





表 1: 系列変換器による実際の応答生成例

発話主	応答
ユーザ:	ここ数年目がおかしくて、アレルギー症状のよう に感じています。
システム:	そうなんですか、、、目が疲れてるんですかね!
ユーザ:	どうしたら目の疲れが取れると思いますか?
システム:	ゲームのやり過ぎは注意だよ

表 2: 系列変換器の学習および実行時の設定

サブワード数 (入出力共有)	9,554
エンコーダ	2 層双方向 LSTM
デコーダ	2 層単方向 LSTM
単語埋込層/隠れ層次元数	512
最適化アルゴリズム	SGD
初期学習率	1.0
ミニバッチサイズ	128
学習エポック数	100
Dropout	0.3
ビームサイズ	5

subword-nmt[8]<sup>4</sup>を用いて各形態素をサブワードに分割した。

系列変換応答モジュールの機能としては、直前のユーザ発話のみならず、直前のシステム (自分自身) の応答文などの対話履歴 (文脈) を考慮してシステムの応答文を生成したい (例: Sordani ら [9])。そこで本モジュールでは、直前のユーザ発話に追加して、直前のシステム応答文を、文脈という位置付けで活用することを考えた。具体的な方法論としては、直前のシステム応答文と直前のユーザ発話を連結し、系列変換器の入力文として利用する方法を用いた [10]。発話履歴を効果的に使う方法論は他にもいろいろ考えられるが (例: Voita ら [11])、本対話システムでは、開発期間の都合を加味して最も簡単な方法論を採用した。今回採用した文を結合する方法論のメリットとして、コードそのもの (あるいは、系列変換器のモデル構成) は変更せずとも簡易的に発話履歴 (文脈) を考慮できる点が挙げられる。

## 4 知識モジュール

人手のテンプレートや定型文の作成にはコストが多くかかるが、システムがより自然にユーザと会話するためには大量のテンプレートや定型文を作成する必要がある。この問題に対して、ルールベース応答に統計的手法による応答を混合させることで、より対話の自然さを向上させる効果があることが報告されている [12]。本対話システムは、これらの統計的手法による応答を、

<sup>4</sup><https://github.com/rsennrich/subword-nmt>

表 3: 仮想人格モジュールでの実際の応答例

発話主	応答
ユーザ:	スポーツとかやってるの?
システム:	好きなスポーツは弓道です。「ずんだアロー」 で打った餅をずんだ餅に変えることができる んですよ!
ユーザ:	ずん子ちゃんにも夢とかあるの?
システム:	私の夢ですか?! 秋葉原にずんだカフェを作 りたいです。

系列変換器によって生成された応答に置き換えたものと見ることができる。以下の 4 種類のモジュールは、ユーザ発話内に特定の単語 (反応語) が含まれていた場合、その単語に反応して対応する応答を出力する。この節では各モジュールの詳細を述べる。

### 4.1 仮想人格モジュール: 『東北ずん子』

**本モジュールの目的** システムが人格に依存するような応答を返すことで、ユーザがより親しみを持って対話できるよう、システムに仮想的な人格を設定した。

**知識ベース作成工程** 本対話システムには、SSS 合同会社によるずんだ餅をモチーフにした少女のキャラクター『東北ずん子』<sup>5</sup>を起用した。誕生日などキャラクター固有の情報を 30 件用意したうえで、それらを矛盾無く応答に組み込むためのモジュールを加えた。

**応答生成法** 「誕生日」などキャラクター固有の情報を引き出し得る反応語がユーザ発話内に含まれていた場合、予め (反応語、定型文) の形で用意されたペアの定型文をそのまま応答として返すよう設計した。なお、本モジュールの反応語は他のモジュールに比べて少ないため、ユーザ発話に反応語が含まれている際には、本モジュールを必ず選択することにした。具体的な応答例を表 3 に示す。

### 4.2 雑学提示モジュール: Wikipedia から構築した知識ベースを用いた雑学提示

**本モジュールの目的** 対話システムにおいては、対話がユーザにとって心地よさを感じられるものであることと同時に、対話によって得られる知識が有用なものであることが望ましい。今回は『東北ずん子』のキャラクター設定から連想しやすい料理ドメインの知識に焦点を絞り、ユーザ発話に含まれる料理名や食材名に関連する雑学的な知識を提示するモジュールを導入した。その際、知識を手で網羅的に書くのではなく、Wikipedia から自動的に獲得した知識と、知識のタイ

<sup>5</sup><https://zunko.jp>

表 4: 雑学提示モジュールでの実際の応答例

発話主	応答
システム:	あなたの好きな料理はなんですか?
ユーザ:	ラーメンです。
システム:	ラーメンといえば、佐伯ラーメンって知ってますか?佐伯ラーメンは、大分県佐伯市のラーメンらしいですよ♪

天ぶらは、魚介類や野菜等の食材を小麦粉を主体とした衣で包み、油で揚げて調理する日本料理である。

↓ アンカーリンクに対する意味情報の自動付与

Subject	Predicate	Object
天ぶら	has part	魚介類
天ぶら	has part	野菜
天ぶら	has part	小麦粉
天ぶら	cuisine	日本料理
...	...	...

図 3: Wikipedia 本文内のアンカーリンク分類に基づく知識ベース自動構築

ブに対応した発話テンプレートを組み合わせることで、話題の網羅性、発話の自然さ、拡張性を担保することをめざした。

**知識ベース作成工程** 雑学モジュールの応答に用いる知識源として、日本語 Wikipedia のアブストラクトから自動抽出した知識三つ組 (主語 (*s*), 述語 (*p*), 目的語 (*o*)) の集合 (知識ベース) を用いた。

具体的には、エンティティ *s* と各アンカーリンク (図 3 中赤字下線) が指すエンティティ *o* 間の関係を予測する分類器を構築し、自動抽出に利用した。図 3 は抽出された知識の例を示している。例えば、「*s* = 天ぶらと *o* = 魚介類 は *p* = has part の関係である」という知識が抽出される。このプロセスによって、日本語 Wikipedia 全約 100 万記事から、およそ 180 万タプルの新たな知識を獲得した。構築した知識ベースは 900 種類に及ぶ広範な関係を含んでいるが、すべての関係に対して発話テンプレートを準備することは困難であるため、対話設計の戦略として料理関係の知識のみを抽出した。具体的には、Web サイト『コトバンク 和・洋・中・エスニック 世界の料理がわかる辞典』<sup>6</sup> に出現する料理名を主語または目的語エンティティに持つような 3,835 のタプルのみを、システムが知識として保有するよう設計した。

<sup>6</sup><https://kotobank.jp/dictionary/worldcook/>

表 5: クイズモジュールでの実際の応答例

発話主	応答
ユーザ:	私は東北地方に住んでいます
システム:	東北といえば、突然ですが、ここでクイズです!東北地方の中央部を南北に走る様子が背骨にもたとえられる、日本で最も長い山脈は何でしょう?
ユーザ:	奥羽山脈ですね
システム:	正解は~~、「奥羽山脈」です!おめでとうございます!

**応答生成法** Wikipedia から抽出された料理に関する知識ベースをもとに、様々な料理の雑学を提示する。具体的な手法は以下の通りである。

1. ユーザ発話中に、知識ベースタプル群のエンティティ (*s* または *o*) と完全一致するような文字列 (反応語) が含まれている場合、該当するタプルを全探索で抽出する。
2. 抽出されたタプル群から、適当なタプルを 1 つ選択する<sup>7</sup>。
3. 選ばれたタプルに対し、プロパティに応じた発話テンプレートを適用した文とアンカーリンクの出現した文を結合して、ユーザに対する次の応答として自然な形に整形する。

発話テンプレートは、テンプレートを作成しやすい特徴的なプロパティに対してのみ、プロパティ毎に適した文を人手で作成し、全 8 種類用意した。また、それらのプロパティに該当しない場合は、1) 頻度が少ないエンティティを話題として提案する場合、「*s* といえば *o* について知ってますか?」というようなテンプレートを、2)(1) にも該当しない場合は「*o* といえば *s* は美味しいですよね!」というような汎用的なテンプレートをを用いた。本モジュールにおける実際の応答例を表 4 に示す。

### 4.3 クイズモジュール: 早押しクイズ問題を利用したクイズ出題

**本モジュールの目的** 4.2 節の雑談提示モジュールでは、システム側が一方向的に雑学を提示するだけに留まり、ユーザとのインタラクションが減少する恐れがある。本モジュールは、ユーザとのインタラクションを増やすため、ユーザの発話に関連するクイズの出題をおこなう。

#### データ作成工程

クイズのデータセットとして、鈴木ら [13] が作成した解答可能性付き読解データセット<sup>8</sup>に含まれる早押し

<sup>7</sup>このとき、同じ話題が対話中で複数回出現してユーザに違和感を与えないように、それまでの対話中で用いたエンティティが含まれているタプルは選択肢の中から除外した。

<sup>8</sup><http://www.cl.ecei.tohoku.ac.jp/rcqa/>

クイズの問題，すなわち質問文  $q$  と正解  $a$  のペア ( $q, a$ ) を合計 12,591 問用いた．このデータセットでは，1 つの問題につき正解が 1 つしか付与されていないため，そのままでは表記ゆれを吸収した正誤判定をおこなうことができない．この問題を緩和するため，IPA 辞書を用いた MeCab により正解  $a$  の読み（ひらがなおよびカタカナ表記）を得て，それらの文字列も正解として判断するようにした．

また，ユーザの発話に関連したクイズを出題するにあたり，単純にユーザ発話と問題文中の任意の単語とのマッチングを取ると，一般的な単語にも反応してしまう．これに対しては，データセットの全問題文から，TF-IDF の値が一定値以上の単語のみを反応語として抽出し，これらの反応語がユーザ発話に含まれていた場合のみ，該当するクイズを本対話システムからユーザへと出題するようにした．これによって，問題文中の特徴的な単語のみで，ユーザの発話とのマッチングを取れるようにした．ただし，各問題文が必ず 1 つの反応語を含むとは限らないため，最終的に本対話システムでは反応語が含まれる 6,313 問の問題を使用した．

**応答生成法** 本モジュールの動作は，クイズの出題段階と，問題に対するユーザの解答の正誤判定をおこなう段階の 2 段階からなる．

出題段階では，ユーザの発話に含まれる反応語を検知し，対応する問題 ( $q, a$ ) を選択する．反応語に対して紐づく問題が複数ある場合は，その中から 1 つをランダムに選択する．選択された問題  $q$  を，発話の形に整形し出題する．この時点で一時的に正解  $a$  を保存しておく．正誤判定の段階では，解答としてのユーザ発話内に出題段階で保存しておいた正解  $a$  が存在するかどうかを完全一致で判定する．本モジュールでは確実に正しい回答のみを正解とし，正解との完全一致がない時は不正解であることを明示的に述べないことで，正誤判定の誤りによる対話の違和感が生じないようにした．クイズの出題とユーザの解答に対する応答の具体例を表 5 に示す．

#### 4.4 ダジャレモジュール：ダジャレの検知・応答

**本モジュールの目的** 非タスク指向型対話システムにおいてユーザを楽しませるためには，ユーモアを導入することは効果的であるとの報告がある [14]．本対話システムではユーモアの導入として，日本語の言葉遊びとして広く認知されているダジャレの検知およびダジャレを含む定型文での応答を実装した．

**知識ベース作成工程** 使用したダジャレの一覧は Web スクレイピングの手法を用いて『ダジャレ集 ダジャレ

表 6: ダジャレモジュールでの実際の応答例

	発話主	応答
(a)	ユーザ： システム：	自炊でオムライスを作るのが好きです じすいず (this is)自炊
(b)	ユーザ： システム：	アルミ缶の上にあるみかん 私がずんだ餅を食べている時にダジャレを言うなんて、、、反則です！

辞典』<sup>9</sup>より収集した．反応語とそれに対応する 1 つ以上のダジャレのリストからなる辞書形式で，984 語の反応語および 1,083 のダジャレを格納している．

**応答生成法** 反応語がユーザ発話に含まれる場合と，ダジャレそのものが発話中に含まれる場合の応答をそれぞれ用意した．具体的な応答例を表 6 に示す．

本モジュールがユーザ発話から反応語を発見した場合 (表 6(a))，反応語に対応するダジャレの候補，および用意された定型文から無作為に 1 つずつを選択し応答文を生成する．また，発話中にダジャレが含まれる場合 (表 6(b)) にも同様にモジュール選択制御の後，定型文による応答をおこなう．ここで反応語および，ダジャレの存在判定は完全一致によりおこない表記揺れには対応していない．反応語への意図しない合致による対話破綻を最小限に抑えるように実装した．

## 5 評価・分析

本コンペティションでは，実際に本対話システムと対話をおこなった 30 人のクラウドワーカーにより「どれくらいまた話したいと思うか」という指標で 1(とてもそう思う)~5(まったくそう思わない) 点の 5 段階評価がおこなわれた．ワーカーによる評価点の分布を表 7 に示す．得られた回答の点数を平均すると本対話システムのスコアは 2.27 点で全体 3 位の成績となった．

表 8 に，実際に対話をおこなったクラウドワーカーからのコメントを示す．各モジュールに対するコメントからは，概ね目的どおりにユーザを楽しませることができたことがうかがえる．しかし，改善点も指摘されているため，さらなる工夫が必要である．

今後の課題の一つとして，定型文応答の反応頻度を高めるために，表記揺れへのより柔軟な対応や反応語の拡充が求められる．また，本対話システムは，対話を主導しようとする傾向にあり，ユーザの質問に適切な応答を返せない事例が見受けられる．そのため，システムが自ら対話の主導権を握るべきか判断した上で適切な応答を選択できれば，会話をより成立させやすくなると思われる．

<sup>9</sup><https://dajareshoo.web.fc2.com/>

表 7: 30 人の評価者による評価点の分布

評価点	評価内容	人数	(割合)
1	とてもそう思う	5	(16.7%)
2	そう思う	16	(53.3%)
3	どちらとも言えない	5	(16.7%)
4	そう思わない	4	(13.3%)
5	まったくそう思わない	0	(0%)

表 8: クラウドワーカーによる本対話システムの評価

モジュール	コメント [評価点]
仮想人格	ずんだ餅や仙台の知識が得られて楽しくお話できました。ありがとうございました！ [評価点:1] 設定は面白かったのですが、ずんだ餅が会話に出てきたのは最初だけでしたので、もう少し会話に入れてきたら面白かったと思います。 [評価点:3]
雑学提示	ご飯の説明はさすがですが一部誤字ありました。ご飯について wikipedia 的な知識の他にも作り方のコツなどあれば会話をもっと面白いものになったかな、と思いました。 [評価点:2]
クイズ	突然の問題が多くて驚いたが、面白かったです。 [評価点:2]
全般	こちらの投げかけた質問に度々話の腰を折られましたが、まあ会話が成り立つ分だけまだ許容できます。 [評価点:2] こちらの会話にすんなりと合わせてくれた。不自然といえば不自然。 [評価点:2]

## 6 おわりに

本稿では、SLUD 対話システムライブコンペティションに参加した対話システム『Zunkobot』について述べた。本対話システムは、テンプレートおよび定型文を利用した応答(大規模知識を活用する3種類のモジュールと『東北ずん子』の人格に沿ったモジュールの応答)を知識モジュールとして用意し、ユーザ発話中にある特定の単語に反応し、反応したモジュールに応じた応答文を返す。また、知識モジュールでは対応できないユーザ発話に関しては、系列変換器による応答文を生成し対話をおこなった。結果として、多くのユーザに「また対話したい」と思わせることができた。今後の課題として、対話が破綻している箇所も少なからず見受けられたため、ワーカーからのコメントを踏まえ、さらなる性能改善に取り組む予定である。

## 謝辞

本対話システムの試運転に協力いただいた東北大学 乾・鈴木研究室の皆様へ感謝いたします。

## 参考文献

- [1] 東中竜一郎, 船越孝太郎, 稲葉通将, 角森唯子, 高橋哲朗, 赤間怜奈. 対話システムライブコンペティション. 人工知能学会 言語・音声理解と対話処理研究会第 84 回 (第 9 回対話システムシンポジウム), 2018.
- [2] Weizenbaum and Joseph. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, Vol. 9, No. 1, pp. 36–45, January 1966.
- [3] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.
- [4] 坪井祐太, 海野裕也, 鈴木潤. 深層学習による自然言語処理. 機械学習プロフェッショナルシリーズ. 講談社サイエンティフィック, 2017.
- [5] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP2015*, pp. 1412–1421, 2015.
- [6] Makoto Morishita, Jun Suzuki, and Masaaki Nagata. Ntt neural machine translation systems at wat 2017. In *Proc. WAT2017*, pp. 89–94, 2017.
- [7] 佐藤祥多, 乾健太郎. 因果関係に基づくデータサンプリングを利用した雑談応答学習. 言語処理学会第 24 回年次大会, 2018.
- [8] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proc. ACL2016*, pp. 1715–1725. Association for Computational Linguistics, 2016.
- [9] Alessandro Sordani, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proc. NAACL-HLT*, pp. 196–205, 2015.
- [10] Jörg Tiedemann and Yves Scherrer. Neural machine translation with extended context. In *Proc. the Third Workshop on Discourse in Machine Translation*, pp. 82–92, 2017.
- [11] Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. Context-aware neural machine translation learns anaphora resolution. In *Proc. ACL2018*, pp. 1264–1274, 2018.
- [12] 小林峻也, 萩原将文. ユーザの嗜好や人間関係を考慮する非タスク指向型対話システム. 人工知能学会論文誌, Vol. 31, No. 1, pp. 1–10, 2016.
- [13] 鈴木正敏, 松田耕史, 岡崎直観, 乾健太郎. 読解による解答可能性を付与した質問応答データセットの構築. 言語処理学会第 24 回年次大会, pp. 702–705, 2018.
- [14] 松井辰哉, 萩原将文. 発話極性を考慮したユーモアを有する非タスク指向型対話システム. 日本感性工学会論文誌 Vol.14 No.1(特集号), pp. 9–16, 2015.