

主題性にもとづく雑談対話システムの構築

吉越 卓見^{1,a)} 児玉 貴志^{1,b)} 新 隼人^{1,c)} 坂田 宜^{2,d)} 田中 リベカ^{1,†1,e)} 黒橋 穎夫^{1,f)}

概要：人間の対話は、文脈中に出現した語句のうち、その場において最も意識の向いた話題に沿って展開される。本研究では、各語句に向けられる意識の度合いを主題性と呼ぶ。既存の雑談対話システムは、文脈についてはそれを言語モデルへの入力として考慮しているが、話題は陽に意識することなく応答を生成する。そのため、話題に沿わない応答をしてしまうことが少なくない。本研究では、主題性の最も高い語句にもとづいて応答を生成することで雑談対話システムが話題に沿った自然な応答をすることを目的とする。この対話システムは、文脈中の主題性の最も大きい語句を推定する主題性推定モジュールと、与えられた語句にもとづいて応答を生成する応答生成モジュールからなる。主題性推定モジュールは、対話データにもとづいた自己教師あり学習により各語句の主題性を量化し、主題性の最も高い語句を選択する。応答生成モジュールは、与えられた語句が応答に含まれるように生成モデルを学習することで実現する。人手評価により、主題性を考慮した対話システムは、主題性を考慮しないときより自然な応答をすることが確認された。

キーワード：主題性、雑談対話システム、自己教師あり学習

Construction of Chat Dialog System Based on Topicality

TAKUMI YOSHIKOSHI^{1,a)} TAKASHI KODAMA^{1,b)} HAYATO ATARASHI^{1,c)} WATARU SAKATA^{2,d)}
RIBEKA TANAKA^{1,†1,e)} SADAQ KUROHASHI^{1,f)}

1. はじめに

人間の対話は、複数出現する語句の中からその場において最も意識の向いた話題に沿って展開していく。本研究では、語句を名詞または複合名詞として、各語句に向けられる意識の度合いを**主題性**と呼ぶ。また、文脈中の語句の中で最も主題性の高い語句を**主題語**と呼ぶ。

図1が主題性のイメージを示した図である。円の大きさ

が各語句の主題性の大きさを表す。システムの最初の発話において、主題語は「ゴルフ」である。図左のAさんの発話では、主題語が「ゴルフ」から「タイガーウッズ」へと変わっている。ここで、システムの次の応答としては、主題語である「タイガーウッズ」に関連した話をするのが自然な応答となる。一方で、図右のBさんの発話では、主題語が「ゴルフ」から「サッカー」へと変わっており、次のシステムの応答としては主題語である「サッカー」に関連した話をするのが話題に沿った応答となる。このように主題性を推定することはシステムが話題に沿った応答をするのに有用である。

近年、オープンドメインの雑談を目的とした対話システムの研究が盛んに行われている。特に、より柔軟な応答が可能なEnd-to-Endの雑談対話システムは著しく発展しており、相槌などの単調な応答だけでなく多様な応答の生成が可能となってきている[1][2]。現状の雑談対話システムは、文脈についてはそれを言語モデルへの入力とすること

¹ 京都大学
Kyoto University

² LINE 株式会社
LINE Corporation

†¹ 現在、お茶の水女子大学
Presently with Ochanomizu University

a) takumi.yoshiko@nlp.ist.i.kyoto-u.ac.jp

b) kodama@nlp.ist.i.kyoto-u.ac.jp

c) atarashi@nlp.ist.i.kyoto-u.ac.jp

d) wataru.sakata@linecorp.com

e) tanaka.ribeka@isocha.ac.jp

f) kuro@i.kyoto-u.ac.jp

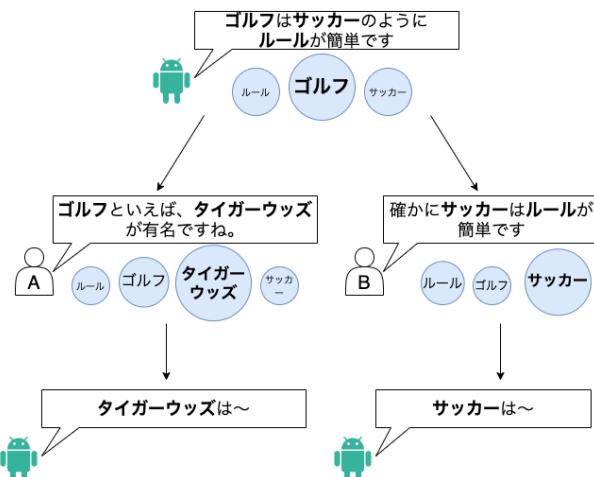


図 1: 主題性のイメージを表した図。円の大きさが発話時点での各語句の主題性の大きさを表す。左の対話では、「ゴルフ」から「タイガーウッズ」に話題が遷移していくが自然な一方、右の対話では、「ゴルフ」から「サッカー」の話題に遷移するほうが自然となる。

で考慮しているが、話題を陽に意識して応答を生成することはしない。したがって、多様な応答をすることにより、話題に沿わない応答となってしまうことがしばしば起こる。話題に沿わないような応答はユーザーを困惑させてしまう。

こうした背景から、本研究では主題性を踏まえ、話題に沿った応答を行う対話システムを提案する。提案システムは主題性推定モジュールと応答生成モジュールの2つからなる。主題性推定モジュールは、文脈中の各語句の主題性を定量化し、主題語を推定する。本研究では、文脈中の語句がその直後の応答中にも出現していれば、その語句の主題性は高いという仮説にもとづいて自己教師あり学習を行い、各語句の主題性を定量化するモデルを構築する。応答生成モジュールは、主題語が応答中に含まれるように生成モデルを学習する。推論時には、文脈中の語句から主題性推定モジュールによって推定された主題語をもとに応答を生成することで、話題に沿った自然な応答を行う。

結果として、この主題性を考慮した提案システムは、従来の文脈のみを考慮した対話システムよりも人手評価において高い評価を得た。また、主題語を含む応答と含まない応答では、主題語を含む応答の方がより高い評価を得た。したがって、主題性を考慮することは、対話システムが話題に沿った自然な応答を実現する上で有用であることが確認された。

2. 関連研究

近年、オープンドメインの雑談を目的とした対話システムの研究が盛んに行われている。特に、より柔軟な応答が可能な End-to-End の雑談対話システムは著しく発展し

ている。従来のシステムは、自然な応答を実現しようとするあまり、相槌などの中身のない応答を多く生成するという課題があった [3]。そうした課題の中で提案されたのが Meena [1] である。Meena は、パブリックドメインのソーシャルメディアでの会話から抽出された 400 億語の単語を使って学習された、生成ベースの対話システムであり、より具体的で質の高い応答を実現している。さらに、Meena を超えるモデルとして提案されたのが Blender Bot [2] である。Blender Bot は、Wizard of Wikipedia [4], ConvAI2 Dataset [5], Empathetic Dialogues [6] の3つで Fine-tuning することにより、知識、個性、共感の3つの能力を学習する。日本語においても、Blender Bot と同規模の対話システム [7] が構築されている。これらの対話システムは、中身のある応答の生成を可能としている。一方で、話題を意識することなく応答を生成するため、話題に沿わない応答となってしまうことがしばしば起こる。

検索ベースの対話システムにおいては、話題を意識した対話システムがいくつか提案されている。Target-Guided Open-Domain Conversation [8] では、オープンドメインの雑談から特定の話題に誘導する対話システムが提案されている。この対話システムは次に続きそうなキーワードの確率分布を求め、ターゲットに近づくようなキーワード候補から最も確率分布の高いキーワードを選び、そのキーワードにもとづいて応答を検索する。その手法をさらに拡張した手法としてキーワードの遷移をグラフ化した手法 [9] や、ConceptNet [10] を利用した手法 [11] が提案されている。検索ベースでの手法は、予め決められた応答候補の中からしか応答を選ぶことができないため、柔軟性に欠けるという欠点がある。本研究では、検索ベースで行われている話題を意識した手法を生成ベースの手法に適用し、話題を意識しながら応答を生成する対話システムを提案する。

3. 提案手法

本研究では、主題性にもとづいて応答を生成する雑談対話システムを提案する。この対話システムは主題性推定モジュールと応答生成モジュールの2つからなる(図 2)。主題性推定モジュールは文脈中の各語句の主題性を定量化することによって主題語を推定する。応答生成モジュールは、推定された主題語をもとに応答を生成することで話題に沿った自然な応答を実現する。

3.1 主題性推定モジュール

本研究では、対話データにもとづく自己教師あり学習により主題性推定モデルを構築する。対話の中で、ある文脈中の語句がその直後の応答にも出現していれば、その文脈においてその語句に向けられている意識の度合いは高いと考えられる。したがって、対話データから、文脈中の語句のうちその直後の応答に含まれる語句を正解として学習

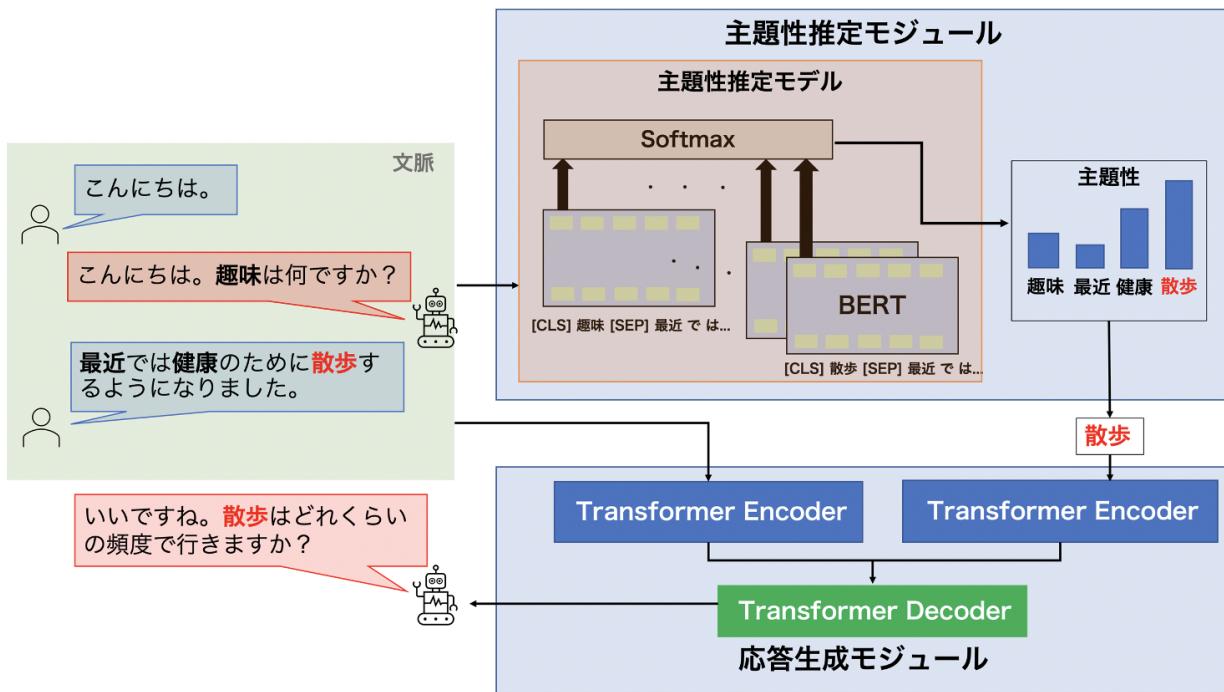


図 2: 提案システムの概要図。主題性推定モジュールでは、各語句の主題性を推定し主題語を選択する。図中では、主題語として「散歩」が選択されている。応答生成モジュールでは、文脈と選択した主題語をそれぞれエンコードしてから結合し、デコーダに入力することで応答を生成する。

データを構築する。この学習データを汎用言語モデルに学習させることにより、主題性推定モデルを構築する。

3.1.1 学習データの構築

まず、対話データから文脈と応答のペアを抽出する。ある 1 対話の発話数を k とすると 1 発話目から n 発話目 ($1 \leq n \leq k - 1$) までを文脈として $n + 1$ 発話目をその応答とする。次に、対話データの文脈中の名詞を文脈末尾から順に最大 8 個、候補語句として抽出する。同一文節内の連続する名詞は複合名詞としてまとめて抽出する。文脈中に重複して出現する語句は、最後の出現のみ抽出し、それ以外は考慮しない。候補の語句のうち文脈直後の応答中に出現した語句に正解ラベルを付与する。正解となる語句が複数ある場合は、文脈中の最も後ろに出現した語句を正解とする。^{*1}

対話において、主題となるような重要な語句が省略されることがよくある。そこで、正解、不正解のラベルを付与する際、応答中の省略された語句を省略解析器 [12] により補う。文脈中に出現する語句が 5 個未満の場合、もしくは文脈中の語句が応答中に出現しなかった場合は学習データから削除する。学習データの例を図 3 に示す。

3.1.2 主題性推定モデルの学習

汎用言語モデルとして BERT [13] を用いる。BERT は Self-attention 機構をベースとしたモデルであり、Pre-

^{*1} 正解の選び方として、応答中に含まれる語句を全て正解とする手法も考えられるが、これは予備実験で精度が高くないことを確認した。

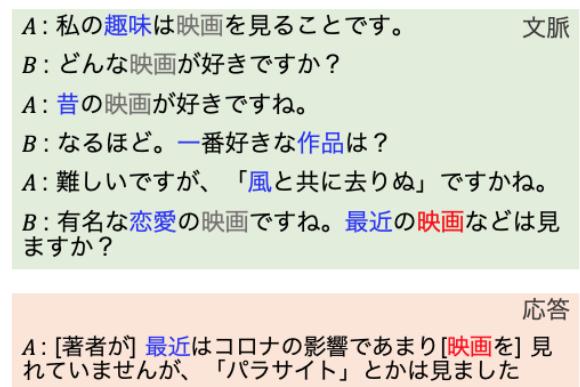


図 3: 主題性推定学習データセットの例。赤色の語句が正解、青色の語句が不正解、灰色の語句は重複のために候補から取り除かれた語句。

training と各タスク固有の Fine-tuning を施すことで、様々なタスクで State-of-the-art を達成している。本研究では、入力は先頭に [CLS] トークンと候補の語句 1 つを入力し、その後ろに 1 発話ごとに [SEP] トークンと対話履歴を新しいものから順に挿入する(図 2)。候補の語句それぞれに対して計算した BERT の出力を結合し、ソフトマックスにより確率分布として出力する。この確率分布と正解の確率分布とのクロスエントロピー誤差を最小化することにより主題性推定モデルを学習する。推論時は、主題性推定モデルの出力が最も大きい語句を主題語として出力する。

3.2 応答生成モジュール

応答生成モジュールは、文脈と主題語を入力として、その主題語にもとづいて応答を生成する。学習時は、主題語が応答中に含まれるように学習するため、応答中の語句を主題語とする。推論時は、主題性推定モジュールによって推定された、主題性が最も高い語句を主題語とする。

3.2.1 モデルアーキテクチャ

生成モデルのベースとしてTransformer [14] を用いる。主題語にもとづいて応答を生成するため、エンコーダでは、通常の Transformer エンコーダによって文脈と主題語をそれぞれエンコードし、それらを結合して出力する。デコーダでは、通常の Transformer デコーダに加え、Copy mechanism [15] と Rewarding [16] の機構を用意し、これらを組み合わせて使用する。

入力する文脈のトークン列を $x = (x_1, \dots, x_M)$ (M は文脈の系列長)、主題語のトークン列を $s = (s_1, \dots, s_N)$ (N は主題語の系列長)、生成モデルが出力する応答のトークン列を $y = (y_1, \dots, y_T)$ (T は応答の系列長) とする。 x と s は、単語埋め込み行列 $\mathbf{E} \in \mathbb{R}^{d \times |w|}$ (d は単語ベクトルの次元数、 $|w|$ は語彙数とする) によって、ベクトル列 $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_M]^T \in \mathbb{R}^{M \times d}$ と $\mathbf{s} = [\mathbf{s}_1, \dots, \mathbf{s}_N]^T \in \mathbb{R}^{N \times d}$ に変換される。この \mathbf{E} は学習を行うパラメータで、ランダムに初期化して用いる。文脈の中で主題語が重要であることをモデルに学習させるため、 \mathbf{x} 中の主題語のトークンの埋め込みにのみ学習可能なベクトル $\mathbf{t}_{topic} \in \mathbb{R}^{1 \times d}$ を足し合わせ、 \mathbf{x}' を得る。 \mathbf{x}' 、 \mathbf{s} はそれぞれエンコーダに入力され、隠れ状態 $\mathbf{h}_{\mathbf{x}s}^{src} \in \mathbb{R}^{M \times d}$ と $\mathbf{h}_s^{src} \in \mathbb{R}^{N \times d}$ に変換される。それらを結合することで $\mathbf{h}_{\mathbf{x}s}^{src} \in \mathbb{R}^{(M+N) \times d}$ を得る。

エンコーダ側からの隠れ状態 $\mathbf{h}_{\mathbf{x}s}^{src}$ をもとに、デコーダ側でトークン y_t ($1 \leq t \leq T$) を出力する際の、 \mathbf{w} の確率分布 $P_t(\mathbf{w})$ を算出する。以下、Transformer デコーダのみを使用する場合 (Transformer)、Copy mechanism を導入する場合 (Copy mechanism)、Rewarding を導入する場合 (Rewarding) の3つに分けて説明する。

Transformer $t - 1$ 番目までに出力された応答のトークン列 $y_{<t}$ として、 \mathbf{E} を用いてベクトル列 $\mathbf{y}_{<t}$ を得る。 $\mathbf{h}_{\mathbf{x}s}^{src}$ 、 $\mathbf{y}_{<t}$ を Transformer デコーダに入力することで出力 $\mathbf{h}_t^{tgt} = decoder(\mathbf{h}_{\mathbf{x}s}^{src}, \mathbf{y}_{<t}) \in \mathbb{R}^{t \times d}$ を得る。線形層を $\mathbf{W}_{out} \in \mathbb{R}^{d \times |w|}$ とすると Transformer デコーダによって予測される \mathbf{w} の確率分布 $P_t^{gen}(\mathbf{w})$ は次のように求まる。

$$P_t^{gen}(\mathbf{w}) = softmax(\mathbf{h}_t^{tgt} \mathbf{W}_{out}) \quad (1)$$

\mathbf{W}_{out} は学習によってパラメータを更新する。

最終的な y_t を生成する際の \mathbf{w} の確率分布 $P_t(\mathbf{w})$ は $P_t^{gen}(\mathbf{w})$ である。

Copy mechanism Copy mechanism を使用したのは、入力した文字列を出力にコピーできる機構を取り入れること

で主題語の出現確率を上昇させるためである。

入力した文脈及び主題語から得られた確率分布を $P_t^{copy}(\mathbf{w})$ と表すとき、Transformer デコーダから得られた \mathbf{w} の確率分布 $P_t^{gen}(\mathbf{w})$ と $P_t^{copy}(\mathbf{w})$ を重み $\alpha_t^{copy} \in [0, 1]$ で混ぜ合わせることで最終的な確率分布 $P_t(\mathbf{w})$ を求める。

$$P_t(\mathbf{w}) = (1 - \alpha_t^{copy}) P_t^{gen}(\mathbf{w}) + \alpha_t^{copy} P_t^{copy}(\mathbf{w}) \quad (2)$$

$P_t^{copy}(\mathbf{w})$ は、 \mathbf{h}_t^{tgt} に対する \mathbf{h}_{xs}^{src} の Attention の重みを求め、softmax によって確率分布として求まる。線形層を $\mathbf{W}_q \in \mathbb{R}^{d \times d}$ 、 $\mathbf{W}_k \in \mathbb{R}^{d \times d}$ 、 $\mathbf{W}_v \in \mathbb{R}^{d \times d}$ として、以下のよう求めること。

$$\mathbf{q}_t = \mathbf{h}_t^{tgt} \mathbf{W}_q^\top, \quad \mathbf{K} = \mathbf{h}_{xs}^{src} \mathbf{W}_k^\top \quad \mathbf{V} = \mathbf{h}_{xs}^{src} \mathbf{W}_v^\top \quad (3)$$

$$P_t^{copy}(\mathbf{w}) = softmax(\mathbf{q}_t \mathbf{K}) \quad (4)$$

式 (3)において、Attentionにおけるクエリが \mathbf{q}_t 、キーが \mathbf{K} 、バリューが \mathbf{V} にそれぞれ対応している。 α_t^{copy} は、 \mathbf{V} と $P_t^{copy}(\mathbf{w})$ から、 $\mathbf{W}_{copy} \in \mathbb{R}^{d \times 1}$ を線形層として、式 (5) のように求まる。

$$\alpha_t^{copy} = sigmoid\left(\mathbf{W}_{copy}^\top \sum P_t^{copy}(\mathbf{w}) \cdot \mathbf{V}\right) \quad (5)$$

各線形層 \mathbf{W}_q 、 \mathbf{W}_k 、 \mathbf{W}_{copy} 、 \mathbf{W}_v は学習によってパラメータを更新する。

Rewarding Rewarding は竹林ら [16] の手法にもとづく。Transformer デコーダもしくは、Copy mechanism によって予測された確率分布 $P_t^{dec}(\mathbf{w})$ (Transformer デコーダの場合は式 (1)、Copy mechanism の場合は式 (2)) に対して、トークン $w_i \in \mathbf{w}$ ($1 \leq i \leq |w|$) が主題語のトークン列 s に含まれる場合のみそのトークンの生成確率を上昇させる。 $P_t^{dec}(\mathbf{w})$ における w_i の生成確率を $p_t^{dec}(w_i)$ 、 $P_t(\mathbf{w})$ における w_i の生成確率を $p_t(w_i)$ とする。リワードスコア λ_{reward} をハイパーパラメータとすると、最終的な生成確率 $p_t(w_i)$ は式 (6) により表される。

$$p_t(w_i) = p_t^{dec}(w_i) \exp(\lambda_{reward} r_{w_i}) \quad (6)$$

$$where \quad r_{w_i} = \begin{cases} 1 & (w_i \in s) \\ 0 & (\text{otherwise}) \end{cases}$$

3.2.2 生成モデルの学習

$p_t(w)$ ($w \in \mathbf{w}$) を t 番目のトークン生成時の w の生成確率とする。生成モデルは正解の応答のトークン列 $y' = (y'_1, \dots, y'_{T'})$ (T' は正解応答の系列長) に対して以下の

表 1: 各コーパスの統計情報

| コーパス | 対話数 | 発話数 |
|------------------|---------|---------|
| 趣味雑談コーパス | 2,772 | 46,279 |
| Twitter 疑似対話コーパス | 約 2.5 億 | 約 6.7 億 |

損失関数を最小化するように学習される。

$$\mathcal{L} = - \sum_{t=1}^{|T'|} \log p_t(y_t') \quad (7)$$

3.2.3 デコード

デコードには Sample-and-rank [1] を使用する。Sample-and-rank は、温度 T のランダムサンプリングを使用して N 個の独立した応答候補を生成し、最終出力として各トークンの生成確率の積が最も高い応答候補を選択する手法である。

本研究では、主題語に含まれるトークン $s_n \in s$ ($1 \leq n \leq N$) の出現確率を上昇させるため、ある応答候補のトークン列を $y^{cand} = (y_1^{cand}, \dots, y_T^{cand})$ として、 y^{cand} が s_n を含む場合は、生成スコア s_{gen} に報酬を加える。 s_{gen} は主題語スコア λ_{topic} をハイパーパラメータとして式 (8) のように表される。

$$s_{gen} = \sum_{t=1}^{|T|} \log p_t(y_t^{cand}) + \lambda_{topic} r_{y^{cand}} \quad (8)$$

$$r_{y^{cand}} = \begin{cases} 1 & (s_n \in y^{cand}) \\ 0 & (\text{otherwise}) \end{cases}$$

最終的にはこの s_{gen} が最も大きい応答候補を応答として出力する。

また、その他の設定として、主題語が応答中に複数回出現するのを防ぐため、一度出現した主題語の生成確率は 0 に設定する。また、語句が重複して出現することを防ぐため、N-gram penalty [17] [18] を採用する。

4. コーパス

主題性推定モデルの学習データとして、趣味雑談コーパスを用いる。応答生成モジュールの学習データには、Pre-trainig に Twitter のリプライを繋げて構築された Twitter 疑似対話コーパスを用い、Fine-tuning には趣味雑談コーパスを用いる。各対話コーパスのサイズは表 1 の通りである。

4.1 趣味雑談コーパス

日本語話者同士の雑談対話を収集した利用可能な対話コーパスは非常に少ない。そこで、クラウドソーシングで 2 人のユーザの趣味に関する雑談を収集した趣味雑談コーパスを構築した。ユーザとシステム間の雑談対話において、通常システム側が自らの趣味等について話すことはな



図 4: 趣味雑談コーパス収集用の Web アプリ画面

表 2: 趣味雑談コーパスの対話例

| | |
|------|--------------------------------------|
| 話し手: | 私の趣味は映画です |
| 聞き手: | どんな映画が好きですか？ |
| 話し手: | 昔の映画が好きですね。 |
| 聞き手: | なるほど。一番好きな作品は？ |
| 話し手: | 難しいですが、「風と共に去りぬ」ですかね。 |
| 聞き手: | 有名な恋愛の映画ですね。最近の映画などは見ますか？ |
| 話し手: | 今年はコロナの影響であまり見れていませんが、「パラサイト」とかは見ました |
| | ... |

い。したがって、ユーザ側を「話し手」、システム側を「聞き手」として学習を行えるように立場を明確にして雑談を収集した。

Yahoo!クラウドソーシング^{*2}から対話収集用の Web アプリ [19] に誘導し、2 人のユーザをマッチングさせる。ユーザのうち 1 人を「聞き手」、1 人を「話し手」の役割に振り分ける。「話し手」は、自身の趣味について相手に伝えるように話してもらい、「聞き手」は「話し手」が話した趣味についてうまく話を広げるように話してもらう。ただし、「聞き手」は自身の行動に関する発言は行わないようにしてもらう。マッチング後は、Web アプリ上で「話し手」と「聞き手」交互に 1 発話ずつ、一人当たり 8-10 発話程度行なってもらう(図 4)。

結果として、2,772 対話、46,279 発話のサイズのコーパスを構築した。表 2 が実際の対話例である。

4.2 Twitter 疑似対話コーパス

2015 年の Twitter の日本語ツイートからツイートとそのリプライに該当するツイートをつなげることで作成された擬似的な対話コーパスである(表 3)。Twitter の元データで lang= ‘ja’ 以外のツイートを除去した。また、杉山ら [7] の手法にしたがい、文中でひらがな・カタカナの割合が 3 割以下のツイートは除去した。

^{*2} <https://crowdsourcing.yahoo.co.jp/>

表 3: Twitter 疑似対話コーパスの対話例

- A: 目の前立ったのに自動ドアが開きませんでした。
B: それ俺もよくある……
A: やっぱりありますよね。最近ないなあって安心してたらこの様です笑
B: あれなんなんだろうね?

5. 主題性推定モジュール

主題性推定モデルには、NICT BERT 日本語 Pre-trained モデル BPE あり^{*3} を使用し、趣味雑談コーパスから構築された学習データを用いて Fine-tuning することで、モデルを学習する。その結果をクラウドソーシングにより作成した Gold データをもとに評価を行う。

5.1 実験設定

Baseline として文脈中の一一番後ろの語句を主題語とする手法を考える。

学習データの構築手法として、学習データ構築の際に省略を補うかどうかの 2 つの設定で実験を行う。

Hobby w/o ZAR 省略を復元していない趣味雑談コーパスにより Fine-tuning を行う。

Hobby w/ ZAR 省略を復元した趣味雑談コーパスにより Fine-tuning を行う。

5.2 評価手法

主題性推定モデル評価のため、趣味雑談コーパスのテストデータからランダムに 57 対話 608 件を抽出し、クラウドソーシングで Gold データを作成した。ここでは、Yahoo! クラウドソーシングから独自で作成した Web アプリ(図 5)に誘導してタスクを行ってもらう。10人のワーカーに文脈を見せ、次に続きそうな話題 (= 語句) を選択してもらう(複数選択可)。5人以上のワーカーが選択した語句全てを Gold データとする。5人以上選択した名詞句が 1つもない場合は Gold データから取り除く。以上の手順により、525 件の Gold データを得た。

評価基準として P@1, R@K (K=1,3) を用いる。P@1 は、スコアがトップの Precision、R@K (K=1, 3) はトップ K での Recall である。

5.3 結果と考察

評価結果を表 4 に示す。まず、Baseline は単純な手法であるが、高い精度を出している。一般的に、日本語においては、主題が助詞「は」を伴って主語位置に生起することが多いため、文脈の中でより後ろに出現する語句ほど次の話題として選ばれやかたと考えられる。一方で、人間の対

^{*3} <https://alaginrc.nict.go.jp/nict-bert/>

Aさん・Bさん二人の対話文を見て次に続きそうな話題として最も適切だと思うものを選んでください。

例えば次の例では、「散歩」の話題が次に続きそうな可能性が最も高いため、「散歩」を選択してください。

A: 初めまして

B: 初めまして。よろしくお願ひします。

A: 何か **趣味** は、ありますか。

B: **休日** によく **散歩** をしています

図 5: クラウドソーシングのインストラクション画面。この画面では「散歩」が主題語として選択されている。

表 4: 主題性推定モデルの評価

| モデル | P@1 | R@1 | R@3 |
|---------------|--------------|--------------|--------------|
| Baseline | 0.561 | 0.440 | 0.777 |
| Hobby w/o ZAR | 0.594 | 0.457 | 0.796 |
| Hobby w/ ZAR | 0.606 | 0.484 | 0.815 |

話においてはそのような単純な場合だけでなく、主題が継続して続く場合や主題と焦点がはっきりしないような複雑な場合も多くある。そうした複雑な場合も含めて、Hobby w/o ZAR、Hobby w/ ZAR は Baseline を超える結果となり、提案手法がより高い精度で主題性を推定できていることを示している。^{*4}

Hobby w/o ZAR と Hobby w/ ZAR を比較すると省略の復元によって Precision、Recall ぞれぞれにおいて精度が上昇している。これは、文脈直後の主題となるような重要な語の省略を復元することにより主題性推定モデルの精度が上昇したと考えられる。

主題性推定モデルによって主題性を推定し、主題語を正しく推定できた例を表 5 に示す。Baseline では、一番後ろに出現した語句である「毎週」を選んでいるのに対し、H w/ ZAR モデルは Gold と同じ「映画」を選んでおり、正しい主題語を選択できている。

6. 応答生成モジュール

杉山ら [7] の手法を参考に、事前学習済み Transformer を用意し、それを各設定で Fine-tuning することで実験を行う。入力する文脈と主題語は、Juman++ [20] で形態素分割したのち、黒橋研究室で公開されている BERT 日本語 Pretrained モデル (LARGE WWM 版)^{*5} の BPE を適用し、トークン列に分割する。単語埋め込み行列 E は、事前学習時は初期化した行列を学習によって更新し、Fine-tuning も行列を学習してパラメータを更新する。

^{*4} モノローグデータを対話形式に変換して主題性推定モデルを学習する実験も行なったが、その結果はベースラインよりも精度が低かったため詳細には記載しない。

^{*5} https://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese

表 5: 主題語を正しく推定できた例

| 文脈 | |
|---------------|---------|
| モデル | 選択した主題語 |
| Baseline: | 毎週 |
| Hobby w/ ZAR: | 映画 |
| Gold: | 映画 |

表 6: エンコーダへの入力例 ([LST] が「聞き手」を表す特殊トークン、[SPK] が話し手を表す特殊トークン)

[LST] よろしくお願ひします。 [SEP] [SPK] こちらこそ よろしくお願ひします。 [SEP] [LST] あなたの趣味は何ですか？ [SEP] [SPK] 私の趣味は映画です。 [SEP]

6.1 事前学習

事前学習では、4.2 節の Twitter 疑似対話コーパスを使用する。エンコーダには文脈中の発話を [SEP] トークンでつなげて入力する。全トークン w の語彙数は 32,015 トークンである。モデルパラメータは、エンコーダ 2 層、デコーダ 24 層、各隠れ層 2560 次元、Attention head 数 32 とした。モデルサイズは 28 億パラメータであり、State-of-the-art の雑談対話システムである Blender Bot の 27 億と同程度の規模をもつモデルである。

6.2 Fine-tuning

Fine-tuning には、4.1 節の趣味雑談コーパスを使用する。エンコーダの入力形式は、文脈中の各発話を [SEP] トークンと話し手と聞き手を表す特殊トークンで区切ってエンコードを行う（表 6）。

入力する語句は、応答中の語句を候補として Hobby w/ ZAR の主題性推定モデルにより選択する。本研究では以下の 5 つの設定で実験を行う。

Baseline 文脈のみをエンコードする。デコーダは Transformer デコーダを使用する。

Concat 文脈と主題語をそれぞれエンコードし結合する。デコーダは Transformer デコーダを使用する。

Concat (Copy) 文脈と主題語をそれぞれエンコードし結合する。デコーダは Transformer デコーダに Copy mechanism を導入する。

Concat (Reward) 文脈と主題語をそれぞれエンコードし結合する。デコーダは Transformer デコーダに Rewarding を導入する。

Concat (Copy+Reward) 文脈と主題語をそれぞれエン

表 7: クラウドソーシングの選択肢と点数の対応表

| 選択肢 | 点数 |
|-----------|----|
| そう思う | 5 |
| ややそう思う | 4 |
| どちらとも言えない | 3 |
| あまりそう思わない | 2 |
| そう思わない | 1 |

コードし結合する。デコーダは Transformer デコーダに Copy mechanism と Rewarding を導入する。

6.3 デコード設定

Baseline を含む全てのモデルで Sample-and-rank を用いてデコードを行う。ランダムサンプリングには、top- k サンプリング、top- p サンプリングを適用し、 $k=40$ 、 $p=0.9$ とする。各パラメータは温度 $T=1.2$ 、応答候補数 $N=50$ 、主題語スコア $\lambda_{topic} = 100$ 、リワードスコア $\lambda_{reward} = 1.5$ とする。N-gram Penalty は Bigram のみを対象とする。

6.4 評価手法

自動評価では、趣味雑談コーパスのテストデータ 1,756 件に対して BLEU-1/2/3/4 と Perplexity を算出する。自動評価においては、主題語にもとづいて応答を生成できているかを評価するため、応答中の語句を入力とする。応答中の語句が複数存在する場合は Hobby w/ ZAR のモデルによって推定された主題性の最も高い語句を選択する。

人手評価では、趣味雑談コーパスのテストデータ 1,756 件から次の応答が「聞き手」側となるような文脈 100 件を抽出し、その文脈に対して生成した応答を Yahoo! クラウドソーシングにより評価する。文脈から主題性モジュール (Hobby w/ ZAR) によって選択された主題語を入力として応答を生成する。各設定ごとに文脈とそれに続く応答を提示し、1 つの応答に対して 5 人のクラウドワーカーに話題に沿った応答であるかを 5 段階で評価してもらう。5 段階をそれぞれ 1-5 点とした点数の平均値をその設定でのスコアとする（表 7）。

6.5 結果と考察

自動評価の結果を表 8 に示す。BLEU は、Concat (Copy+Reward) が最も精度が高かった。全体としては、Concat (Copy) が他の手法に比べやや低い精度となったが、その他の設定では手法による大きな違いは見られなかった。Perplexity は、Concat 以外の全てのモデルで Baseline を上回る結果となった。最も精度が良かったのは Concat (Copy+Reward) であった。また、Concat (Copy) と Concat (Reward) では、Concat (Reward) の方が精度が良いという結果となった。

人手評価の結果を表 9 に示す。全体のスコアとしては

表 8: 応答生成モデルの自動評価

| モデル | BLEU-1/2/3/4 | Perplexity |
|----------------------|---------------------|------------|
| Baseline | 10.0/6.37/3.73/1.94 | 72.6 |
| Concat | 10.0/6.38/3.70/1.91 | 72.6 |
| Concat (Copy) | 9.47/6.20/3.83/2.07 | 71.7 |
| Concat (Reward) | 9.93/6.37/3.77/1.71 | 66.3 |
| Concat (Copy+Reward) | 10.2/6.56/3.89/1.93 | 62.9 |

表 9: 応答生成結果の人手評価と応答中に主題語が出現した応答の割合

| モデル | スコア | 主題語の出現割合 |
|----------------------|-------------|-------------|
| Baseline | 2.11 | 0.12 |
| Concat | 2.30 | 0.45 |
| Concat (Copy) | 2.37 | 0.40 |
| Concat (Reward) | 2.63 | 0.64 |
| Concat (Copy+Reward) | 2.59 | 0.61 |

2~3 の値をとっており、クラウドワーカーの回答としては、「どちらとも言えない」から「ややそう思う」の間が平均的な回答であった。全体的に評価が低かった原因として、杉山ら [7] の手法と比べて、モデルのパラメータ数は同程度だが、事前学習に使用したコーパスの対話数が約 1/8、発話数が約 1/12、Fine-tuning に使用したコーパスの対話数が約 1/4、発話数が約 1/9 であり、コーパスのサイズがモデルの精度に影響していると考えられる。

各設定の結果を比較すると、主題語を入力して応答を生成する手法全てが Baseline を上回る結果となった。また、応答中に主題語が出現しているような応答の割合を調査した結果、主題語の出現割合が高いものほど、スコアも高くなる傾向が見られた（表 9）。提案手法の中では、Concat (Reward) が最もスコアが高く、主題語の出現割合も最も高かった。Concat (Reward) と Concat (Copy+Reward) の結果を見ると Copy mechanism を追加した場合の方がスコアが低くなっている。本実験では Copy mechanism は人手評価の観点では、それほど有効に働かなかった。

主題語を含む応答と含まない応答のスコアを比較するため、各設定ごとにそれぞれのスコアの平均値を表 10 に示す。主題語を含む応答のスコアは 2.5~3.0 の評価を得ているのに対して、主題語を含まない応答のスコアは 1.9~2.2 の評価しかない。この結果からユーザは主題語を含む応答の方がより話題に沿った応答であると思っていることがわかる。

表 11、表 12、表 13 に実際にシステムが主題語にもとづいて応答を生成した例を示す。表 11 は提案システムが主題語にもとづいて話題に沿った応答をしている例である。Baseline では、話題を意識して応答を生成していないため、話題を無視して「旅行」に関する話をてしまっている。一方で、提案システムは、文脈中の語句から「読書」を主題語として選択し、「読書」にもとづいて話題に沿った自

表 10: 主題語を含む応答と含まない応答の比較

| モデル | 主題語を含む | 主題語を含まない |
|----------------------|--------|----------|
| Baseline | 2.50 | 2.05 |
| Concat | 2.77 | 1.94 |
| Concat (Copy) | 2.93 | 2.03 |
| Concat (Reward) | 2.92 | 2.11 |
| Concat (Copy+Reward) | 2.85 | 2.13 |

表 11: 主題語を用いて話題に沿った応答を生成できた例 1

| 文脈 | |
|------------------|--------------------------------|
| 話し手: | こんにちは |
| 聞き手: | こんにちは。 |
| 話し手: | よろしくおねがいします。 |
| 聞き手: | 宜しくお願ひ致します。 |
| 話し手: | 私の趣味は読書です。 |
| モデル | 生成した応答 |
| Baseline: | 私の趣味は旅行とか行けないので、家族で旅行に行ってきました。 |
| Concat (Reward): | 読書ですか。どんなジャンルの本を読むのですか？？(読書) |

然な応答をしている。表 12 においても、提案システムは「反応」という主題語から話題に沿った自然な応答をしている。表 13 は、主題語にもとづいて応答を生成しているが不自然な応答となっている例である。「効果」という主題語を使えてはいるが、文脈ではサブリメントの「効果」を表しているのに対し、応答ではマスクの「効果」について話しており、文脈に沿った応答になっていない。このように、主題語にもとづいて応答をしているが、応答の内容そのものが文脈に沿っていない例も一定数見受けられた。

7. 結論

本研究では、主題性にもとづいて話題に沿った自然な応答を生成する対話システムを提案した。提案システムは、主題性推定モジュールと応答生成モジュールの 2 つからなる。主題性推定モジュールは、対話データにもとづく自己教師あり学習により主題性を推定し、主題性の最も高い語句を主題語として出力する。応答生成モジュールは主題語にもとづいて応答を生成することで話題に沿った応答を生成する。

提案システムは、従来の文脈のみを考慮した対話システムよりも人手評価において話題に沿った応答であることが示された。一方で、全体として人手評価のスコアは低く、ベースラインの精度の底上げが今後の課題である。

表 12: 主題語を用いて話題に沿った応答を生成できた例 2

| 文脈 | |
|------------------|---|
| 聞き手: | こんにちは |
| 話し手: | こんにちは (中略) |
| 話し手: | 長年気功法を習っています |
| 聞き手: | 名前しか知らないので教えて下さい |
| 話し手: | 3つの要素があって基本はゆっくり行う 腹式呼吸です |
| 聞き手: | そうですか |
| 話し手: | 2つめはストレッチの様な一連の動作で 全身を緩めます |
| 聞き手: | 身体に良さそうですね |
| 話し手: | その通りです。3つめは指導員と対気を行って気を入れ合います。 |
| 聞き手: | 何だか本格的ですね |
| 話し手: | 対気では様々な反応があって面白いです。 |
| モデル | 生成した応答 |
| Baseline: | はい。一度、二度目はストレッチの様な一連の動作で全身を緩めながら筋トレをしたりしてます |
| Concat (Reward): | へえー。そうなんですね。どんな反応ですか？(反応) |

参考文献

- [1] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemadé, Yifeng Lu, and Quoc V. Le. Towards a Human-like Open-Domain Chatbot, 2020.
- [2] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot, 2020.
- [3] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics.
- [4] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of Wikipedia: Knowledge-Powered Conversational agents, 2019.
- [5] Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. The Second Conversational Intelligence Challenge (ConvAI2), 2019.
- [6] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards Empathetic Open-domain Conversation Models: a New Benchmark and Dataset, 2019.
- [7] 杉山弘晃, 成松宏美, 水上雅博, 有本庸浩, 千葉祐弥, 目黒豊美, 中嶋秀治. Transformer encoder-decoder モデルによる趣味雑談システムの構築. In *SIG-SLUD No. 02*, p. pp.104–109, 2020.
- [8] Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. Target-Guided Open-Domain Conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5624–5634, Florence, Italy, July 2019. Association for Computational Linguistics.
- [9] Jinghui Qin, Zheng Ye, Jianheng Tang, and Xiaodan Liang. Dynamic Knowledge Routing Network for Target-Guided Open-Domain Conversation. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 05, pp. 8657–8664, Apr. 2020.
- [10] Robyn Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, 2018.
- [11] Peixiang Zhong, Yong Liu, Hao Wang, and Chunyan Miao. Keyword-Guided Neural Conversational Model. In *Association for the Advancement of Artificial Intelligence*, pp. 14568–14576, February 2021.
- [12] Nobuhiro Ueda, Daisuke Kawahara, and Sadao Kurohashi. BERT-based Cohesion Analysis of Japanese Texts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1323–1333, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019.
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob

表 13: 主題語を用いて応答を生成できたが話題に沿っていない例

| 文脈 | |
|------------------|--|
| 聞き手: | こんにちは |
| 話し手: | こんにちは (中略) |
| 話し手: | どくだみ茶いいですね。汗疹とか便秘、傷にも効きますからね。一時期はサプリメント 4 種類と漢方薬を 2 種類飲んでましたが、飲み過ぎだと思ったのでサプリメント 2 種類だけにしました。 |
| 聞き手: | すごいですね。最近、目にいいと言われ、何て言ったかな、飲みましたよ。 |
| 話し手: | ルテインですね。効きました？ |
| 聞き手: | そうです、そんな名前でした。効果は、わからないですね。 |
| 話し手: | ブルーベリーは体臭がブルーベリーのニオイになるぐらい摂ってましたけど、よくわからないのでやめました。効果がわからなくても楽しんでやってます。 |
| モデル | 生成した応答 |
| Baseline: | そうですね。その人が使っているとなかなか上からです。それ以降は、マスクを着用できない環境なので、なかなかできませんよね…。家ではジョギングとかは気にならないなと思っていました。 |
| Concat (Reward): | そうですね。やはりマスクされるときはマスクとかをしながらの運動にもよりますが、効果が大きいですよね(効果) |

- Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, 2017.
- [15] Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue, 2020.
- [16] Yuto Takebayashi, Chenhui Chu, Yuki Arase, and Masaaki Nagata. Word Rewarding Model Using a Bilin-gual Dictionary for Neural Machine Translations. *Journal of Natural Language Processing*, Vol. 26, No. 4, pp. 711–731, 2019.
- [17] Romain Paulus, Caiming Xiong, and Richard Socher. A Deep Reinforced Model for Abstractive Summarization, 2017.
- [18] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation, 2017.
- [19] 児玉貴志, Frederic Bergeron, 新隼人, 田中リベカ, 坂田亘, 黒橋禎夫. クラウドソーシングで利用可能な日本語対話収集基盤. 言語処理学会第 27 回年次大会発表論文集, pp. 859–864, 2021.
- [20] Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. Juman++: A morphological analysis toolkit for scriptio continua. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 54–59, Brussels, Belgium, nov 2018. Association for Computational Linguistics.