

FoodChatbot: 料理ドメインの雑談対話システム

FoodChatbot: A Chat Dialogue System in the Food Domain

中野 幹生^{1*} 駒谷 和範²
Mikio Nakano¹ and Kazunori Komatani²

¹ (株) ホンダ・リサーチ・インスティテュート・ジャパン

¹ Honda Research Institute Japan Co., Ltd.

² 大阪大学 産業科学研究所

² The Institute of Scientific and Industrial Research, Osaka University

Abstract: This paper presents FoodChatbot, which is a chat dialogue system in the food domain. It is built using a framework for developing closed-domain chatbots. It works on a Web interface, and uses a graph database containing domain knowledge and the system's personal information as the backend database. Its dialogue strategy is to mainly perform system-directed conversation and answer user questions when it detects them.

1 はじめに

FoodChatbot は、料理に関する雑談を行うシステムである。Web インタフェース上で動作し、テキストベースの対話を行う。対話毎に一つのトピックを決め、システムがユーザにそのトピックに関する質問をしたり、意見を言ったりすることで対話を進めていく。ユーザの質問に答えたり、ユーザ発話の中にあるキーワードに反応して発話を行うこともできる。

本稿では FoodChatbot の概要を述べる。FoodChatbot が用いている知識や処理の詳細および評価実験の結果は文献 [3] を参照して頂きたい。

2 システム構成

FoodChatbot は、(株) ホンダ・リサーチ・インスティテュート・ジャパンで開発した、Python で書かれた特定ドメインの雑談対話システム構築フレームワーク [3] に基づき、システム開発者が料理ドメインの対話知識を記述することにより構築している。図 1 に FoodChatbot の構成を示す。

Bootstrap と Tornado を用いて構築した Web インタフェース (図 2) 上でユーザが発話を入力すると、その発話の言語理解を行う。具体的には、辞書に mecab-ipadic-NEologd¹ を用いた MeCab[1] を用いて形態素解析を行ったあと、bag-of-words を特徴量としたロジスティック回帰を用いてユーザ発話のタイプ推定を、単語や品詞の連鎖などを特徴量とし IOB2 タグを出力する CRF を用いてスロット抽出をそれぞれ行う。ロジス

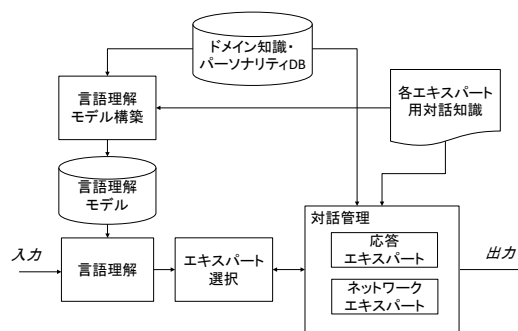


図 1: モジュール構成

ティック回帰の実装は scikit-learn を、CRF の実装は CRFSuite[4] を用いている。タイプ推定は粗いタイプ (supertype, 16 種類) と詳細なタイプ (type, 約 300 種類) の 2 つの推定を行う。スロットは、food-drink (料理・飲み物・食材など)、time-event (時間・季節・特定の料理を食べるイベントなど)、place (地名・レストランなど) の 3 種類である。ロジスティック回帰のモデルは対話知識中の例文を用いて訓練した。また、CRF のモデルは FoodChatbot の古いバージョンを用いて収集した対話のユーザ発話にアノテーションしたものをを用いて訓練した。

対話管理はマルチエキスパートモデル [2] に基づいており、ユーザ発話が入力されると、異なる対話戦略を持つ複数のエキスパートのうち一つが選ばれて対話管理と応答選択を行う。FoodChatbot では、ネットワークエキスパートと応答エキスパートの 2 つのエキスパートを用いている。

*連絡先: (株) ホンダ・リサーチ・インスティテュート・ジャパン
〒351-0188 埼玉県和光市本町 8-1
E-mail: nakano@jp.honda-ri.com

¹<https://github.com/neologd/mecab-ipadic-neologd>



図 2: Web インタフェース

ネットワークエキスパートは、システム開発者が記述した、システム発話の言語理解の結果に応じて状態が遷移する有限状態ネットワークを用いて対話管理を行う。これは、システムが主導権をとって、ユーザに質問したり自分の意見を言ったりしながら、ユーザの反応に合わせて次の発話を変えて行くような対話を行う。各状態にはシステム発話の候補が割り当てられている。また、状態遷移の条件として、supertype, type, および開発者が定義したブール関数を用いることができる。

応答エキスパートは、ユーザ発話に応じて応答する、一問一答型の対話を行う。応答エキスパートは、様々な種類の対話知識を用いている。例えば、言語理解結果の type や supertype とシステム発話のペアや、ユーザ発話例とシステム応答のペア、ユーザ発話中のスロット値とシステム発話のペアなどを用いる。これらの知識を用いて発話候補をリストし、その中から発話を選択する。

システム発話には、ネットワークエキスパートを起動するコマンドと起動後の初期状態が付加されているものがある。これにより、応答エキスパートが発話した後、ネットワークエキスパートが起動され、新たなシステム主導の対話を始めることもできる。

ユーザの入力に対してどちらのエキスパートを選択するかは、ユーザ発話の言語理解結果と各エキスパートの状態を用いた規則に基づいて決めている。

FoodChatbot は、バックエンドデータベースとして、言語理解結果のスロットに入りうる名詞句やそれらの間の関係を記述したグラフデータベースを用いている。このデータベースは辞書としても利用する。また、システムキャラクターの嗜好や経験なども記述されている。

データベース管理には Neo4j Community Edition²を用いている。このグラフデータベースにアクセスする関数を実装し、対話知識の様々なところで用いている。例えば、システム発話が選ばれるための条件を表すブール関数、ネットワークエキスパートの状態遷移の条件を表すブール関数、システム発話の中に埋め込んで使う文字列を返す関数などである。これらの関数の引数には、スロット値を保持する変数などを用いることができる。

3 対話戦略

システムのキャラクターは、アメリカから最近日本に来て日本の料理に興味を持っている 21 歳の女性 Sofia という設定にし、それに応じて料理の好き嫌いや食べたことのある料理などを決めてグラフデータベースに記述してある。このデータベースの内容を用いて発話を行うことで、システム発話の整合性を保つことができる。

各対話では、話題を 1 つ決め（セッションピックと呼ぶ）それに沿った対話を行うように対話知識を設計している。具体的には、セッションピックに沿った数ターンのネットワークをいくつか用意し、ネットワークエキスパートがそれらのネットワークを順次用いることで対話を進めていく。これにより、話題が頻繁に変わらない対話を行うことができる。

4 おわりに

料理ドメインの雑談対話システム FoodChatbot を紹介した。今後改善していくべき点として、言語理解の性能向上、および、ユーザ発話からのユーザ情報や興味 の推定結果を用いた対話の実現があげられる。

謝辞

FoodChatbot の開発に携わって頂いた皆様に感謝します。

参考文献

- [1] T. Kudo, K. Yamamoto, and Y. Matsumoto. Applying conditional random fields to Japanese morphological analysis. pp. 230–237, 2004.
- [2] M. Nakano, Y. Hasegawa, K. Funakoshi, J. Takeuchi, T. Torii, K. Nakadai, N. Kanda, K. Komatani, H. G. Okuno, and H. Tsujino. A multi-expert model for dialogue and behavior control of conversational robots and agents. *Knowledge-Based Systems*, 24(2):248–256, 2011.
- [3] M. Nakano and K. Komatani. A framework for building closed-domain chat dialogue systems. arXiv preprint arXiv:910.13826, 2019.
- [4] N. Okazaki. CRFsuite: a fast implementation of conditional random fields (CRFs), 2007. <http://www.chokkan.org/software/crfsuite/>.

²<https://neo4j.com/>