入があったならば、庭に足跡が残る。
- 窓が開いたならば、部屋に侵入できる。

<観測事実>

- 庭に足跡があった。

<仮説推論>

- 窓が開いていたのだろう

★“密室殺人の背景知識”を使って「背中の曲がった男」「悪魔の足」「まだらの紐」で仮説推論を行う

★「花婿失踪事件」では、「花婿＝継父」を見抜く

ステップ③手法説明 ASPによる“嘘つき問題”も含む犯行解の解集合の導出

①嘘つきは、登場人間の中で、0人からn人まで設定。

`0 {liar(P):person(P)}n.`

②各人物の発言を記述

例「ジェーンは、『バークレとナンシーが討論していた』と言った」

`says(jane,argue(berkeley,nancy),1).`

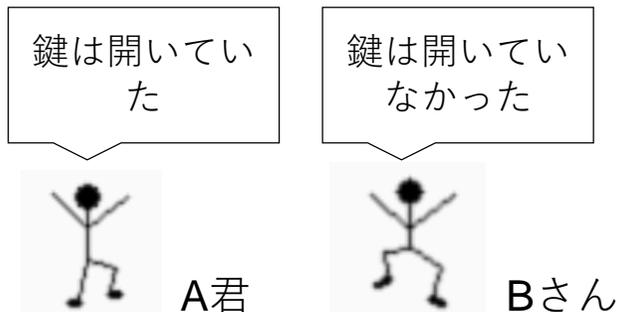
③正直者が話すことは正しいとするが、嘘つきが話したことは誤り。

(ただし、必要に応じて、正直者の“勘違い”や、嘘つきが正直発言と嘘発言を混ぜる場合も考慮する)

`holds(S) :- says(P,S,1),honest(P).` Pが正直者ならば、Pの発言内容Sは、成立する。

`0{-holds(S)}1 :- says(P,S,1),liar(P).` Pが嘘つきならば、Pの発言内容Sは、成立するか不成立かどちらか。

④各登場人物の発言内容毎に、成立と不成立を区別した解モデルを、ASPで解く。



	A正直者	A嘘つき
B正直者	×	○
B嘘つき	○	○

各短編の重点ポイントと解法手法

題材	課題	重点ポイントと、各ステップの実施有無
同一事件 (花婿失踪事件)	花婿はなぜ消えたか？ (説明)	①結婚式からの失踪のフレーム BERT 類似文判定 ②仮説推論 花婿(ホズマ)=継父(ウィンディバンク)と、結婚詐欺を導く。
まだらの紐	ヘレンを殺したのは誰か？ (犯人+説明)	②密室殺人の仮説推論 “密室に侵入する小さいもの”の存在の可能性を推論。 (③第一回コンテストで充足可能性問題 SAT で示したため、今年の実施なし)
悪魔の足	各人物を殺したのは誰か？ (犯人+説明)	②密室殺人の仮説推論 密室の“毒ガス”で殺人が起こった可能性を推論。 ③ ASP で、複数解を提示。各解が一意になる条件を提示する。
背中の曲がった男	バークリはなぜ死んだのか？ (説明)	①密室殺人のフレーム BERT 類似文判定 ②密室殺人の仮説推論 “庭に男が潜んで、部屋に侵入した”可能性を推論。 ③各人物が嘘をついている可能性を考慮。“嘘つき”問題を含んで、 ASP で複数解を提示。各解が一意になる条件を提示する。
踊る人形	暗号を解け (暗号の解読)	④各人形図の同一性を、事前学習なしの画像認識 AI で判別する。 ⑤シンボル化した人形図を、単換字暗号解読プログラムにて、暗号解読

全ての短編で、ホームズの解決シーン以前の文のみを利用

同一事件(花婿失踪事件) あらすじ

- 花嫁のメアリーサザーランドは行方不明の花婿ホスマーエンジェルを探してほしいとホームズの家を訪ねる。結婚式当日教会に来るはずの花婿が失踪してようとして行方が分からない。
- ホスマーが、ひげをたくわえ、薄いけれど色のついたメガネをかけたひどく内気な人物であることをメアリーから説明を受ける。
- ホームズは、ホスマーが変装していることに気づき、ホスマーがメアリーの継父であることを見抜く。
- メアリーの母親は一回りも年下の若い男と再婚していたが、この継父はメアリーに託されていた財産を使い込んでいた。そのため、メアリーに結婚されてはお金に困ってしまうこの継父は変装してメアリーに近づき、結婚式にまでこぎつけたのだ。そして、結婚式から土壇場で逃げ出し、しばらくはメアリーがショックで結婚のことは考えられないようにした。

同一事件(花婿失踪事件) フレームを設定するための知識

結婚式から、花嫁もしくは花婿が失踪した場合の背景知識

背景知識1 結婚式からの失踪の原因は、死ぬか、嫌いになるか、結婚詐欺

背景知識2 結婚詐欺の特徴 <https://fraud-lawyer.xyz/archives/515> より

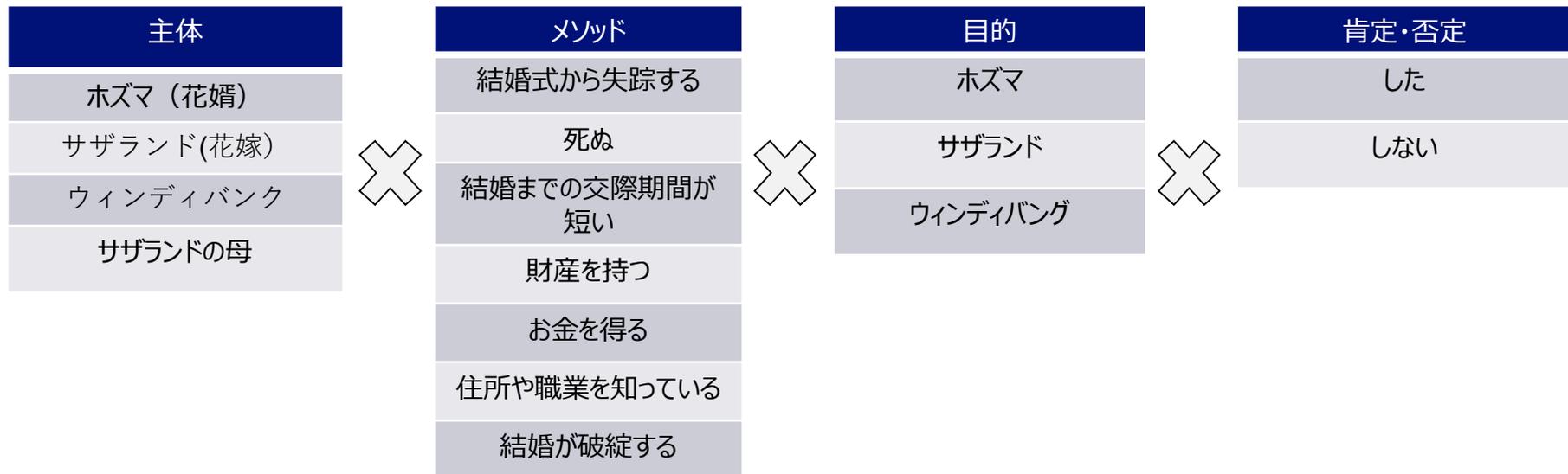
- ・ yは財産を持つ
- ・ yは x の住所や職業をよく知らない。
- ・ x は y からお金を得る
- ・ yと x の婚約までの交際期間が短い。

背景知識3 xが y の結婚に反対する理由

- ・ y が結婚すると、 x がお金を失う。
- ・ x が y のことを好き。

同一事件(花婿失踪事件) 知識からフレームの生成

「結婚式からの失踪・結婚詐欺」のフレーム



組合せの中で、主体と目的が同じ場合などを省いて、69文を半自動作成

フレームの組み合わせで作られる文の例

- ・ホズマは結婚式から失踪した
- ・サザランドはホズマと結婚までの交際期間が短い

同一事件(花婿失踪事件) BERT類似文判定

次の2パターンで類似文判定を実施。

「フレームから生成した文」と「短編本文」の比較

「フレームから生成した文」と「ナレッジグラフ化時の短文」の比較

結果は「花婿失踪事件類似文判定」エクセルを参照

結論

- ・「短編本文」では、まだ精度がでない。
- ・「ナレッジグラフ化した短文」の結果を使い、仮説推論に必要な文の抽出を半自動的に実施
(次頁のステップを参考)

同一事件(花婿失踪事件) BERT類似文判定

類似文のスコアを使い、観測事実（ファクト）に含めるかの判定ロジック

- ①類似度が**0.85**以上のものを抽出
- ②肯定・否定など、反対の意味になっているものは、スコアの高い方を入れる。
- ③スコアが低くても、“意味が同じようなグループ“の中で一番スコアの高いものを入れる。（一部手作業）

フレームから作った文	番号	短編の文	類似度
ホズマは結婚式から失踪した	292	ホズマは14日朝失踪した	0.82
ホズマとサザランドの婚約までの交際期間は短い	142	サザランドは初散歩の時にホズマと婚約した	0.79
ホズマの行方は分からない	202	サザランドはホズマの行方を分からない	0.87
サザランドはホズマの住所や職業をよく知らない	230	サザランドはホズマの住所を知らない	0.91
サザランドは財産を持っている	87	サザランドは稼ぎを持つ	0.86
ウィンディバンクはサザランドの結婚を承知しない	104	ウィンディバンクは、サザランドとサザランドの母が舞踏会に行くことを承知しない	0.90

同一事件(花婿失踪事件) 仮説推論① 背景知識の述語論理式

背景知識の文を述語論理式に変換 ⇒ 仮説推論エンジンの背景知識とする

#結婚式の失踪の原因

```
rule kb1 {dead(x) => disappear(x)} #死ぬか、  
rule kb2 {hate(x,y) => disappear(x)} #嫌いになるか  
rule kb3 {marrigetrick(x,y) => disappear(x)} #結婚詐欺
```

#yは財産をもって、yがxをよく知らず、xはyからyがお金を得て、交際期間が短い場合は、結婚詐欺

```
rule kb4 {rich(y) ^unknown(y,x)^ getmoney(x,y) ^mijikai(x,y) => marrigetrick(x,y)}
```

#結婚詐欺ならば、結婚は破綻する。

```
rule kb6 {marrigetrick(x,y)=>faultmarrige(y)}
```

#yの結婚が破綻して、xがお金を得られるならば、xはyの結婚を承知しない。

```
rule kb7 {faultmarrige(y)^getmoney(x,y)=>hantai(x,y)}
```

#死体が発見されたら、死である。

```
rule kb5 {founddead(x) => dead(x)}
```

同一事件(花婿失踪事件) 仮説推論②

- 類似文の中で、スコアの高いものを観測事実とする
- フレームから作った文は述語論理式に直すのは容易

短編の文から得られた観測事実	観測事実の論理式表現
ホズマは結婚式から失踪した	disappear(Hozuma)
ホズマとサザランドの婚約までの交際期間は短い	mijikai(Hozuma,Sutherland)
ホズマの死体は見つかっていない	!founddead(Hozuma)
サザランドはホズマの住所や職業をよく知らない	unknown(Sutherland,Hozuma)
サザランドは財産を持っている	rich(Sutherland)
ウィンディバンクはサザランドの結婚を承知しない	hantai(keifu,Sutherland)

- 背景知識 + 観測事実を仮説推論エンジンに入れると

仮説結果

- ① **marrigetrick(Hozuma,Sutherland)** ホズマは、サーザランドに対して、結婚詐欺をした。
- ② **(keifu = Hozuma)** ホズマと継父は同一だ。
- ③ **not dead(Hozuma)** ホズマは死んでいない。

「まだらの紐」 あらすじ

- ◆ジュリアとヘレンの姉妹は、亡き母親の財産を相続した。姉妹は、結婚して家を出る際に、遺産を持って出ることができる。義理の父のロイロット博士は、娘が結婚すると、遺産の取り分減る。
- ◆2年前のある夜、姉のジュリアは、結婚直前に謎の死を遂げる。妹ヘレンの証言によれば、低い口笛のような音が聞こえた後「まだらの紐 (Speckled Band) が！」と言う意味不明の言葉を残して死んでしまった。
- ◆2年後、結婚が決まったヘレンは、夜に屋敷での不気味な体験に恐怖を感じ、ホームズに相談する。ヘレンは、修理のため、かつてのジュリアの部屋で寝ていること、昨夜は姉が亡くなった時に聞いた低い口笛の音が聞こえたことなどを話す。
- ◆ホームズとワトソンは、すぐさま屋敷の各部屋を調査する。ジュリア部屋とロイロット部屋の間の通風孔と、そこに取り付けられた鳴らない呼び鈴の紐、その紐の真下に固定されたベッド。ロイロット部屋にある金庫と、ミルクの入った皿、先の部分が輪の形に細工された奇妙な鞭を見つける。
- ◆ホームズは、ヘレンが非常に危険な状態であることを見抜き、ヘレンに当日夜の行動を指示。ヘレンは自分の部屋へ移ってベッドに入り、ジュリア部屋にはホームズとワトソンが忍び込む。そして、夜中を待っていると・・・。
- ◆毒蛇を調教し、義理の娘2人を殺そうとしたロイロットは、逆に蛇にかまれて死んでしまう。
- ◆ワトソンは「これでは、君が間接的にロイロットを殺したことになる云々」と言うが、ホームズは「この事に対して僕は少しも良心の呵責を感じない云々」と答えた。

まだらの紐 仮説推論「密室殺人の背景知識」 (悪魔の足・背中の曲がった男共通)

< 密室殺人の背景知識 >

rule kb0 {murder(x,y,r)=>dead(y,r)} #殺人が起きたら死亡する。

#密室部屋の殺人は、外からの侵入か、自殺か、部屋内部に潜む犯行か、空気性毒物か、ドリンク系薬物か、

#外からの侵入のケース

rule kb11 {motive(x) ^ invade(x,r,o) => murder(x,y,r)} #外部から部屋へ侵入して殺害

rule kb12 {windowopen(r) ^ hide(x,o) => invade(x,r,o)} #窓が空いているならば、部屋に侵入可能。

rule kb13 {invade(x,r,o) => footprint(o)} #侵入があったならば、外部に足跡が残る。

rule kb14 {locatbig(o) => hide(x,o)} #隠れる場所は大きくなければならない。

rule kb15 {keylock(r,t)=> !invade(x,r,t)} #ドアロックされていると通路からは部屋には侵入できない。

rule kb16 {keyholder(x,r)=>invade(x,r,o)} #キーの保有者は部屋に入れる。

rule kb101 {tuukikou(r,o) ^ small(x) => invade(x,r,o)} #通気口は小さいものならば、通れる

#自殺のケース

rule kb21 {motive(y) => murder(y,y,r)} #自殺動機ありの場合は、自殺

#部屋内部に潜む犯行の場合、

rule kb31 {motive(x)^hide(x,r)^escape(x,r,o) => murder(x,y,r)} #中に隠れ、かつ、そこから、外に脱出できる必要がある。

rule kb32 {escape(x,r,o) => footprint(o)} #脱出できるならば、外部に足跡が残る。

rule kb33 {windowopen(r) ^ hide(x,r) => escape(x,r,o)} #窓が開いていて、部屋に隠れれば脱出できる。

#空気系毒物のケース

rule kb41 {airpoison(r) ^ !windowopen(r) ^ !tuukikou(r) => dead(y,r)} #窓と通気口が開いておらず、部屋に空気系毒があると死亡

rule kb42 {airpoison(r) ^ !windowopen(r) ^ !tuukikou(r) => damagednerve(y,r)} #窓と通気口が開いておらず、部屋に空気系毒があると神経がマヒする。

rule kb43 {airpoison(r) => moekasu(r)} #空気系毒が部屋にあると、その燃えカスが残る。

#ドリンク系毒物

rule kb51 {drinkpoison(y) => dead(y,r)} #ドリンク系毒を飲むと死亡する。

rule kb52 {drinkpoison(y) => damagednerve(y,r)} #ドリンク系毒を飲むと神経がマヒする。。

rule kb53 {drinkpoison(y) => foundpoison(y)} #ドリンク系毒を飲むと毒が体内から毒が発見される。

rule kb54 {foundpoisoncup(y,r) => drinkpoison(y)} #毒入りカップが発見されると、ドリンク系毒を飲んでいる。

まだらの紐 仮説推論 観測事実と仮説推論の結果

<観測事実> 小説本文から手作業で作成

ヘレンは、ヘレンの部屋で死んだ。	dead(Helen,HelenRoom)
窓は閉まっていた。	!windowopen(HelenRoom)
庭は広い	locatbig(Garden)
部屋は広くない	!locatbig(HelenRoom)
通気口がある。	tuukikou(HelenRoom,RoylotRoom)
何者かは小さい	small(strange)

<仮説推論の結果>

drinkpoison(Helen)	×
airpoison(HelenRoom)	×
hide(man,Garden)	×
invade(man,Living,Garden)	×
murder(man,Helen,HelenRoom)	×
hide(man,HelenRoom)	×
invade(man,HelenRoom,Garden)	×
invade(strange,HelenRoom,RoylotRoom)	○
murder(strange,Helen,HelenRoom)	○

★小さい生き物がヘレンの部屋の中に侵入し、殺害した。

悪魔の足 あらすじ

- モーティマー・トリゲニスが、恐ろしい知らせを持って来る。妹のブレンダが死亡し、兄弟のオーウェンとジョージが発狂してしまったという。昨夜、モーティマーは、妹と兄弟の家と一緒に夕食をとった後、1人で牧師館に帰ったらしい。今朝になって、居間の椅子に座ったまま恐怖の表情を浮かべ、死亡・発狂している3人が発見されたのである。
- ホームズとワトソンは、3人の家政婦に事情を聞く。自分も恐怖で気絶してしまったという。侵入の形跡はなく、居間の暖炉には火を入れていた。
- 近所の住人でアフリカ探検家のスターンデイル博士が訪ねてくる。
- 翌朝、今度はモーティマーが自室で、前の事件同様に死んでいるという知らせが入る。ホームズは、ランプから灰を採取する。
- ホームズは、2件とも閉め切られた部屋で火の存在があり、立ち入った者が気絶したり息苦しくなったりしていることから、何かが燃やされ有毒な気体が発生していたと推理。
- ホームズは呼び出したスターンデイル博士に、1件目の犯人はモーティマー、2件目の犯人が博士であることを説明する。
- 博士は観念し、真相を明かす。モーティマーはトリゲニス一家の財産を独占するために、博士の家から「悪魔の足の根」というアフリカに伝わる毒草の粉末を盗み出し、使用したのである。博士とブレンダは長年の恋人同士であり、復讐のため、モーティマーのもとを訪れ、同様に「悪魔の足の根」を用いて殺害したのであった。

悪魔の足 仮説推論

■ 空気系毒で死んだ可能性を仮説推論する。

■ 犯行状況の観測事実

- observe { motive(man) ^ dead(Brend, Living) ^ !windowopen(Living) ^ locatbig(Garden) ^ moekasu(Living) ^ !locatbig(Living) ^ foundcup(Living) ^ damagednerve(George) ^ damagednerve(Owen) ^ !footprint(Garden) }
- require { drinkpoison(Living) ^ airpoison(Living) ^ hide(man, Garden) ^ invade(man, Living, Garden) ^ murder(man, Brend, Living) ^ hide(man, Living) ^ escape(man, Living) }

■ 推論結果

- { "literal" : "drinkpoison(Living)", "state" : "unsatisfiable" },
- { "literal" : "escape(man, Living)", "state" : "unsatisfiable" },
- { "literal" : "invade(man, Living, Garden)", "state" : "unsatisfiable" },
- { "literal" : "airpoison(Living)", "state" : "satisfied" }, ←居間に空気系の毒があった。
- { "literal" : "murder(man, Brend, Living)", "state" : "unsatisfied" },
- { "literal" : "hide(man, Garden)", "state" : "unsatisfiable" },
- { "literal" : "hide(man, Living)", "state" : "unsatisfied" } }

悪魔の足 ASPによる解法

- 仮説推論による方法で、薬物による犯行との仮説が立った。
 - 仮説に対する検証のプロセスとして、ASPで解く。
- 事件は、登場人物の発言によって、その状況が伝えられている。
 - 各登場人物の証言の真偽を精査し、**結論の分岐を探索**する。
 - 以下は、主な情報の伝達に関する本文中の記述。

第1の殺人

- 兄妹達の発狂・死亡は、モーティマーが発見し、司祭ラウンディに報告し、ホームズに伝えられる。
- 事件の状況が司祭ラウンディからスタンデールに伝えられる。

第2の殺人

- モーティマーの死は、司祭ラウンディによってホームズ伝えられる。
- スタンデールが、モーティマーの殺害状況を自白する。
- ホームズは、スタンデールの自白の発言を信じ、見逃す。

※ポータ夫人は真実しか話さないものと仮定した。

悪魔の足 背景知識

■ 殺害方法の知識（薬物によるものを中心に、条件を記述）

- `direct(P,Q,T) :-`
`person(P),person(Q),enter(P,X,T),enter(Q,Y,T),X==Y,P!=Q,time(T),room(X),room(Y),kill_Motive(P,Q,T,direct).`
- `airpoison(P,Q,T) :-` `person(P),person(Q),enter(P,X,T-1),enter(Q,X,T-1),P!=Q,room(X),time(T),kill_Motive(P,Q,T,airpoison).`
- `apertr(P,Q,T) :-`
`person(P),person(Q),room(X),room(Y),enter(P,X,T),enter(Q,Y,T),X!=Y,P!=Q,passable(X,Y),kill_Motive(P,Q,T,apertr).`

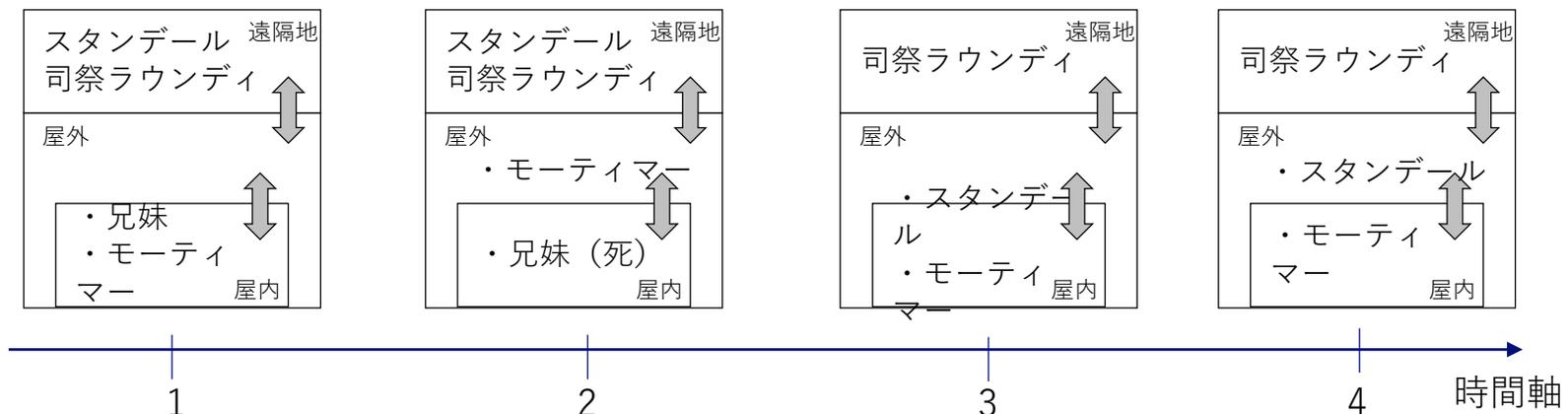
■ 共犯の条件

- `support(P,Q) :-` `person(P),person(Q), kill_Motive(P,R,T,M), kill_Motive(Q,R,T,M)`
- 共犯の意思があると、居場所についてうそをつく

■ 嘘の発言から得る情報

- ここでは、嘘の発言は、発言内容の裏を採用するのではなく、取り得る選択肢全てを考える。
 - ・ 例：「スタンデールは遠隔地にいた」が嘘の場合、スタンデールの所在は、遠隔地、屋外、屋内のどこかであるとする。

■ 図のように、トポロジー情報を与える。1, 2はトレゲニス家、3, 4はモーティマーの家。



悪魔の足 発言者を考慮に入れたファクト

- ラウンディ、スタンデール、モーティマーの居場所に関する事実と発言をASPに入れる。
- 発言者が、他人の所在や生存についての言及しているものすべてを採用した。

第1の殺人	第2の殺人
<pre>says(mortimer, live(brothers, 1), 1). live(mortimer, 1). live(Sterndale, 1). live(roundhay, 1). says(mortimer, dead(brothers, 2), 1). live(mortimer, 2). live(Sterndale, 2). live(roundhay, 2).</pre>	<pre>says(Sterndale, live(mortimer, 3), 1). live(Sterndale, 3). live(roundhay, 3). dead(mortimer, 4). live(Sterndale, 4). live(roundhay, 4).</pre>
<pre>says(mortimer, enter(brothers, inside, 1), 1). says(mortimer, enter(mortimer, inside, 1), 1). says(Sterndale, enter(Sterndale, farplace, 1), 1). says(roundhay, enter(roundhay, farplace, 1), 1). says(mortimer, enter(brothers, inside, 2), 1). says(mortimer, enter(mortimer, outside, 2), 1). says(Sterndale, enter(Sterndale, farplace, 2), 1). says(roundhay, enter(roundhay, farplace, 2), 1).</pre>	<pre>says(Sterndale, enter(mortimer, inside, 3), 1). says(Sterndale, enter(Sterndale, inside, 3), 1). says(Sterndale, enter(roundhay, farplace, 3), 1). says(Sterndale, enter(mortimer, inside, 4), 1). says(Sterndale, enter(Sterndale, outside, 4), 1). says(Sterndale, enter(roundhay, farplace, 4), 1).</pre>

悪魔の足 clingoのスキript

■ スクリプトの全容

- オブジェクトの設定、ルール記述部、F A C T 記述部からなる。
- 提出済ファイル (Devil_root.lp) と同じもの。適当な箇所で、4部に分割しているだけで、1つのファイルである。

person(brothers ; mortimer; standale; roundhey; porter). room(inside ; outside ; farplace). method(direct ; poison ; apertr). time(1..4). %%%%%%%%%% %%嘘つき問題 %%%%%%%%%% {liar(P):person(P)} = n. honest(brothers). honest(porter). {honest(P):person(P);liar(P):person(P)} = 5. :- honest(P),liar(P). fact(S) :- says(P,S,0),liar(P). fact(S) :- says(P,S,1),honest(P). -fact(S) :- says(P,S,1),liar(P). -fact(S) :- says(P,S,0),honest(P). %%%%%%%%%% %%事実がいくつあるか %%%%%%%%%% 1 {enter(Q,inside,T);enter(Q,outside,T); enter(Q,farplace,T)} 1:- says(P,enter(Q,R,T),1),liar(P),person(P),pers on(Q,room(R),time(T)). 1 {enter(Q,inside,T);enter(Q,outside,T); enter(Q,farplace,T)} 1:- says(P,enter(Q,R,T),0),liar(P),person(P),pers on(Q,room(R),time(T)). 1 {enter(Q,inside,T);enter(Q,outside,T); enter(Q,farplace,T)} 1 :- person(Q),time(T). %%first live(brothers,1). live(mortimer,1). live(standale,1). live(roundhey,1).	says(mortimer,dead(brothers,2),1). %says(roundhay,dead(brothers,2),1). live(mortimer,2). live(Sterndale,2). live(roundhay,2). says(mortimer,enter(brothers,inside,1),1). says(mortimer,enter(mortimer,inside,1),1). says(Sterndale,enter(Sterndale,farplace,1),1). says(roundhay,enter(roundhay,farplace,1),1). says(mortimer,enter(brothers,inside,2),1). says(mortimer,enter(mortimer,outside,2),1). says(Sterndale,enter(Sterndale,farplace,2),1). says(roundhay,enter(roundhay,farplace,2),1). %%second says(Sterndale,live(mortimer,3),1). %dead(brothers,3). %live(mortimer,3). live(Sterndale,3). live(roundhay,3). %dead(brothers,4). dead(mortimer,4). live(Sterndale,4). live(roundhay,4). says(Sterndale,enter(mortimer,inside,3),1). says(Sterndale,enter(Sterndale,inside,3),1). says(Sterndale,enter(roundhay,farplace,3),1).	says(Sterndale,enter(mortimer,inside,4),1). says(Sterndale,enter(Sterndale,outside,4),1). says(Sterndale,enter(roundhay,farplace,4),1). enter(P,R,T) :- fact(enter(P,R,T)),person(P),room(R),time(T). live(P,T) :- fact(live(P,T)),person(P),time(T). %%%%%%%%%% 1 {kill_Motive(P,Q,T,M):person(P), person(Q), time(T), method(M), P!=Q, live(Q,T-1)}). dead(Q,T) :- person(P), person(Q), time(T), kill(P,Q,T), live(Q,T-1). :- live(P,T), dead(P,T), person(P), time(T). dead(brothers,2). :- person(P),person(Q),time(T),direct(P,Q,T). kill(P,Q,T) :- person(P),person(Q),time(T),poison(P,Q,T). kill(P,Q,T) :- person(P),person(Q),time(T),apertr(P,Q,T). %:- can_other_kill(P,Q,T), can_direct_kill(P,Q,T), can_drug_kill(P,Q,T), can_aperture_kill(P,Q,T).	direct(P,Q,T) :- person(P),person(Q),enter(P,X,T),enter(Q,Y,T ,X==Y,P!=Q,time(T),room(X),room(Y),kill_Mo tive(P,Q,T,direct). poison(P,Q,T) :- person(P),person(Q),enter(P,X,T- 1),enter(Q,X,T- 1),P!=Q,time(T),room(X),kill_Motive(P,Q,T,poi son). apertr(P,Q,T) :- person(P),person(Q),room(X),room(Y),enter(P,X,T),enter(Q,Y,T),X!=Y,P!=Q,passable(X,Y ,kill_Motive(P,Q,T,apertr). passable(outside,inside). passable(inside,outside). %直接攻撃した状況は外傷がないので否定す る。 :- direct(P,brothers,Time), person(P). :- apertr(P,brothers,Time), person(P). :- direct(P,mortimer,Time), person(P). :- apertr(P,mortimer,Time), person(P). %表示用 %殺した人 criminal_1st(P) :- person(P), live(P,2), kill(P,brothers,Time),time(T). criminal_2nd(P) :- person(P), live(P,2), kill(P,mortimer,Time),time(T). #show criminal_1st/1. #show criminal_2nd/1. #show liar/1.
--	--	---	---

悪魔の足

- スクリプトの人物・建物・方法オブジェクトの定義
- オブジェクトの定義
- 提出

記述音と同

発言・ファクトの定義

る。部に分割しているだ

殺害方法におけるルール

```
person(brothers ; mortimer; standale; roundhey; porter).
room(inside ; outside ; farplace).
method(direct ; poison ; apertr).
time(1..4).
```

```
%%%%%%%%%%
%%嘘つき問題
%%%%%%%%%%
{liar(P):person(P)} = n.
honest(brothers).
honest(porter).
{honest(P):person(P);liar(P):person(P)} = 5.
:- honest(P),liar(P).
fact(S) :- says(P,S,0),liar(P).
fact(S) :- says(P,S,1),honest(P).
-fact(S) :- says(P,S,1),liar(P).
-fact(S) :- says(P,S,0),honest(P).
%%%%%%%%%%
%%事実がいくつあるか
%%%%%%%%%%
1 {enter(Q,inside,T);enter(Q,outside,T);
enter(Q,farplace,T)} 1:-
says(P,enter(Q,R,T),1),liar(P),person(P),pers
on(Q),room(R),time(T).
1 {enter(Q,inside,T);enter(Q,outside,T);
enter(Q,farplace,T)} 1:-
says(P,enter(Q,R,T),0),liar(P),person(P),pers
on(Q),room(R),time(T).
1 {enter(Q,inside,T);enter(Q,outside,T);
enter(Q,farplace,T)} 1 :- person(Q),time(T).
```

```
%%first
live(brothers,1).
live(mortimer,1).
live(standale,1).
live(roundhey,1).
```

```
says(mortimer,dead(brothers,1)).
%says(roundhay,dead(brothers,2),1).
live(mortimer,2).
live(Sterndale,2).
live(roundhay,2).
```

```
says(mortimer,enter(brothers,inside,1),1).
says(mortimer,enter(mortimer,inside,1),1).
says(Sterndale,enter(Sterndale,farplace,1),1).
says(roundhay,enter(roundhay,farplace,1),1).
```

```
says(mortimer,enter(brothers,inside,2),1).
says(mortimer,enter(mortimer,outside,2),1).
says(Sterndale,enter(Sterndale,farplace,2),1).
says(roundhay,enter(roundhay,farplace,2),1).
```

```
%%second
says(Sterndale,live(mortimer,3),1).
%dead(brothers,3).
%live(mortimer,3).
live(Sterndale,3).
live(roundhay,3).
```

```
%dead(brothers,4).
dead(mortimer,4).
live(Sterndale,4).
live(roundhay,4).
```

```
says(Sterndale,enter(mortimer,inside,3),1).
says(Sterndale,enter(Sterndale,inside,3),1).
says(Sterndale,enter(roundhay,farplace,3),1).
```

```
says(Sterndale,enter(mortimer,inside,4),1).
says(Sterndale,enter(Sterndale,outside,4),1).
says(Sterndale,enter(roundhay,farplace,4),1).
```

```
enter(P,R,T) :-
fact(enter(P,R,T)),person(P),room(R),time(T).
live(P,T) :- fact(live(P,T)),person(P),time(T).
%%%%%%%%%%
```

```
1 {kill_Motive(P,Q,T,M):person(P), person(Q),
time(T), method(M), P!=Q, live(Q,T-1)}.
```

```
dead(Q,T) :- person(P), person(Q), time(T),
kill(P,Q,T), live(Q,T-1).
:- live(P,T), dead(P,T), person(P), time(T).
```

```
dead(brothers,2). :-
person(P),person(Q),time(T),direct(P,Q,T).
kill(P,Q,T) :-
person(P),person(Q),time(T),poison(P,Q,T).
kill(P,Q,T) :-
person(P),person(Q),time(T),apertr(P,Q,T).
```

```
%:- can_other_kill(P,Q,T),
can_direct_kill(P,Q,T), can_drug_kill(P,Q,T),
can_aperture_kill(P,Q,T).
```

```
direct(P,Q,T) :-
person(P),person(Q),enter(P,X,T),enter(Q,Y,T),
X==Y,P!=Q,time(T),room(X),room(Y),kill_Mo
tive(P,Q,T,direct).
poison(P,Q,T) :-
person(P),person(Q),enter(P,X,T-
1),enter(Q,X,T-
1),P!=Q,time(T),room(X),kill_Motive(P,Q,T,poi
son).
```

```
apertr(P,Q,T) :-
person(P),person(Q),room(X),room(Y),enter(
P,X,T),enter(Q,Y,T),X!=Y,P!=Q,passable(X,Y)
,kill_Motive(P,Q,T,apertr).
```

```
passable(outside,inside).
passable(inside,outside).
```

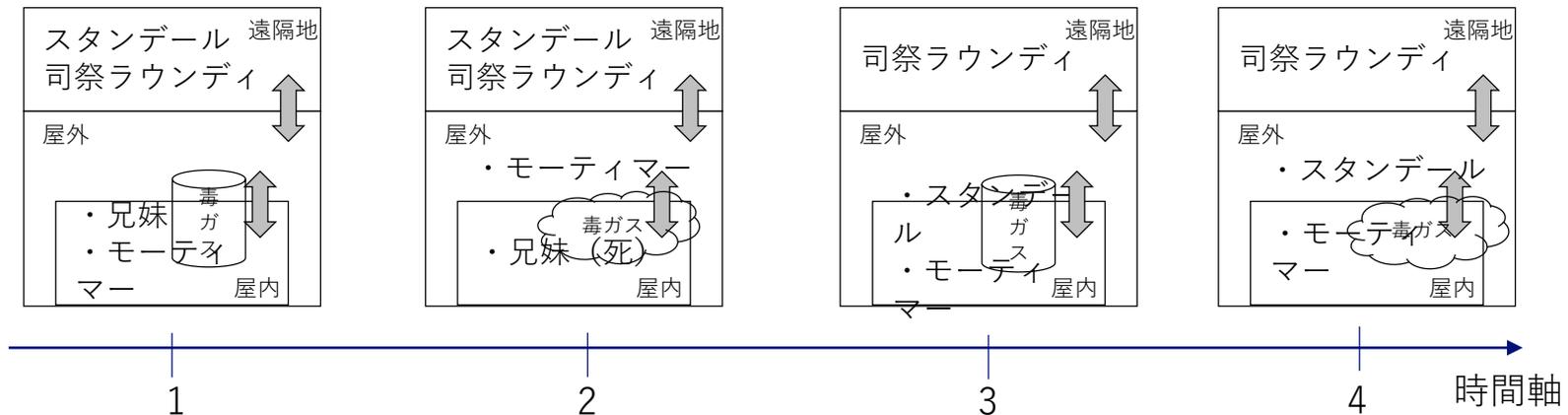
```
%直接攻撃した状況は外傷がないので否定す
る。
:- direct(P,brothers,Time), person(P).
:- apertr(P,brothers,Time), person(P).
:- direct(P,mortimer,Time), person(P).
:- apertr(P,mortimer,Time), person(P).
```

```
%表示用
%殺した人
criminal_1st(P) :- person(P), live(P,2),
kill(P,brothers,Time),time(T).
criminal_2nd(P) :- person(P), live(P,2),
kill(P,mortimer,Time),time(T).
```

```
#show criminal_1st/1.
#show criminal_2nd/1.
#show liar/1.
```

悪魔の足 ASPの結果（本解）

- ASPの出力として、登場人物の所在地や発言をそのまま採用すれば、矛盾なく解が求まり、ストーリーと同じ状況が得られる。
 - 第1の事件はモーティマー、第2の事件はスタンデールであることが導かれる。
 - 事故や自死が含まれる結論は除外した。
- 具体的な出力は以下の通り。
 - スタンデールがモーティマーを時刻4に、モーティマーが兄妹達を時刻2に殺害し、どちらもその殺害手法として、薬物（ガス）が条件に整合する。モーティマーとスタンデールが、それぞれ第1事件、第2事件の殺人犯である。
 - Answer: ストーリと同じ結論。
 - `dead(mortimer,4) dead(brothers,2) poison(mortimer,brothers,2) poison(Sterndale,mortimer,4) criminal_2nd(Sterndale) criminal_1st(mortimer)`



悪魔の足 (別解) 嘘つき2人のときの解

- liar () は嘘をついている人物。第1、第2事件の犯人がcriminal_1stとcriminal_2ndで表される。

- Answer:
 - liar(standale) liar(roundhey) criminal_2nd(standale) criminal_1st(mortimer)
- Answer:
 - liar(mortimer) liar(roundhey) criminal_2nd(standale) criminal_1st(mortimer)
- Answer:
 - liar(mortimer) liar(roundhey) criminal_2nd(standale) criminal_1st(mortimer) criminal_1st(roundhey)
- Answer:
 - liar(mortimer) liar(roundhey) criminal_2nd(standale) criminal_1st(mortimer) criminal_1st(standale)
- Answer:
 - liar(mortimer) liar(standale) criminal_2nd(standale) criminal_1st(mortimer) criminal_1st(roundhey)
- Answer:
 - liar(mortimer) liar(standale) criminal_2nd(standale) criminal_1st(mortimer)
- Answer:
 - liar(mortimer) liar(standale) criminal_2nd(standale) criminal_1st(mortimer) criminal_1st(standale)

悪魔の足 犯行解の可能性のまとめ

- 共犯の意思を持つ場合の解の分岐
- 所在が嘘であったとしても、犯行と関係ない場所にいることもある。

解の区分	モーティマー 「兄妹と自分だけ」	スタンデール 「プリマスにいた」	ラウンディ 「教会地区にいた」	犯人	解を一意に定めるための条件
本解	真実	—	—	モーティマー	事件当夜の現場の状況
—	嘘	真実	真実	不審者(?)	仮説推論より否定
別解 1			嘘	ラウンディ モーティマー	事件当夜の ラウンディの所在 スタンデールの所在
別解 2		真実	スタンデール モーティマー		
別解 3		嘘	スタンデール ラウンディ モーティマー		

悪魔の足 真偽によって分岐を起こす発言

- 証言に嘘が含まれているときの、事件の真相を分岐させる発言（条件）とその結果は以下の通り。
- `says(Sterndale,enter(Sterndale,farplace,1),1)` :
 - 真： スタンデールが第1の殺人の日に、アフリカへ出発しようと.プリマスにいた。
 - 偽： スタンデールは第1の殺人の日に、事件現場近く（屋内か屋外）にいた。
 - ⇒第1の殺人にスタンデールが関与
- `says(roundhay,enter(roundhay,farplace,1),1)` :
 - 真： 司祭ラウンディは第1の殺人の日は、事件現場にいない。
 - 偽： 司祭ラウンディは第1の殺人の日に、事件現場近く（屋内か屋外）にいた。
 - ⇒第1の殺人に司祭ラウンディが関与（動機？）
- スタンデール、司祭ラウンディの関与は考えられる。
 - ポータ夫人は、事件当時の情報を正確に持っていないため、ここでの検証は不能。
- 第2の殺人の日に関する発言は、真偽を問うても、スタンデールが犯人orモーティマーの事故・自殺の結論しか出ない。
 - 事故の場合、スタンデールが自白するメリットがないので、スタンデールが犯人。

悪魔の足 (参考) 解と発言

- 本解 : 嘘つきが誰もいない場合
 - ストーリーとおなじ結論。
- 解なし : 嘘つきが1人いた場合。
 - 解なし。
- 別解 : 嘘つきが2人いた場合。
 - 別解1 : モーティマーとラウンディの共犯
 - ・ 嘘 : モーティマー「事件当日には、自分以外は兄妹だけいた。」
 - ・ 嘘 : ラウンディ「事件当夜は、教会地区にいた」
 - 別解2 : モーティマーとスタンデールの共犯
 - ・ 嘘 : モーティマー「事件当日には、自分以外は兄妹だけいた。」
 - ・ 嘘 : スタンデール「事件当日はアフリカにむけ出発するため遠隔地にいた。」
- 別解3 : 嘘つきが3人いた場合。
 - モーティマーとスタンデールとラウンディの共犯 (最大3人)
 - ・ 嘘 : モーティマー「事件当日には、自分以外は兄妹だけいた。」
 - ・ 嘘 : スタンデール「事件当日はアフリカにむけ出発するため遠隔地にいた。」
 - ・ 嘘 : ラウンディ「事件当夜は、教会地区にいた」

悪魔の足 まとめ

■ 解法のアプローチ

- 仮説推論により、事件現場状況から、ガス状の毒物を使用したという仮説が立った。
- ASPにより、時間・空間・場所やそこでの行動を、制約条件に整合するものから求めた。
- 特に、FACT情報のもととなっている登場人物の発言の真偽について探索した。

■ 結論

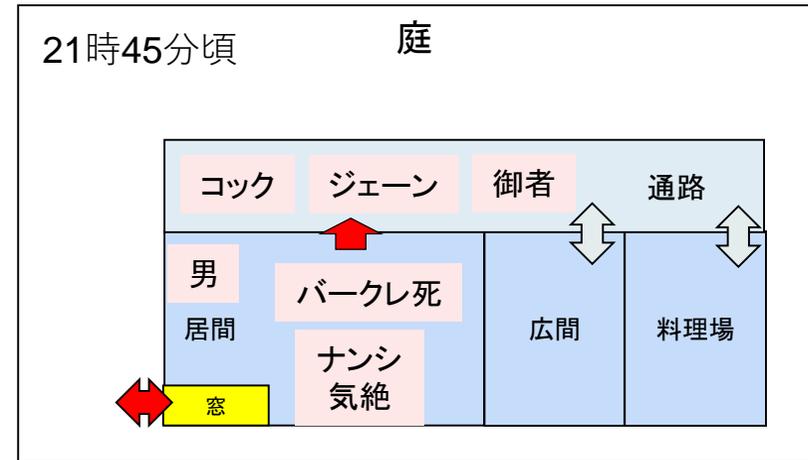
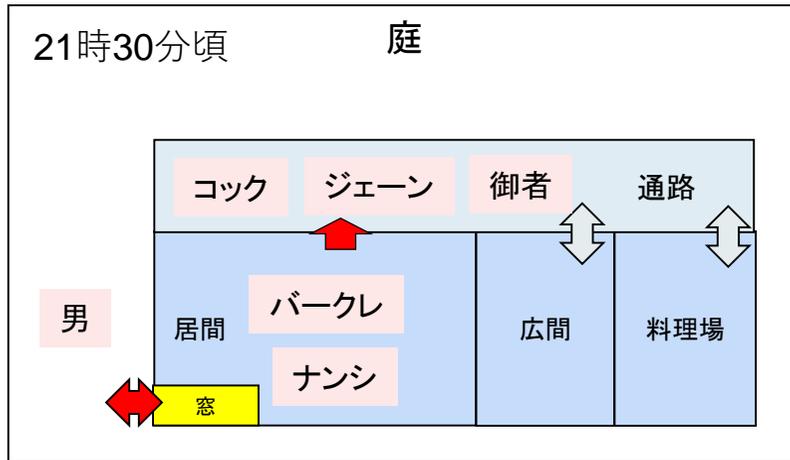
- 第1事件の犯人はモーティマー、第2事件の犯人はスタンデールという本編の結論を導くことができた。
- 証言の真偽を確かめて、真偽によって解の分岐を確認した。
- 別解として、以下のものが存在した。
 - ・ スタンデールが、モーティマーと共謀して第1の殺人を行い、第2の殺人でスタンデールがモーティマーを殺害する。
 - ・ ラウンディが、モーティマーと共謀して第1の殺人を行い、第2の殺人でスタンデールがモーティマーを殺害する。
 - ・ ラウンディとスタンデールが、モーティマーと共謀して第1の殺人を行い、第2の殺人でスタンデールがモーティマーを殺害する。
- ただし、スタンデールや司祭ラウンディが関与する動機については、明確なものが存在しない。
 - ・ 有力な理由としては羽振りの良いトレゲニス家から遺産を共謀して奪取するなどがある。
 - ・ スタンデールはブレンダと恋仲であることが今回の事件のきっかけというが、それすらスタンデールの発言によるものなので、確証が得られない。

■ 背中の曲がった男 あらすじ

- インドで活躍したイギリス軍の将校、ジェームス・バークレー大佐の怪死事件。
- 大佐の死体は、庭に出られるモーニングルームで発見された。事件のあった晩、お茶を持ってくるように言われた女中は、部屋に鍵がかけられ、大佐と夫人が激しく口論しているのを聞く。女中は料理女と馭者を呼びに行ったが、まだ激しい口論は続いていた。突然、恐ろしい男の叫び声とガシャンという音、そして女の悲鳴が聞こえたので、馭者は外の窓から部屋に入る。大佐は頭から血を流して死んでいて、その横で夫人が気を失って倒れていたのが発見される。
- 警察は夫人を第一容疑者として捜査を開始するが、使用人たちは口論中に夫人が「デービット！」と叫ぶのを聞いている。「デービット」なる男が事件にかかわっているのではないかという可能性も浮上していた。調査したところ、夫人が事件の起こった夜、ある背の曲がった男と会っていたという事実が分かる。その男に会った後から、バークレー夫人が急に夫を憎むようになったのだらうと推理する。ホームズはその男＝ヘンリー・ウッドを探し出し、事件の真相を聞き出す。
- その後、警察で解剖学的な結果が発表され、怪死の真相が心臓卒中であったことが判明する。そして、ホームズはワトソンに「デービット」が誰だったのか説明する。

背中曲がった男 犯行時の状況と各人の発言

屋敷には、バークリ大佐、ナンシー夫人、ジェーンメイド、御者、コック、“男？”がいた。



御者の発言例

- ・居間の窓は開いていた。
- ・居間は鍵がかかっていた。

ジェーン、コックの発言例

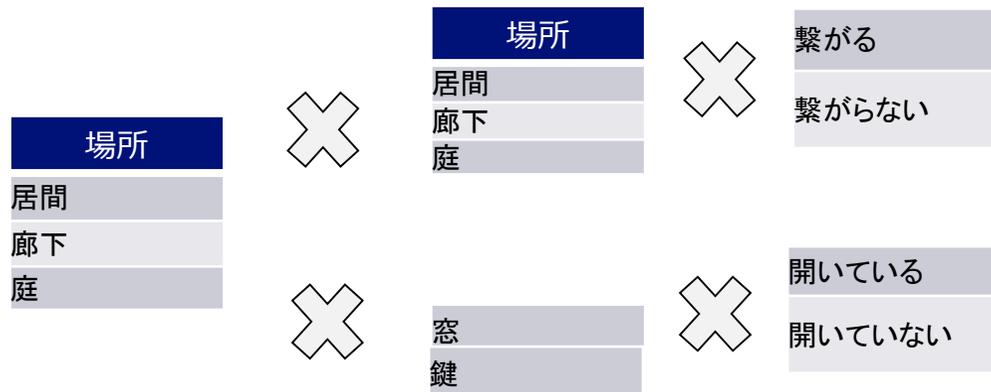
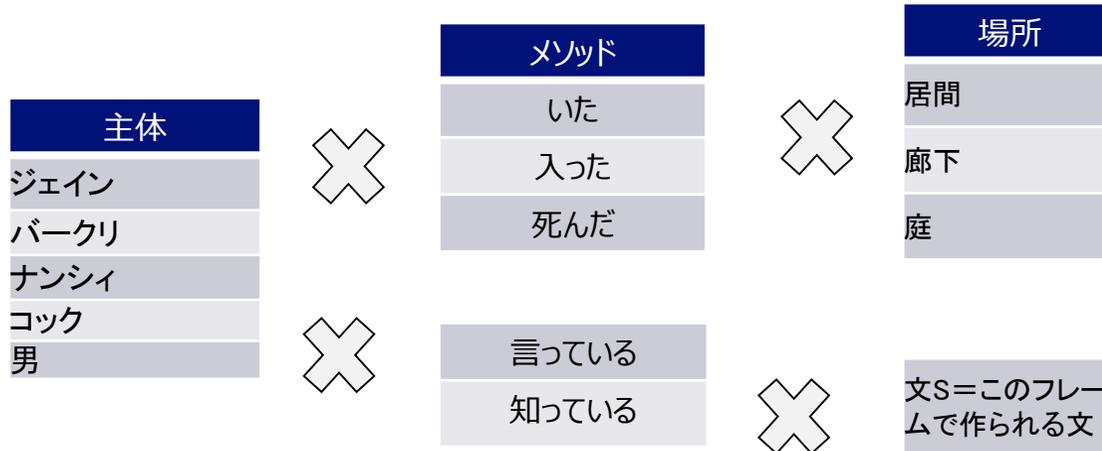
- ・ナンシーと旦那は言い争いをしていた。
- ・凶器の棍棒を知らなかった。

状況

- ・バークリ大佐が居間で死亡
- ・凶器は棍棒
- ・ナンシー夫人は居間で気絶中
- ・居間の鍵がない。

背中の曲がった男 密室殺人の犯行フレームと、組合せ文

「密室殺人」のフレーム例（実際に使ったのはもっと種類がある）



組合せ文の例

- ・居間の窓は開いている
- ・コックは「居間の鍵は開いていない」と言っている。
- ・ナンシィは居間にいた

背中の曲がった男 BERT類似文判定

次の2パターンで類似文判定を実施。

「フレームから生成した文」と「短編本文」の比較

「フレームから生成した文」と「ナレッジグラフ化時の短文」の比較

類似文判定結果は「背中の曲がった男類似文判定」エクセルを参照

結論

- ・仮説推論やASPの観測事実とファクトに必要な文は、類似度の高いものを採用。
しかし、手作業での追加も行った。類似度だけでは出来ない理由は、最後の課題で挙げる

背中の曲がった男 仮説推論

「男が庭から居間に侵入した」ことは、途中から出てきた仮説にすぎない。
よって、観測事実から、密室殺人事件の背景知識を持つ仮説推論エンジンから、この仮説を出す

■ 観測事実

短編から得られた観測事実	観測事実の論理式表現
動機のある男	motive(man)
バークレイは居間で死んだ	dead(Berkeley,Living)
居間の窓は開いていた	windowopen(Living)
庭は広い	locatbig(Garden)
庭に足跡がある	footprint(Garden)
居間は鍵がかかっている	keylock(Living,Rouka)

■ 密室殺人の仮説推論による仮説

```
{ "literal" : "invade(man, Living, Garden)", "state" : "satisfied" },    男が庭から居間に侵入
{ "literal" : "hide(man, Garden)", "state" : "satisfied" },
{ "literal" : "hide(man, Living)", "state" : "unsatisfied" },
{ "literal" : "(man = Berkeley)", "state" : "unsatisfied" },
{ "literal" : "murder(man, Berkeley, Living)", "state" : "satisfied" },    男がバークレイを居間で殺害
{ "literal" : "murder(Berkeley, Berkeley, Living)", "state" : "unsatisfied" } }
```

⇒ 男の庭から居間への侵入は、“証明”されたわけではなく、“仮説の1つ”に過ぎない。

■ 背中の曲がった男 各登場人物の発言の真偽による犯行解への影響

発言の真偽による、犯行解への影響の可能性

- 居間の鍵がかかっていた ⇒ 廊下から居間に入ることが可能
- 窓が開いていた ⇒ 庭から“男”が入ることが出来ない。
- ナンシーとバークリが言い争い ⇒ ナンシーの動機がなくなる
- 凶器を知らなかった ⇒ 犯行現場にいた場合は、凶器を知っているはずで、発言者が嘘つきになる。

★発言ではないが、ナンシー夫人が嘘つきだと、“気絶”が嘘の可能性はある

登場人物の発言

	居間の鍵	凶器	言い争い	窓が開いていた
ナンシー夫人	—	—	—	—
男	—	—	—	—
ジェーン (メイド)	鍵がかかっていた	知らなかった	言い争いしていた	—
御者	鍵がかかっていた	知らなかった	言い争いしていた	窓は開いていた
コック	鍵がかかっていた	知らなかった	言い争いしていた	—

背中の曲がった男 ASPで犯行を記述 その1

%短編の登場人物と時間と場所

```
person(berkeley;coachman;jane;man;cook;nancy).
time(0;1). %犯行前時間と、犯行時間のみ設定
locat(living;rouka;garden).
```

```
0 {motive(P):person(P)}n. %動機のある人の数がパラメーター
%被害者の設定
1 {victim(P):person(P)}1.
1 {victim_time(T):time(T)}1.
1 {victim_locat(L):locat(L)}1.
:- victim(P),motive(P).%被害者は動機を持たない
```

%嘘つきは複数。動機がある場合は嘘つきである。

```
0{liar(P):person(P)}.
liar(P):- motive(P).
motive(P):-liar(P).
```

%6人の登場人物は嘘つきか正直者かどちらか

```
R{honest(P):person(P);liar(P):person(P)}R :-
R=#count{person(berkeley;coachman;jane;man;cook;nancy)}.
:- honest(P),liar(P).
```

%前部屋と屋敷のトポロジーがつながっていれば、移動可能

```
1{person_at(P,T,L) :valid_move(Lp,L)}1 :- person(P),
time(T),person_at(P,T-1,Lp),locat(Lp).
```

%言い争いしていた二人は、動機がある

```
0{motive(P)}1:-argue(person(P),person(Q)).
0{motive(Q)}1:-argue(person(P),person(Q)).
```

%殺害事件は、被害者と動機がある人が同じ場所にて、かつ、凶器があると成立。この場合同時に、そこにいた場合は、殺人者とみなす

```
Q{murder(M,V,T,W,L):victim(V),victim_time(T),victim_locat(L),motive(M),person_at(M,T,L),
person_at(V,T,L),murder_method(W),not
murder_method(jiko),not sleep(M,T)}Q :- Q=#count{person(P):person_at(P,T,L),not
victim(P),not sleep(P,T),motive(P)}.
```

%事故の場合は自分で死ぬ。

```
1{murder(V,V,T,W,L):victim(V),victim_time(T),victim_locat(L),person_at(V,T,L),murder_method(W),not sleep(V,T)}1:-murder_method(jiko).
```

%殺害事件現場には、被害者以外の正直者が1人もいないこと（寝ていないことが前提）

```
{murder(M,V,T,W,L):victim(V),victim_time(T),victim_locat(L),motive(M),person_at(M,T,L),person_at(V,T,L),murder_method(W)}=0
:- 2{honest(P):person_at(P,T,L),victim_time(T),victim_locat(L),not sleep(P,T)}.
```

%Pが嘘つきならば、Pの発言内容は、成立するかしないかどちらかになる。ただしm=1の場合は嘘つきの発言は必ず嘘になる。

```
m{holds(-S)}1 :- says(P,S,1),liar(P).
```

%Pが嘘つきならば、Pの否定発言は、成立するかしないかどちらかになる。ただしm=1の場合は嘘つきの発言は必ず嘘になる。

```
m{holds(S)}1 :- says(P,S,0),liar(P).
```

%Pが正直ものならば、Pの発言は、本当になる。

```
holds(S) :- says(P,S,1),honest(P).
```

%Pが正直ものならば、Pの否定発言は、成立しない。

```
holds(-S) :- says(P,S,0),honest(P).
```

%犯行現場にいたものは、凶器について知っている。

```
know(P,murder_method(W)):-
```

```
victim_time(T),victim_locat(L),murder_method(W),person_at(P,T,L).
```

%凶器が何だったか知っているものは、凶器となった物体について知っている。

```
know(P,W):- know(P,murder_method(W)).
```

```
%know(P,murder_method(W)):- know(P,W).
```

```
%ファクト
%各人の居場所アリバイ
person_at(nancy,0,living).
person_at(nancy,1,living).

person_at(berkeley,0,living).
person_at(berkeley,1,living).

person_at(jane,0,rouka).
%person_at(jane,1,living).

person_at(coachman,0,rouka).
%person_at(coachman,1,garden).

person_at(cook,0,rouka).
%person_at(cook,1,rouka).

%仮説推論で男の存在が出たので、その追加
person_at(man,0,garden).
%not person_at(man,2,living).

%屋敷のトポロジー
valid_move(living,living).
valid_move(rouka,rouka).
%valid_move(rouka,living).←鍵がかかっている発言の真偽
%valid_move(living,garden).←窓は開いていた発言の真偽
%valid_move(garden,living).←窓は開いていた発言の真偽
valid_move(garden,garden).
valid_move(rouka,garden).
```

```
%被害者と被害時刻と場所
victim(berkeley).
victim_time(1).
victim_locat(living).
1{murder_method(stick;jiko)}1.%死因は打撲か、事故
```

```
%夫人は時刻1で気絶していたか、そうでないか
1{sleep(nancy,1)}1 :- honest(nancy).
%夫人が嘘つぎの場合は、夫人は気絶しない。
0{sleep(nancy,1)}0 :- liar(nancy).
```

```
%言い争いについての発言
says(jane,argue(berkeley,nancy),1).
says(cook,argue(berkeley,nancy),1).
says(coachman,argue(berkeley,nancy),1).
```

```
%リビングの鍵が開かなかったことの発言
says(coachman,valid_move(rouka,living),0).
says(jane,valid_move(rouka,living),0).
says(cook,valid_move(rouka,living),0).
```

```
%御者の窓が開いていた発言
says(coachman,valid_move(living,garden),1).
says(coachman,valid_move(garden,living),1).
```

```
%廊下にいたならば、リビングに入る。(緊急事態)
person_at(P,1,living):-valid_move(rouka,living),person_at(P,0,rouka).
```

```
%棍棒を知らなかった発言
says(jane,know(jane,stick),0).
says(coachman,know(coachman,stick),0).
says(cook,know(cook,stick),0).
```

```
%ナンシーは口論の動機が証明されない場合は、動機なし
not motive(Q):--argue(P,Q).
not motive(P):--argue(P,Q).
```

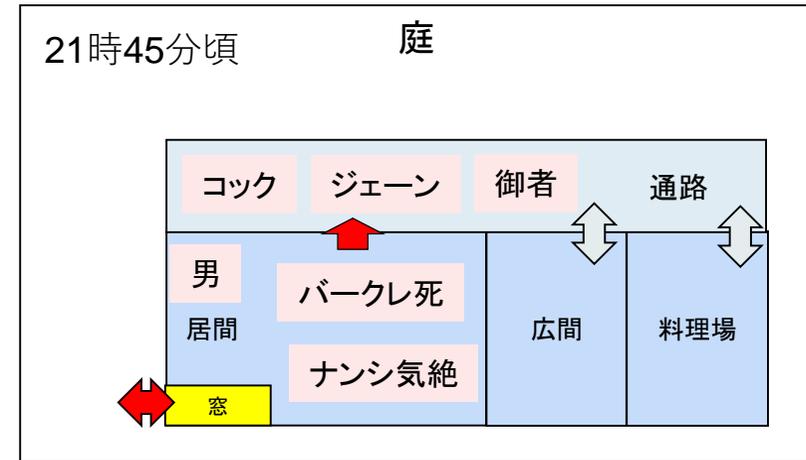
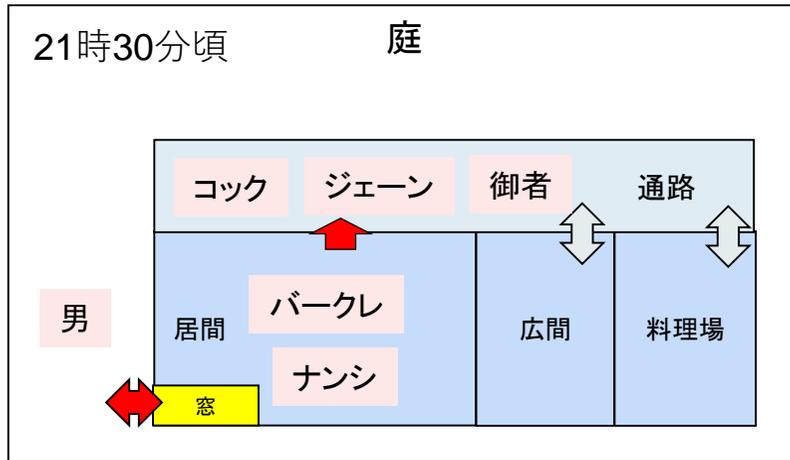
背中の曲がった男 ASPの結論とその解釈

- ・警察の検死により、卒中による死亡と判断され、本解が一意に出てくる
- ・しかし、もし鈍器による後頭部打撲が致命傷ならば、別解が出てくる。

	解の解釈	ASPの解
本解	バークリ大佐は、卒中で死亡	sleep(nancy,1) murder(berkeley,berkeley,1,jiko,living)
別解①	ナンシーは嘘つきで、ナンシー夫人の単独犯	liar(nancy) murder(nancy,berkeley,1,stick,living)
別解②	ナンシーは嘘つきで犯人、“男”がぐる。	liar(nancy) liar(man) murder(nancy,berkeley,1,stick,living) murder(man,berkeley,1,stick,living)
別解③	ナンシーは正直。“男”は嘘つき = 動機があり、“男”の単独犯行。	liar(man) sleep(nancy,1) murder(man,berkeley,1,stick,living)
別解④	コック、召使い、御者が嘘つきで犯人。 ナンシーは正直で気絶。 窓が開いていた、鍵がかかっていたが、両方とも嘘。	liar(cook) liar(jane) liar(coachman) sleep(nancy,1) murder(cook,berkeley,1,stick,living) murder(coachman,berkeley,1,stick,living) murder(jane,berkeley,1,stick,living)
別解⑤	(嘘つきが、正直発言と嘘発言を混ぜる場合) 全員 (ナンシー夫人、御者、召使い、コック) が犯人	liar(cook) liar(jane) liar(coachman) liar(nancy) murder(nancy,berkeley,1,stick,living) murder(cook,berkeley,1,stick,living) murder(coachman,berkeley,1,stick,living) murder(jane,berkeley,1,stick,living)

背中曲がった男 ASPの結論 本解

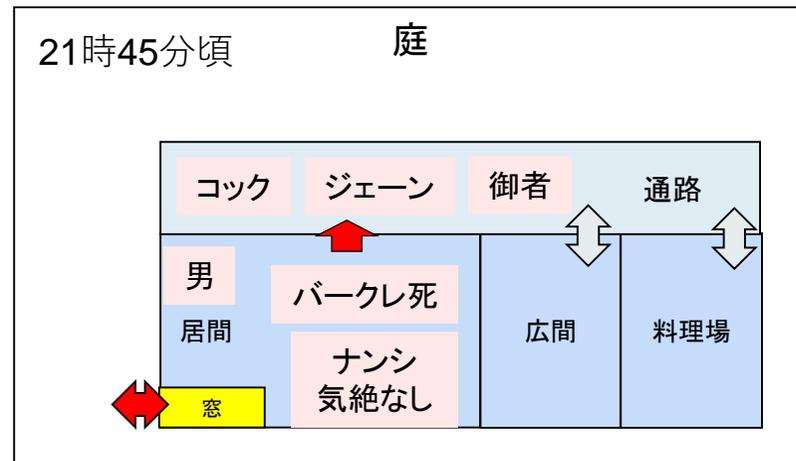
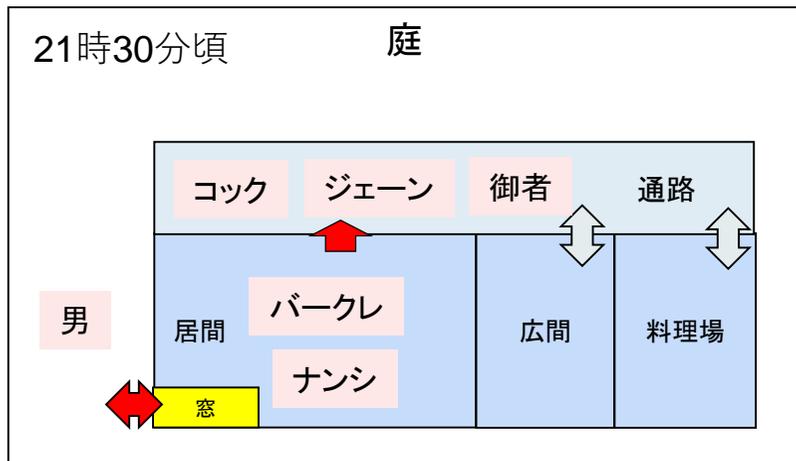
本解①バークリは、事故（卒中）で死亡



	正直嘘つき	居間の鍵	凶器	言い争い	窓が開いていた
ナンシー夫人	正直⇒気絶	—	—	—	—
男	(正直)	—	—	—	—
ジェーン (メイド)	正直	居間は鍵がかかっていた	知らなかった	言い争いしていた	—
御者	正直	居間は鍵がかかっていた	知らなかった	言い争いしていた	居間の窓は開いていた
コック	正直	居間は鍵がかかっていた	知らなかった	言い争いしていた	—

背中の曲がった男 ASPの結論 別解①②

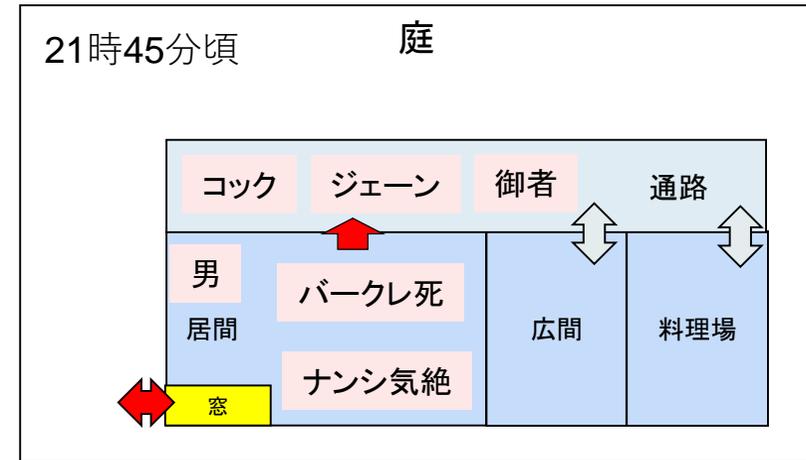
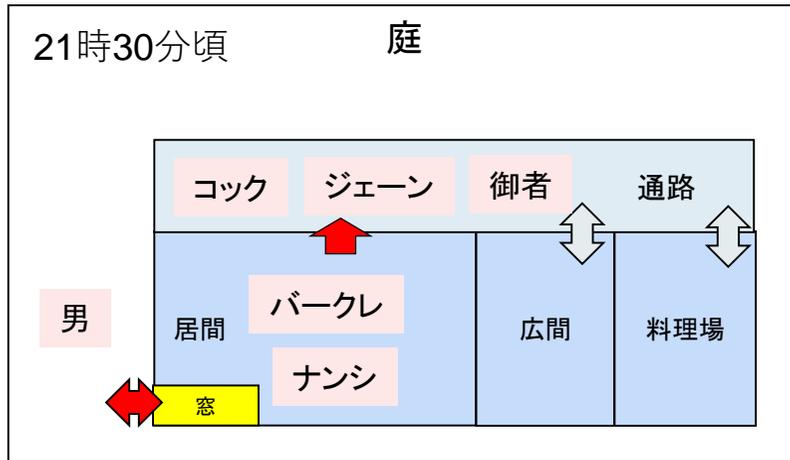
別解①ナンシー夫人の単独犯、または、別解②ナンシーと男の共犯



	正直嘘つき	居間の鍵	凶器	言い争い	窓が開いていた
ナンシー夫人	嘘つき⇒気絶も嘘	—	—	—	—
男	(嘘つき)	—	—	—	—
ジェーン (メイド)	正直	鍵がかかっていた	知らなかった	言い争いしていた	—
御者	正直	鍵がかかっていた	知らなかった	言い争いしていた	居間の窓は開いていた
コック	正直	鍵がかかっていた	知らなかった	言い争いしていた	—

背中の曲がった男 ASPの結論 別解③

別解③男の単独犯

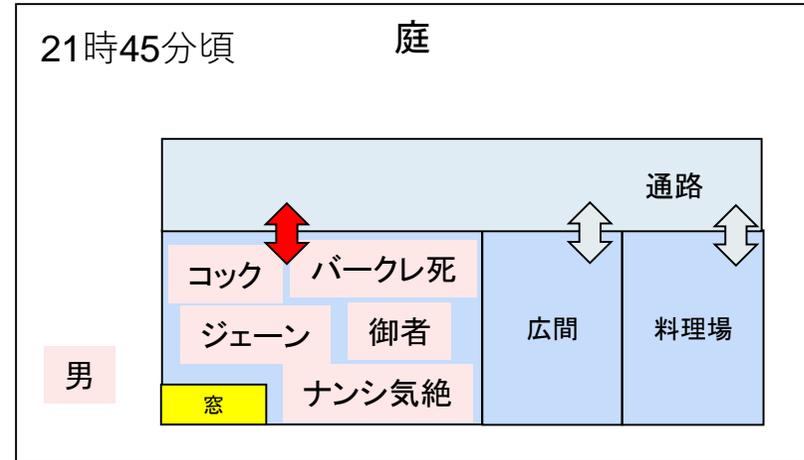
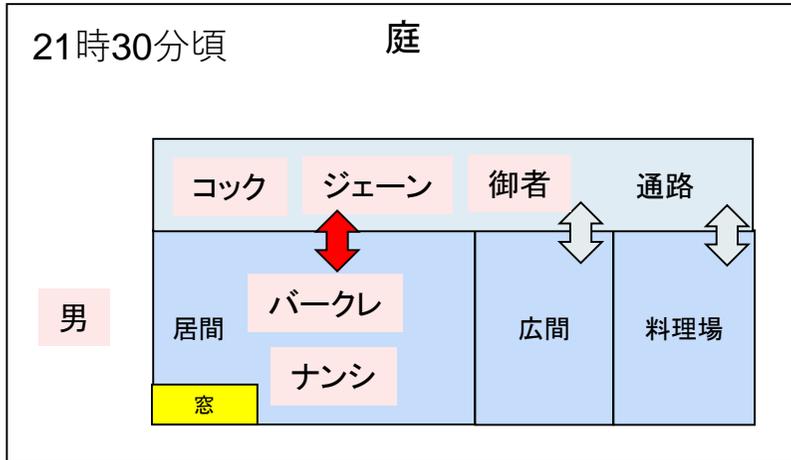


	正直嘘つき	居間の鍵	凶器	言い争い	窓が開いていた
ナンシー夫人	正直⇒気絶	—	—	—	—
男	嘘つき・正直問わず	—	—	—	—
ジェーン (メイド)	正直	鍵がかかっていた	知らなかった	言い争いしていた	—
御者	正直	鍵がかかっていた	知らなかった	言い争いしていた	窓は開いていた
コック	正直	鍵がかかっていた	知らなかった	言い争いしていた	—

背中の曲がった男 ASPの結論 別解④

別解④ ジェーン、御者、コックの3人の共同犯行

- ・「居間に鍵がかかっていた」が嘘より、ジェーン・御者・コックが居間に侵入可能
- ・「窓が開いていた」が嘘より、男は侵入できず。
- ・3人は同じ発言をしているため、1人ないし、2人だけの嘘はつけない。

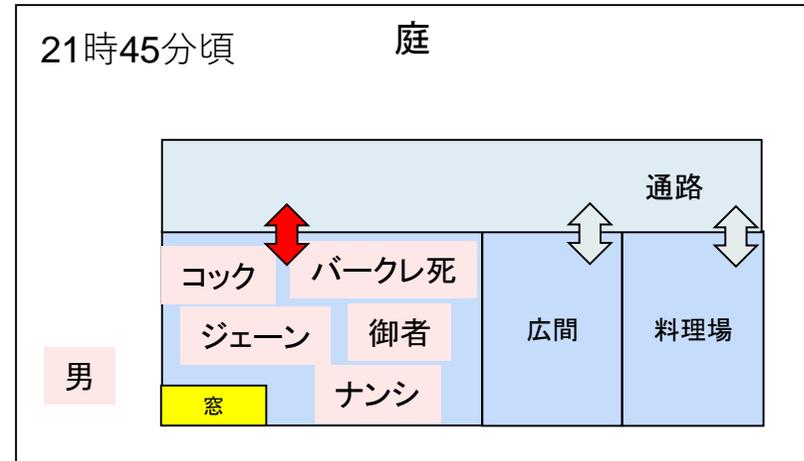
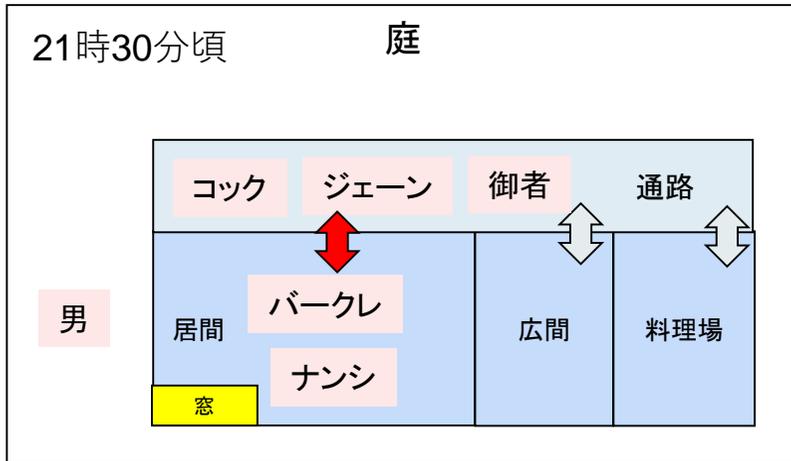


	正直嘘つき	居間の鍵	凶器	言い争い	窓が開いていた
ナンシー夫人	正直⇒気絶	—	—	—	—
男	—	—	—	—	—
ジェーン (メイド)	嘘つき	鍵がかかっていた⇒鍵は開いていた	知らなかった⇒知っていた	言い争いしていた⇒ナンシーに動機がなくなる⇒正直	—
御者	嘘つき	鍵がかかっていた	知らなかった⇒知っていた	言い争いしていた	窓は開いていた⇒窓は閉まっていた⇒男は侵入できず
コック	嘘つき	鍵がかかっていた	知らなかった⇒知っていた	言い争いしていた	—

背中の曲がった男 ASPの結論 別解⑤ 嘘つきが気まぐれ

別解⑤ 全員（ナンシー夫人、御者、ジェーン、コック）が、犯人

嘘つきが、嘘と本当を混ぜる場合とすると、「言い争いしていた」が本当かも知れない。



	正直嘘つき	居間の鍵	凶器	言い争い	窓が開いていた
ナンシー夫人	嘘つき⇒気絶なし	—	—	—	—
男	—	—	—	—	—
ジェーン (メイド)	嘘つき	鍵がかかっていた⇒鍵は開いていた	知らなかった	言い争いしていた⇒ナンシーに動機なしと必ずしも言えず	—
御者	嘘つき	鍵がかかっていた	知らなかった	言い争いしていた	窓は開いていた⇒窓は締まっていたとは限らない。
コック	嘘つき	鍵がかかっていた	知らなかった	言い争いしていた	—

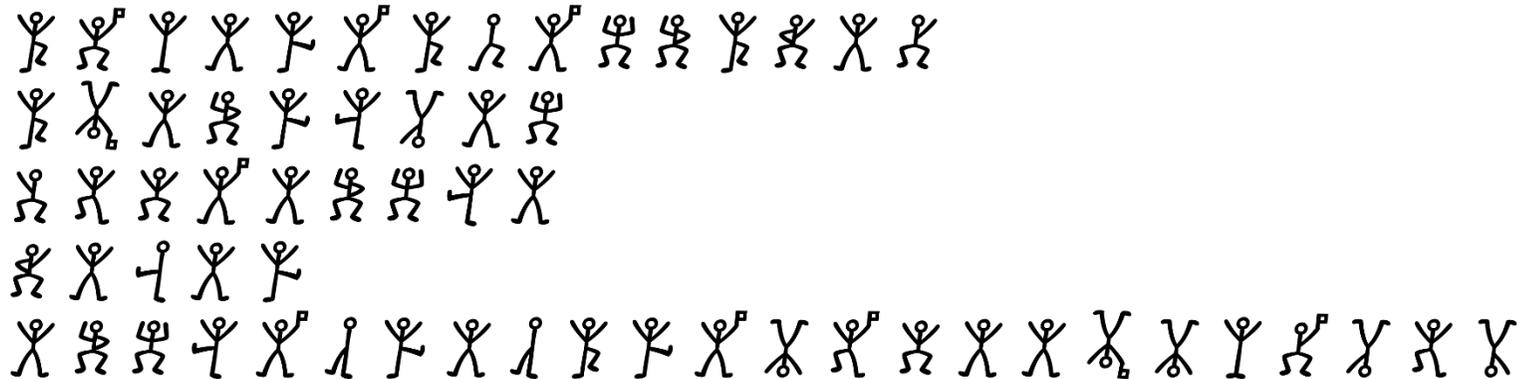
■ 背中の曲がった男 ASPの各解を一意にする条件

本解含む、各解を一意にする条件

	解の解釈	解を一意にする条件
本解	バークリ大佐は、卒中で死亡	<ul style="list-style-type: none"> ・「犯行の凶器は棍棒」が否定されること
別解①	ナンシーは嘘つきで、ナンシー夫人の単独犯	<ul style="list-style-type: none"> ・居間の鍵が開かず、窓が閉まっていたことの証明
別解②	ナンシーは嘘つきで犯人、“男”がぐる。	<ul style="list-style-type: none"> ・コックたち3人の発言に関わらず、居間の鍵が開かないこと ・窓が開いていたことの証明
別解③	ナンシーは正直。“男”は嘘つきで“男”の単独犯	<ul style="list-style-type: none"> ・窓が開いていたこと ・ナンシーの気絶が嘘でないことの証明
別解④	コック、召使い、御者が嘘つきで犯人。 ナンシーは正直で気絶。 窓が開いていた、鍵がかかっていたが、両方とも嘘。	<ul style="list-style-type: none"> ・ナンシーの気絶が嘘でないことの証明 ・御者、召使い、コックが凶器の棍棒を知っていたこと または、鍵がかかっていたが嘘であることの証明
別解⑤	(嘘つきが、正直発言と嘘発言を混ぜる場合) 全員(ナンシー夫人、御者、召使い、コック)が犯人	嘘つきが4人であること (発言に嘘と正直が混ざると、条件を一意にするのは難しい)

踊る人形 あらすじ

- 人形の形をした絵が、犯人から依頼者への秘密のメッセージになる。
- ホームズは、そのメッセージを解読することに成功。逆に、暗号を使ったメッセージで犯人を呼び出す。

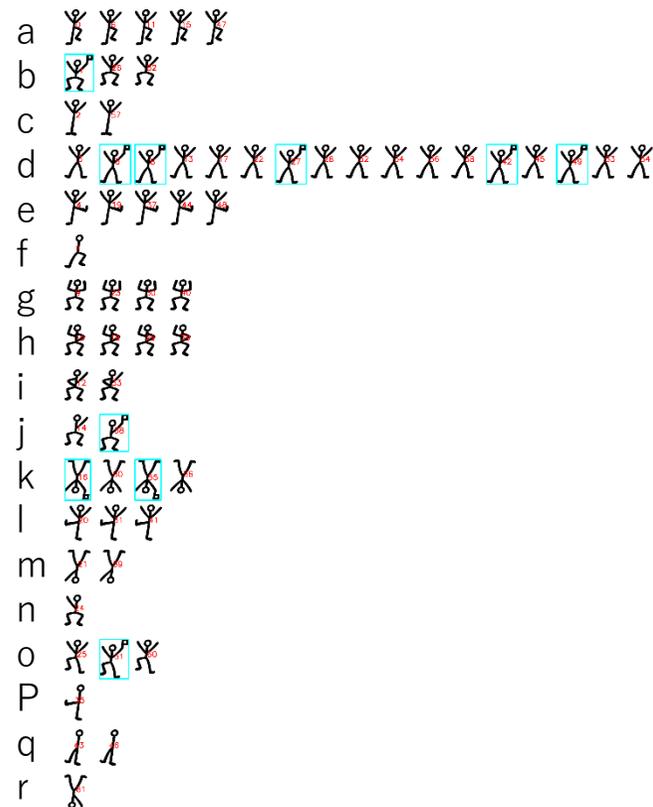
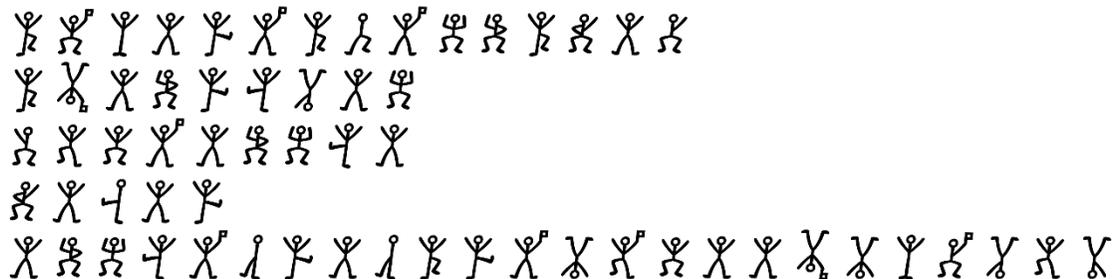


踊る人形

ステップ1) 人形図の同一性を画像認識AIを使って判別する。

ステップ2) 人形図をローマ字一文字に変換して、換字暗号解く。

人形図



①同一視された人形図は同じ文字として、“暗号文字列”に変換

ab cded afd ghaidj

ak dhelmdg

nobd dhgld

idpde

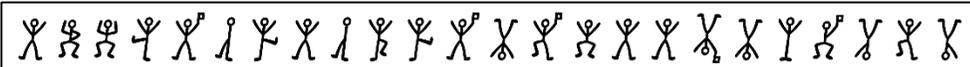
dhgld qedqaed ko bddk kcj mor

②単換字暗号解読プログラムにて、暗号解読。

am here abe slaney at elriges come elsie never elsie prepare to meet thy god

踊る人形 画像認識タスクの概要

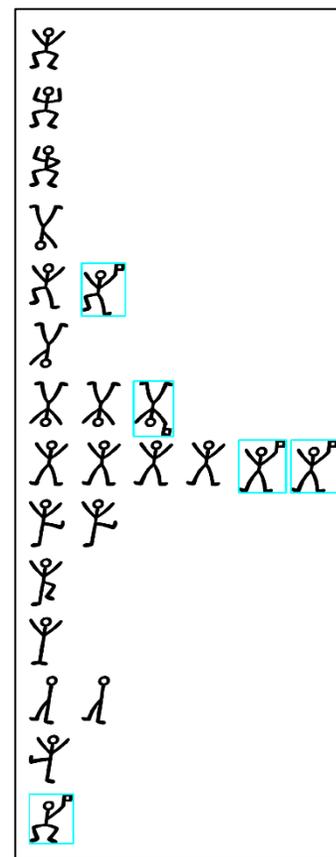
- 各人形図の同一性を画像認識技術を使って判別するタスク。



入力イメージ

画像処理

出力イメージ



考慮すべき条件

- 人形図は初めて見るため事前学習はできない。
- 手に旗を持つ人形図があるが、持たない人形図と同一クラスとして認識する必要がある。

問題の簡易化

- 人形図は原著の挿絵ではなく、オープンソースフォントからイメージを作成。⇒手法1（ピクセル比較）で認識
- 人形図は原著の挿絵を用いるが、全暗号ではなく最後の1文のみを利用。⇒手法2（特徴量比較）で認識

利用したライブラリ

- OpenCV 4.1 (python)

- 人形図をクラス分類して縦に並び替える。
- 手に旗を持った人形図は四角形で囲む。

踊る人形 画像認識手法 1 (ピクセル比較)

■ 標準化された人形図で旗を含め暗号文全てを認識させる

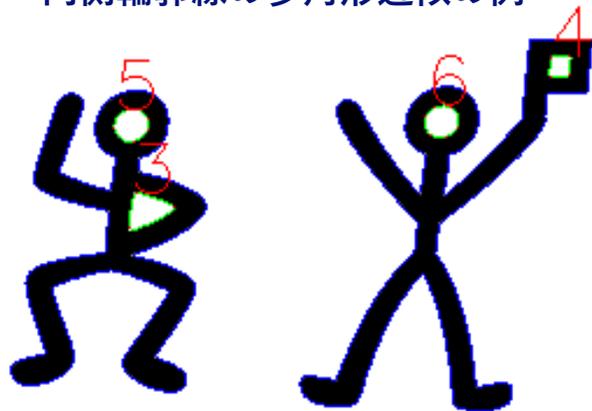
出所：GL-DancingMen/Gutenberg Labo/2011

※MS Wordで作成し、拡大したイメージデータを作成

■ 輪郭線検出と矩形認識

- 外側と内側の2階層で輪郭線を検出し、外側輪郭線で人形を識別。
- 内側輪郭線の内面積が小さすぎるものはノイズとして除外。
- 内側輪郭線をDouglas-Peuckerアルゴリズムで近似、許容距離(ϵ)は輪郭線長から与える。
- 4点で近似される内側輪郭線を旗とみなす。

内側輪郭線の多角形近似の例

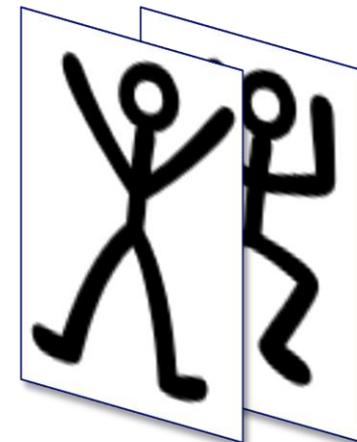


■ 画像類似度を算出しクラス分類

- 外側輪郭線の外接矩形を求め、暗号画像から人形を1体ずつ切り取り。
- 旗を持っている人形は旗を取り除き、比較用の固定サイズにリサイズ。
- 画像比較はピクセル単位で乗算し、総和を求めて正規化。

$$result(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x+x', y+y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x+x', y+y')^2}}$$

- 一定以上の類似度を持つ画像を同一クラスとみなし、分類後の画像を作成する。



踊る人形 画像認識手法 2 (特徴量比較)

■ 原著の手書き人形図の暗号文を認識させる (最後の1文のみで旗認識なし)

出所：https://commons.wikimedia.org/wiki/Category:The_Adventure_of_the_Dancing_Men

※SVGデータから拡大画像を作成、処理し易いように一部を加工

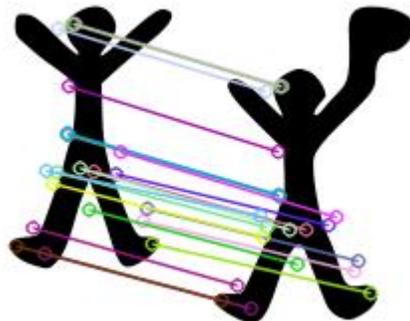
■ 特徴点計算とクラス分類

- 人形図の特徴点抽出はAlcantarilla他考案のKAZEアルゴリズムを利用。
- 人形図ごとに全ての特徴点のユークリッド距離を総当たりでマッチング。
- マッチした特徴点ごとに最も近い上位2点との距離が一定割合近い特徴点をカウント。
- カウント数が一定数以上ある人形図を同一クラスとして分類。

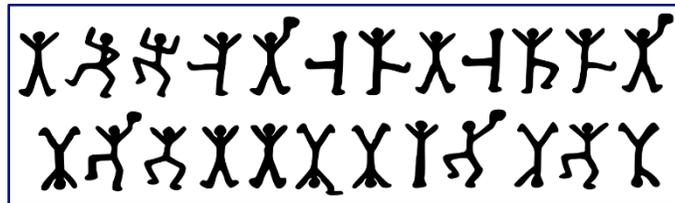
KAZEによる特徴点



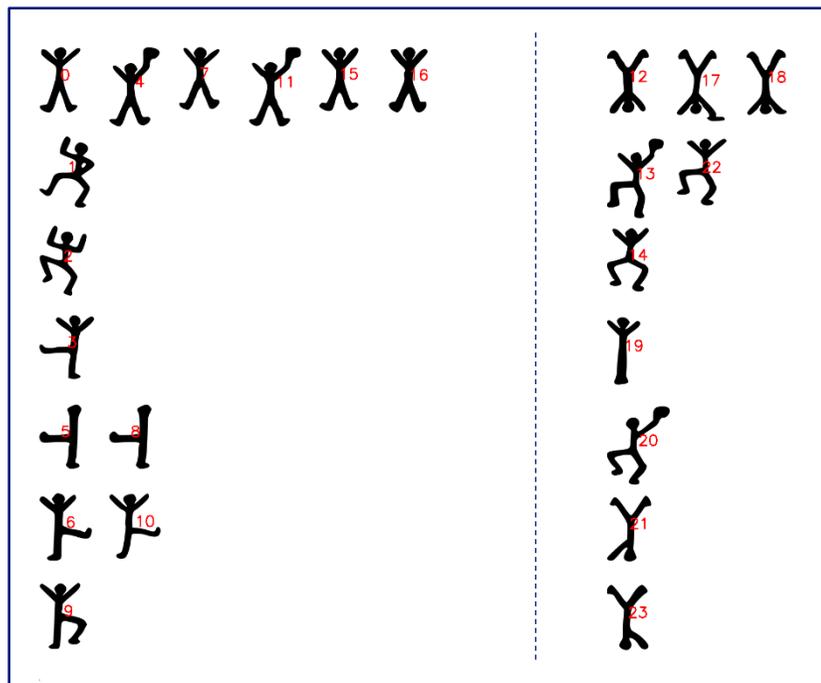
特徴点マッチングの例



入力画像 画像認識結果



出力画像



人形上の数字は元画像での並び順

踊る人形 単換字暗号解読

- 単換字暗号解読プログラムにて、暗号解読。

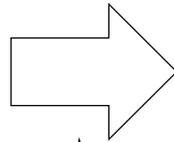
ab cded afd ghaidj

ak dhelmdg

nobd dhgld

idpde

dhgld qedqaed ko bddk kcj mor



am here abe slaney

at elriges

come elsie

never

elsie prepare to meet thy god

単換字暗号解読
キーは総当たり

- ポイント

- ・ 換字暗号は、1つの解読キーで変換後、該当する単語があるかワードリスト利用する
ワードリストに「踊る人形」の本文の単語 **elsie abe slaney elriges** などを入れた。
- ・ 総当たり方法でも、**CPU**マシンで、**30秒弱**で解読可能。

AIの結論と犯行解を“一意”にするための条件、残った疑問

短編	AIの結論 + 犯行解を一意にするための条件	残った疑問 (AIにいった不自然なファクトなど)
同一事件 (花婿失踪事件)	<ul style="list-style-type: none"> ・ホズマは、サザランドに対して、結婚詐欺をした。 ・ホズマと継父は同一人物。 	継父の偽装で、花嫁を欺くことができるだろうか？
まだらの紐	<ul style="list-style-type: none"> ・結論 ロイロットがジュリア殺人の犯人 ・別解 ヘレンが犯行に加わったことは否定できない。ヘレンに殺意がないことの証明 	蛇の毒は、体内に残らない？ (体内から毒が検出されるとドリンク系薬物による犯行の可能性が仮説推論でであるため)
悪魔の足	<ul style="list-style-type: none"> ・結論 モーティマーとスタンデルが犯人 ・別解 スタンデル、司祭ラウンディが別個にあるいは両方ともがモーティマーと共謀していた可能性 <p>(結論の分岐を生む重要な証言の裏を取れば一意になる)</p>	<ul style="list-style-type: none"> ・犯行現場に近かったポータ夫人を無条件に信じ込んでいた根拠はなにか。 ・犯人の自白とはいえ、弁明を無条件で信じてしまうのは不自然ではないか。
背中の曲がった男	<p>本解 バークリ大佐は、卒中で死亡</p> <p>別解① ナンシー夫人単独</p> <p>別解② ナンシー夫人単独または“男”がグル</p> <p>別解③ “男”の単独犯行</p> <p>別解④ コック、召使い、御者のグルで犯行 (一意にするための条件は上述)</p>	<ul style="list-style-type: none"> ・本当に、バークリは卒中だったか。別解の犯行解の方が自然と思える。
踊る人形	<ul style="list-style-type: none"> ・暗号の解読結果 <p>am here abe slaney at elriges come elsie never elsie prepare to meet thy god</p> <p>暗号を解読できたエイブ・スレニーが犯人</p>	—

目指したことへの達成度と課題 チャレンジ①②

チャレンジ① シャーロックの鋭い観察による推理 (= 仮説推論) を、真似できるか？

⇒ ilpベースの仮説推論は機能する。特に、密室殺人に対して同じ背景知識で対応できることが実証

⇒ 事件のケースを増やす (= 経験を積む) と、より幅広いパターンに対応可能になる。

⇒ 今回は、仮説推論への観測事実のインプットは、フレームに合致する文にのみ絞った。

しかし、全文 (= 対応する述語論理式) を観測事実に入れても、同じ結果になると思われる。

チャレンジ② 短編の“不完全さ”を補うことができるか？ 犯行解を一意にする条件の提示

⇒ ASPは充足可能問題を含み解くため、犯行可能性解の列挙が可能。一意にする条件の提示も容易。

目指したことへの達成度と課題 チャレンジ③

チャレンジ③ 登場人物の発言の嘘つき問題を考慮して、犯行が解けるか？

⇒ ASPでは、正直者と嘘つきの発言を真偽を全パターン考慮した解の探索が行うことで、嘘つき問題をうまく解くことが可能

⇒ 嘘つき問題では、次の2パターンを考慮した

A) 正直者が話すことは常に正しい、嘘つきが話すことは常に誤り

B) 正直者の“勘違い”や、嘘つきが正直発言と嘘発言を混ぜる場合を考慮

- ・ Bの場合は、組合せパターンが増えすぎて、解の絞りこみが困難になる。
- ・ ところが、殺人事件の創作物では、犯行者は“クリティカルなところ”のみ嘘をつくBが一般的だ。
- ・ クリティカル（=犯行解が変わる）発言の特定と、その真偽のみ考慮して充足可能問題を解くことが必要

⇒今回は、すべての人物の嘘つき度合いは同等としたが、過去の発言の信頼度などから、重みの設定も可能
例) 次郎は太郎の2倍、三郎は太郎の5倍嘘つき度が高い。

maximize { 1:liar(taro); 2:liar(jiro); 5:liar(saburou) }

目指したことへの達成度と課題 チャレンジ④

チャレンジ④ “背景知識”+日本語文のみ（ナレッジグラフを介さず）で推論できるか？

⇒ “ナレッジグラフを作る際にできた短文“に対しては、BERTの類似度判定は有効

⇒ 短編本文との類似判定は難しい。

難しい理由

1) 照応解析・省略の補いの困難さ

主語などの省略を補う照応解析の難しさ、前文からの説明の引き継ぎ

例) 10月27日の記述後「翌日の朝を迎え」で「28日」を指すなど

2) 1つの文から、複数の必要なファクトが、導かれる、推論に必要。

そのルールを追加する必要がある。

例「AはBに話かけた」→AとBが同じ場所にいる。

例「AはBと会った」→AとBは、その時点で生きている。

例「Aは9時に廊下から居間に入った」→Aは9時前に廊下にいた。Aは9時以降に居間にいた。

3) 技術的相性

BERTの類似文判定は、短文よりも、ある程度の長い文の方が向いている。

目指したことへの達成度と課題 チャレンジ⑤

チャレンジ⑤ 踊る人形の画像認識は、事前学習なしで挑む。

⇒内側輪郭線の多角形近似や特徴点計算とクラス分類のチューニングにより、人間の目と脳が行っているような、初見の画像についても、学習なしで分類可能なことが分かった。

⇒活字よりも難しい、原著の手書き人形図の暗号文解読も達成。

提出ファイル一覧

①BERT類似文判定

プログラム similarity6.py

類似比較文対象ファイル

- ・背中の曲がった男 crookedman_kg.txt target-crookedman_kg.txt
- ・花婿失踪事件 hanamuko_kg.txt target-hanamuko.txt

②仮説推論(Open-David形式ファイル)

- ・悪魔の足 devil_foot.dav
- ・背中の曲がった男 crookedman.dav
- ・花婿失踪事件 hanamuko.dav
- ・まだらの紐 speckledband.dav

③ASP(Clingo形式ファイル)

- ・悪魔の足 devil_foot.lp
- ・背中の曲がった男 crookedman.lp

④画像認識

プログラム dancingman1.py,dancingman2.py

画像ファイル dancingman1.png, dancingman2.png

⑤単換字暗号解読

プログラム decrypt.py

単語リストファイル 0.txt

暗号文ファイル encrypted.txt

各プログラムのインストールの仕方 動かし方

①BERT類似文判定

インストールはややこしいので別紙

②仮説推論

- ・ open-davidを利用。 open-david本体のインストールには一部難しい部分あり（詰まったら聞いてください）
- ・ <https://github.com/aurtg/open-david> ライブラリー lp_solve 5.5を利用

・ 実行

```
bin/david infer -C -c simple,weighted,lpsolve -k compiled/kb -v 5 -o full --max-depth=9 ファイル名.dav
```

③ASP

・ clingo本体のインストール方法 <https://potassco.org/>

・ 直接Web版で、ファイルを動かすことも可能 <https://potassco.org/clingo/run/>

・ 実行

```
$ clingo crookedman.lp 0 -c n=6 -c m=1 -project nは嘘つき人数のマックス数
```

```
$ clingo crookedman.lp 0 -c n=6 -c m=0 -project mを0にすると、嘘つきは正直と嘘を混ぜる
```

(直接Web版で動かす場合には、パラメーター設定、n、mの書き換えが必要)

```
$ clingo devil_foot.lp 0 -project
```

各プログラムのインストールの仕方 動かし方 その2

④画像認識

インストール pythonモジュール OpenCV4 (4.1.2.30)

```
$ pip3 install opencv-python
```

実行

```
$ python3 dancingman1.py
```

```
$ python3 dancingman2.py
```

⑤単換字暗号解読

参考にしたオープンソース https://github.com/alexbers/substitution_cipher_solver

実行

```
$ python3 decrypt.py
```

①BERT類似文判定の実行環境セットアップ1

■ Juman++のインストール (Ubuntu-18.04の例)

- gccとcmakeの準備

```
$ sudo apt -y install build-essential
```

```
$ sudo apt -y install cmake
```

- ソースコードをダウンロード

```
$ wget https://github.com/ku-nlp/jumanpp/releases/download/v2.0.0-rc3/jumanpp-2.0.0-rc3.tar.xz
```

```
$ tar xvf jumanpp-2.0.0-rc3.tar.xz
```

(2019/11時点ではV2正式版は未リリースです)

- コンパイルとインストール

```
$ cd jumanpp-2.0.0-rc3/
```

```
$ mkdir build
```

```
$ cd build/
```

```
$ cmake .. -DCMAKE_BUILD_TYPE=Release
```

```
$ make
```

```
$ sudo make install
```

①BERT類似文判定の実行環境セットアップ2

■ pythonモジュールのインストール

- pyknp (0.4.1)

- \$ pip3 install pyknp

- pytorch_transformers (1.1.0)

- \$ pip3 install pytorch_transformers

- (未導入の場合numpyとtorchもインストールされます)

■ 各種データセットのダウンロード

- BERT日本語Pretrainedモデル

- \$ wget http://nlp.ist.i.kyoto-u.ac.jp/DLcounter/lime.cgi?down=http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/JapaneseBertPretrainedModel/Japanese_L-12_H-768_A-12_E-30_BPE.zip&name=Japanese_L-12_H-768_A-12_E-30_BPE.zip

- \$ sudo apt -y install zip unzip

- \$ unzip Japanese_L-12_H-768_A-12_E-30_BPE.zip

- \$ cd Japanese_L-12_H-768_A-12_E-30_BPE/

- \$ ln -s bert_config.json config.json

アドバイスを頂いた、大学・研究所の方々

■ お茶の水女子大学

- 戸次大介准教授

■ 神戸大学

- 田村直之教授
- 宋剛秀准教授

■ 名古屋大学

- 番原睦則教授

■ 北陸科学技術先端大学

- 青木利晃教授

The text is framed by two decorative swooshes. The top swoosh is a gradient bar transitioning from blue on the left to red on the right. The bottom swoosh is a solid blue bar.

Share the Next Values!