

雑談対話システム構築フレームワーク PyChatに基づく 特定シチュエーション向け対話システム

中島圭祐*1 駒谷和範*1 中野幹生*2

*1 大阪大学 産業科学研究所

*2(株) ホンダ・リサーチ・インスティテュート・ジャパン

1.1 背景

- ▶ 評価軸に合わせて「人らしい」対話システムの構築を目標とした
- ▶ 実働期間が約6週間での作成

1.2 課題

- ▶ 分岐の無い1通りの応答だけを行う場合
 - 破綻する可能性
 - ユーザの話を聞いていないシステム

1通りの応答を行った例

S1: 印象に残った場所ってどこ？
U1: イギリスかな

S2: そっかあ。いつ行ったの？

S1: 印象に残った場所ってどこ？
U1: **去年**行ったハワイかな

S2: そっかあ。いつ行ったの？

既に言った事を聞いている

1.言語理解が必要

短い期間での実装が困難

1.2 課題

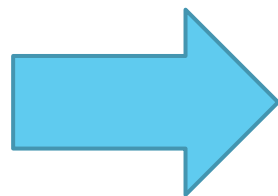
- ▶ 1問1答型のシステム
 - ◆ 唐突な話題転換をする
 - ▶ 文脈を無視している

唐突な話題転換の例

S1: 印象に残った場所ってどこ？

U1: イギリスかな

S2: そっかあ. **ところで, 好きな食べ物って何？**



2. 文脈を踏まえた発話

1.3 アプローチ

1. 言語理解の実装

➡ 対話システム構築フレームワークHRIDChat (PyChat改め)を用いる

- ドメイン依存の言語理解を構築可能
- 短い期間でも構築可能

2. 文脈を踏まえた発話

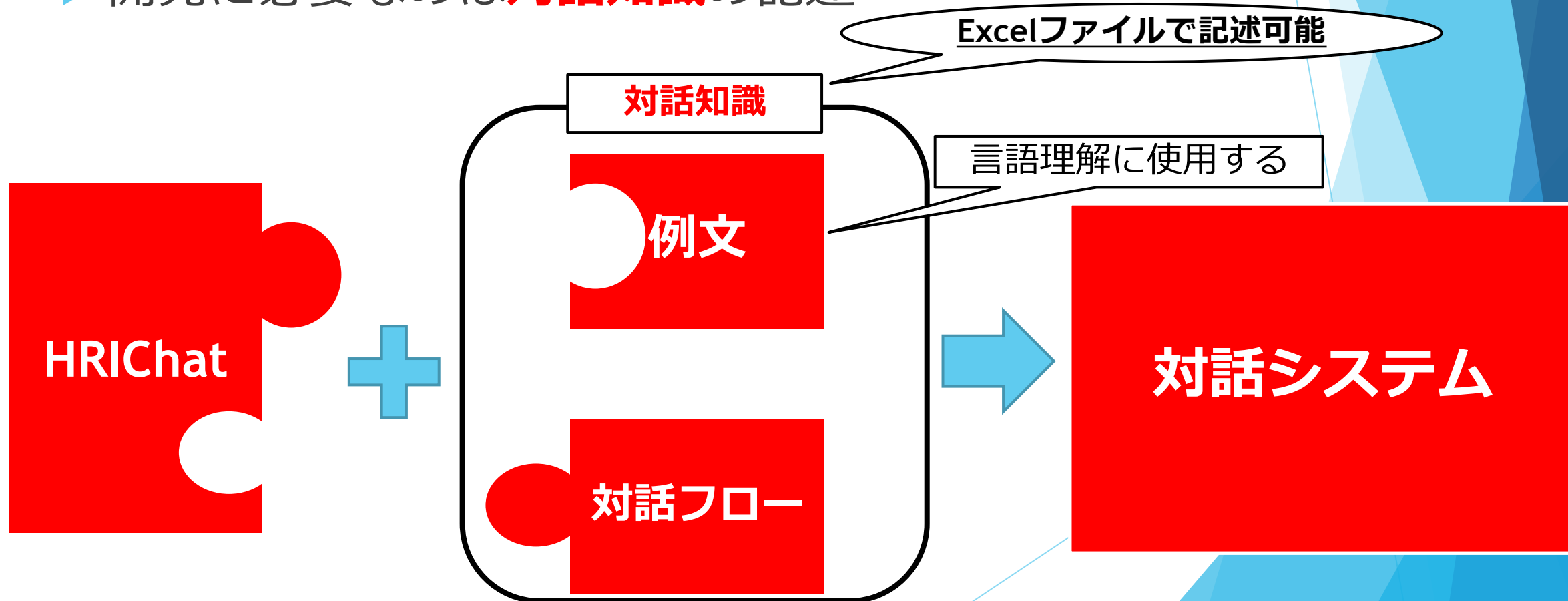
➡ 状態遷移ネットワーク型の対話管理

- 言語理解を利用した分岐も作成できる

- 1 概要
- 2 HRIChat
 - 2.1 言語理解
 - 2.2 例文の記述
 - 2.3 対話フローの記述例
- 3 対話の設計
 - 3.1 対話フローの構成
 - 3.2 システム発話表現の調整
- 4 予選の結果
- 5 まとめ

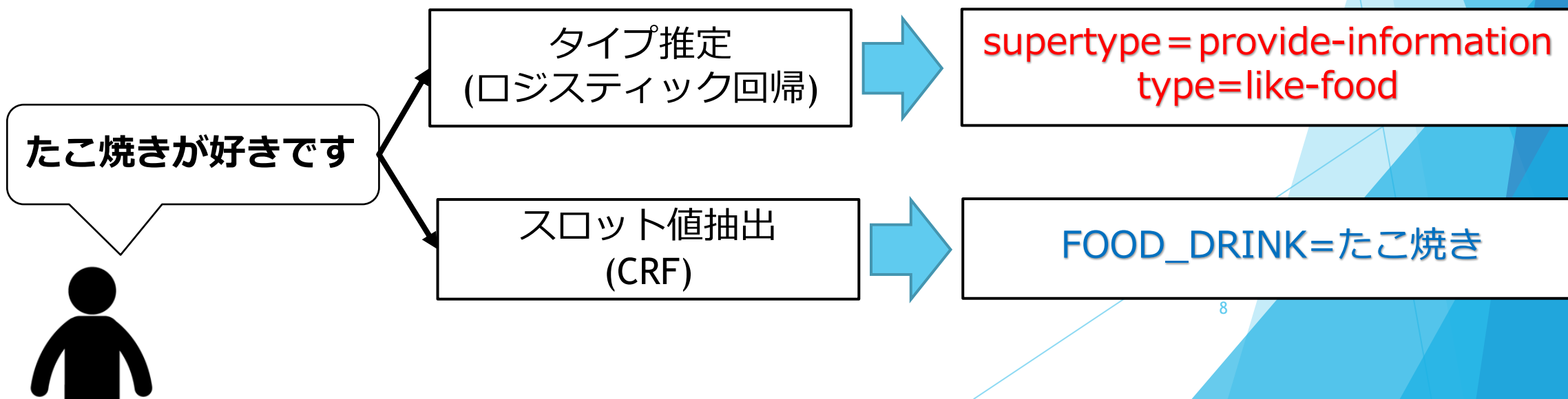
2 HRIChat

- ▶ HRIChatとは特定ドメインの対話システム構築フレームワーク
- ▶ 開発に必要なのは**対話知識**の記述



2.1 言語理解

- ▶ 文のタイプ推定
 - ▶ 粗いタイプ: supertype(19クラス)
 - ▶ 細かいタイプ: type(50クラス以上)
- ▶ スロット値抽出
 - ▶ 「PLACE」, 「FOOD_DRINK」, 「TIME_EVENT」の3クラスの名詞句を抽出



2.2 言語理解のための例文

- ▶ 例文の文中にスロットタグを入れて言語理解の学習に利用
 - 部分文字列を変更して作成し，例文の数量を増やす(432→20000)

supertype	type	発話例
provide-info	refer-place	<PLACE:福岡>がよかったよ
provide-info	refer-place	<PLACE:沖縄県>かな
acknowledge	yes	そうだよ
deny	no	違います

スロットタグ

2.3 対話フローの記述例

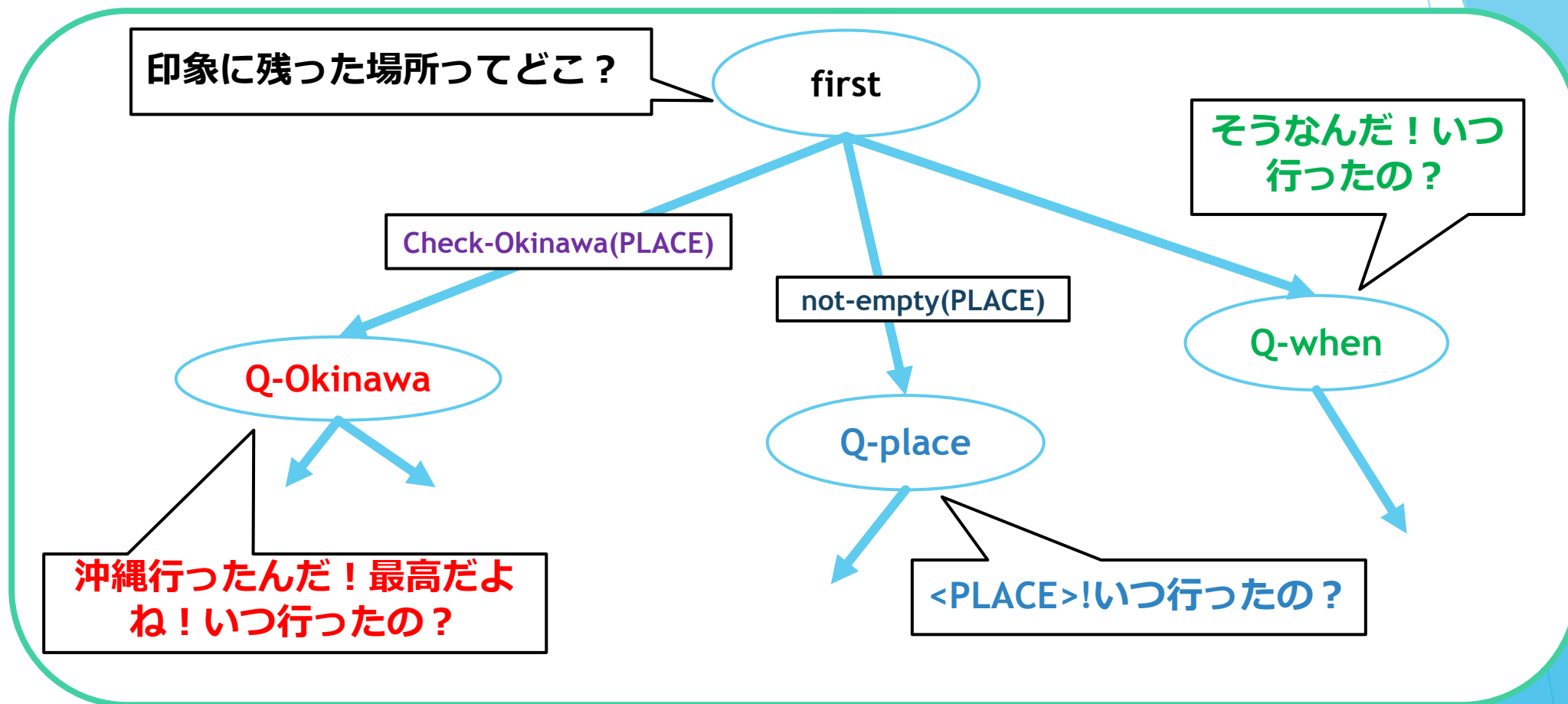
※システム発話は一部省略

状態	システム発話	条件	遷移先の状態
first	印象に残った場所ってどこ？	check-Okinawa(PLACE)	Q-Okinawa
first		not-empty(PLACE)	Q-place
first			Q-when
Q-Okinawa	沖縄行ったんだ！最高だね！いつ行ったの？		
Q-place	<PLACE>！いつ行ったの？		
Q-when	そうなんだ！いつ行ったの？		

スロット値を引数とした関数を分岐条件としている

- ▶ 全部で92行
- ▶ 関数はPythonのプログラムで定義

▶ 表を可視化したグラフ



➡ 対話フローの管理がしやすい

- 1 概要
- 2 HRIChat
 - 2.1 HRIChatの言語理解
 - 2.2 例文の記述例
 - 2.3 対話フローの記述例
- 3 対話の設計
 - 3.1 対話フローの構成
 - 3.2 システム発話表現の調整
- 4 予選の結果
- 5 まとめ

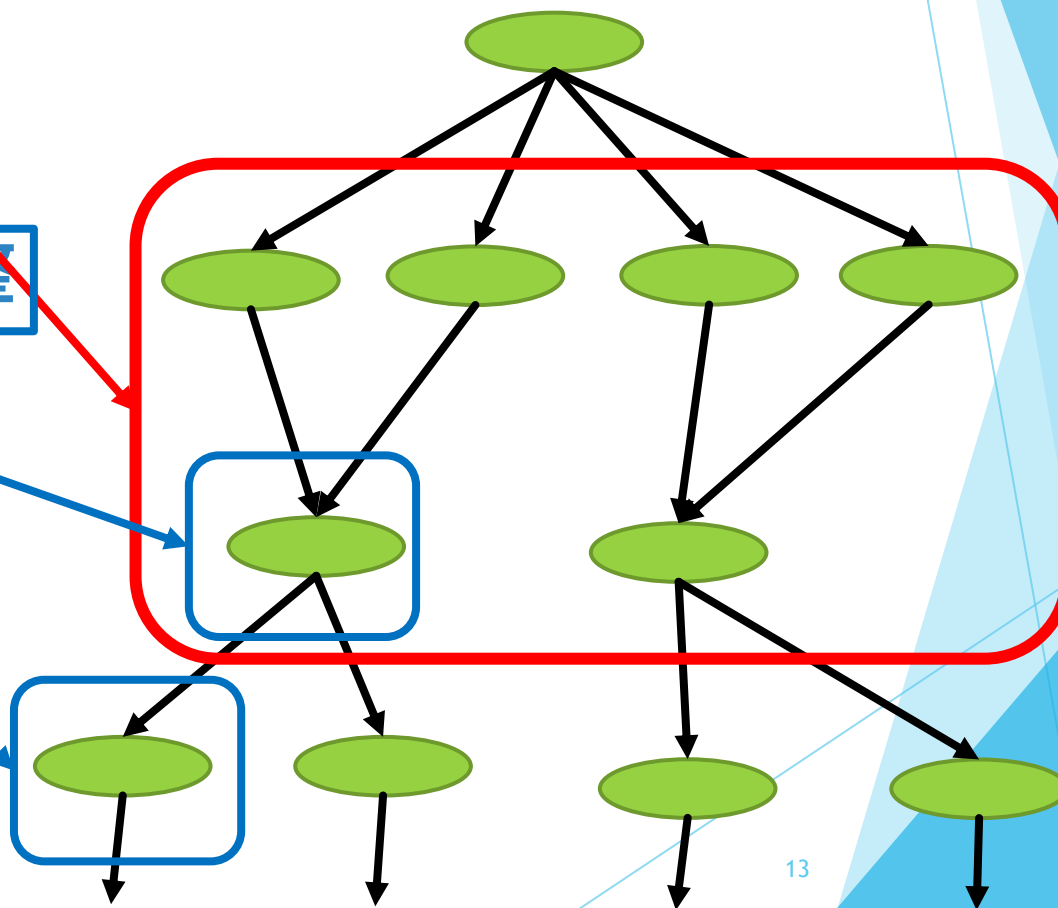
3 対話の設計

1. 構造としての設計

➤ 対話フローの構成

2. 各発話の設計

➤ システム発話表現の調整



3.1 対話フローの構成

▶ 対話を3つに分割(5発話×3)

➤ 管理がしやすい

▶ ユーザの印象に残った場所に合わせて自分語りの変化

S1: 印象に残った場所ってどこ？

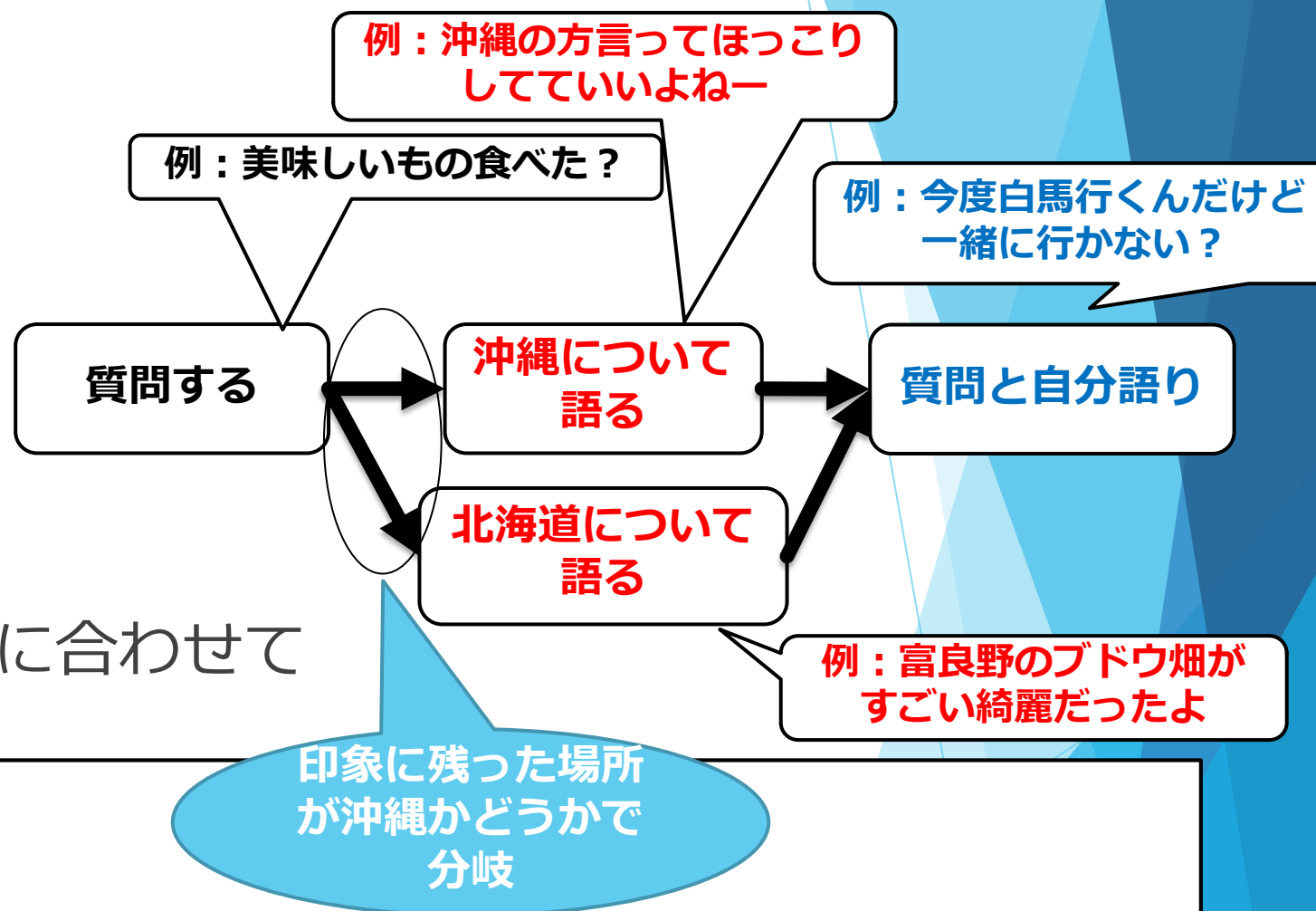
U1: 沖縄かな

S2: 沖縄! なんか美味しいもの食べた

U2: **締め**に食べたステーキが美味しかったなあ

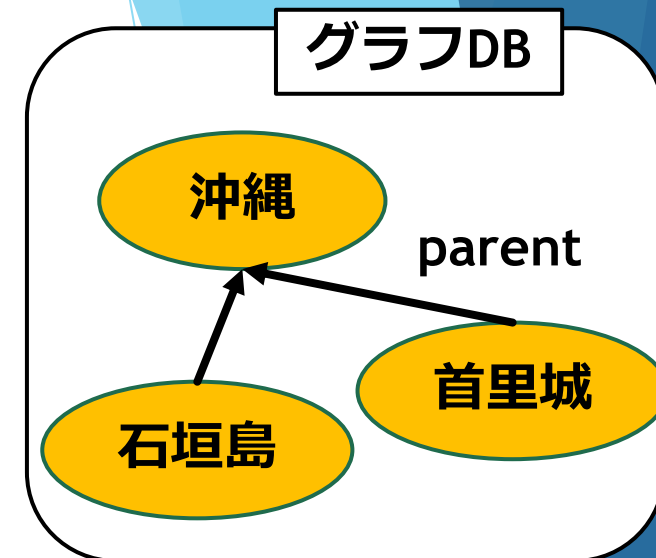
.....

S6: **そうそう、最近、北海道でのスイーツ巡りにハマってるんだ♪**



3.2 システム発話表現の調整

- ▶ ユーザ発話の情報を発話に組み込む
- ▶ 含まれる地名のグラフDB上の親をチェック
 - 「沖縄」か「北海道」



S1:ところで、これまで行ったところでは一番印象に残った場所ってどこ？

U1:石垣島かなー

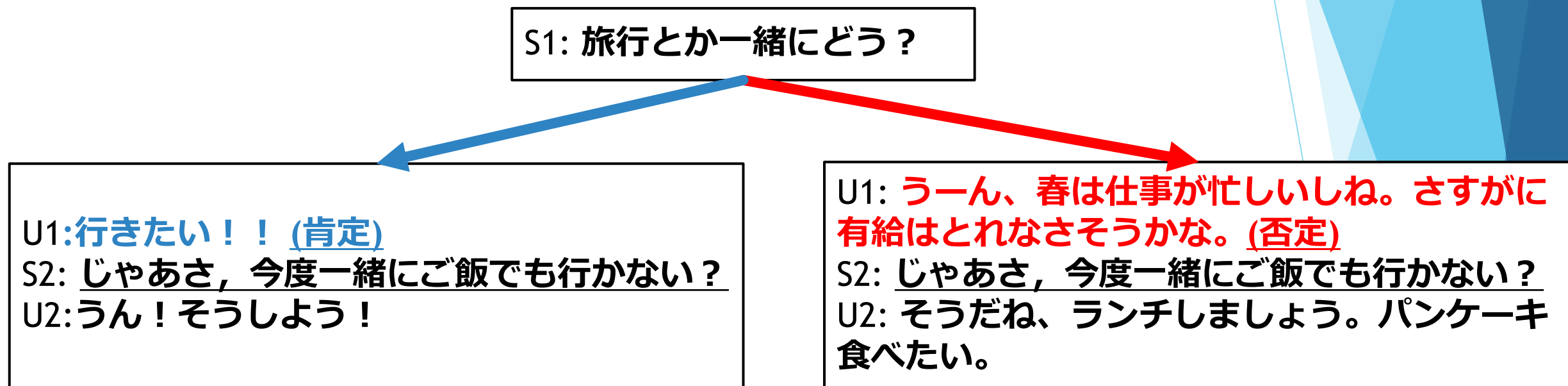
S2:石垣島！いつ行ったの？

S2:沖縄行ったんだ！海綺麗だし最高だよね！
いつ行ったの？

右の方は沖縄の地名だと理解している印象を与えている
⇒印象の向上

3.2 システム発話表現の調整

- ▶ 言語理解結果に依存しない応答



- 肯定でも否定でも成り立っている！

4 予選の結果

- ▶ 5段階で評価(最大5,最小1)して予選での評価点平均は4.1

予選でのコメント

- ▶ まずこちらの話を聞いてから自分の話をしてところが、なかなか聞き上手だと思いました。(評価点:4)
 - **対話を分割する構成がうまくいっている**
- ▶ こちらの投げかけた内容がそのままスルーされているので、もう少し拾ってくれると嬉しい。(評価点:2)
 - **ユーザに質問をされると弱い**

5 まとめ

- ▶ HRIChatを用いて実装する事で対話フローの管理が行いやすく、短期間でも作成できた
- ▶ 予選を経て実際の対話例を見る中で対話フローを分ける事で管理がしやすいだけでなくユーザの印象を上げている事がわかった

<今後の課題>

- ▶ 例文の拡張を行う事で言語理解の精度向上の余地あり