対話システムにおける人間との
コミュニケーションの円滑化に関する研究

提出先　　大阪大学大学院情報科学研究科

提出年月　2022 年 1 月

高 山　隼 矢

本研究に関連する研究業績

論文誌

1. 高山隼矢, 梶原智之, 荒瀬由紀. 対話における間接的応答と直接的応答からなる言い換えコーパスの構築と分析. 言語処理学会論文誌 自然言語処理, Vol.29, Number.1 (2021 年 3 月出版予定).
2. Junya Takayama, Eriko Nomoto, and Yuki Arase. Dialogue breakdown detection robust to variations in annotators and dialogue systems. Computer Speech & Language, Vol.54, pp.31-43, March 2019.

国際会議（査読あり）

1. Junya Takayama, Tomoyuki Kajiwara, and Yuki Arase. DIRECT: Direct and Indirect Responses in Conversational Text Corpus. in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: Findings, pp.1980-1989, November 2021.
2. Junya Takayama and Yuki Arase. Consistent Response Generation with Controlled Specificity. in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp.4418-4427, November 2020.
3. Junya Takayama and Yuki Arase. Relevant and Informative Response Generation using Pointwise Mutual Information. in Proceedings of the NLP for Conversational AI workshop, pp.133-138, August 2019.

国際会議（査読なし）

1. Junya Takayama, Eriko Nomoto, and Yuki Arase. Dialogue Breakdown Detection Considering Annotation Biases. in Proceedings of Dialog System Technology Challenge 6 Workshop, December 2017.

国内会議

1. 高山隼矢, 梶原智之, 荒瀬由紀. 間接的な応答と直接的な応答の対からなる対話コーパスの構築. 情報処理学会第 249 回自然言語処理研究会, Vol.10, pp.1-8, 2021 年 7 月. (若手奨励賞受賞)
2. 高山隼矢, 荒瀬由紀. Distant Supervision を用いた情報量を制御可能な雑談応答生成. 言語処理学会第 26 回年次大会, pp.323-326, 2020 年 3 月.
3. 高山隼矢, 荒瀬由紀. 自己相互情報量を用いた特徴語彙予測に基づく雑談応答生成. 言語処理学会第 25 回年次大会, pp.643-646, 2019 年 3 月.

本論文の梗概

言語能力を持つ人間にとって，自然言語を用いた対話は，円滑にコミュニケーションを行うための容易かつ効果的な手段であると言える．そのため，対話によって人間と情報システムのコミュニケーションを媒介する対話システム技術への注目度は高い．近年では，対話応答生成モデルと呼ばれる深層学習に基づく対話システム構築手法の登場により，大量の対話データを用いて容易に対話システムを構築できるようになった．

対話応答生成モデルは流暢な応答を生成できる利点を持つ一方で，抽象的な発話に対してその意図や要求を解釈できないという問題がある．また，しばしば「そうですね」や「わかりません」といったどのような発話に対しても成り立つ汎用かつ具体的でない応答を生成してしまう問題，さらに対話を破綻させるような応答を生成してしまうなどの問題がある．これらの問題は人間との円滑なコミュニケーションを妨げるものとなる．本論文では発話や応答の具体性に関するこれらの課題を解決するために次の 3 つの研究に取り組む：(1) 抽象的な発話の具体的な発話への事前言い換えに基づく対話応答生成，(2) 応答の具体性を制御可能な対話応答生成モデルの構築，(3) 対話破綻検出．

(1) では，人間はしばしば対話において自らの意図や要求を明示しない抽象的な発話を行うことに着目し，抽象的な発話の意図を理解し適切な応答を生成する手法の構築に取り組む．まず抽象的な発話と具体的な発話の対を含む 7 万件規模の対話コーパス DIRECT (Direct and Indirect REsponses in Conversational Text corpus) を構築する．また，対話応答生成モデルへの入力発話を事前により具体的な発話に言い換える手法を提案する．実験においては提案手法の適用によって通常の対話応答生成モデルよりも発話の意図に即した応答をより多く生成できるようになることを確認した．

(2) では，対話応答生成モデルがしばしば具体性の低い発話を生成してしまうという問題に対し，具体性の高い応答を生成可能な技術の実現を目的として研究を行う．人間は常に具体性の高い応答を行うのではなく必要に応じて応答の具体性を使

い分けていることに着目し，具体性を制御可能な対話応答生成手法の構築に取り組む．発話に対する応答の具体性を測る簡潔な尺度 MaxPMI を提案し，そのスコアを制御変数として訓練データに付与して訓練することで，具体性の制御を可能にする．評価実験では，提案手法が先行研究に比べて応答の具体性をより広い範囲で制御できることを，自動評価・人手評価によって検証・確認した．

(3) では，対話システムが生成した破綻応答を検知する対話破綻検出タスクに取り組む．対話破綻を感じる基準は人によって異なることに着目し，破綻ラベルのアノテーション傾向に基づいて訓練データのアノテータをクラスタリングし，様々なアノテーション傾向を再現する複数の検出モデルを構築してアンサンブルする手法を提案する．実験においては，他のアンサンブル手法や従来の対話破綻検出手法に比べて提案手法が高い検出性能を達成できることを確認した．また，(2) において提案した対話応答生成モデルの生成結果に対して対話破綻検出モデルを適用した場合の性能比較も行い，提案手法の優位性を示した．

本論文では，提案したこれらの手法によって対話応答生成モデルや対話破綻検出器の性能向上を達成し，人間との円滑なコミュニケーションが可能な対話システムの構築に寄与した．

目次

第 1 章　序論

1.1　研究背景

人間は自分の持っている情報や相手への要求・感情などの様々な意図を，自然言語を媒介して他人に伝えることができる．また，人間は相手から受け取った発話を基に相手の意図を推測し，それに答えるような応答を返すことができる．言語能力を持つ人間にとって，自然言語を用いた対話は，円滑にコミュニケーションを行うための容易かつ効果的な手段であると言える．そのため，対話によって人間と情報システムのコミュニケーションを媒介する対話システム技術への注目度は高い．現状でも Siri[1)], Google Assistant[2)], Alexa[3)] などのように，対話によって人間の意図を把握し，メールや情報検索など様々なタスクを行う対話システムが既に実用化されている．また，りんな[4)]やエアフレンド[5)]のように，人間と雑談を行うような対話システムも存在する．このように，対話システムには幅広い応用先が存在しており，今後も更なる技術革新が期待される分野である．

人間・コンピュータ間での対話の実現に向けた研究は古くから盛んに行われてきた．初期の対話システムである ELIZA [1] は，発話文中のフレーズとのパターンマッチに基づいて記述された複数の応答ルールに従って応答を出力するものである．このようなルールベースの対話システムは，その応答ルールを人手で大量に作成する必要があることから，システム構築に膨大なコストがかかるという欠点を持つ．ルールベース手法に代わる新たなアプローチとして，対話応答生成モデル [2] と呼ばれる深層学習を用いた応答生成手法が登場している．対話応答生成モデルは機械翻訳分野において提案された Sequence to Sequence (seq2seq) [3, 4, 5] モ

1） https://www.apple.com/jp/siri/
2） https://assistant.google.com/intl/_jp/
3） https://www.amazon.co.jp/meet-alexa/
4） https://www.rinna.jp/
5） https://airfriend.ai/ja/

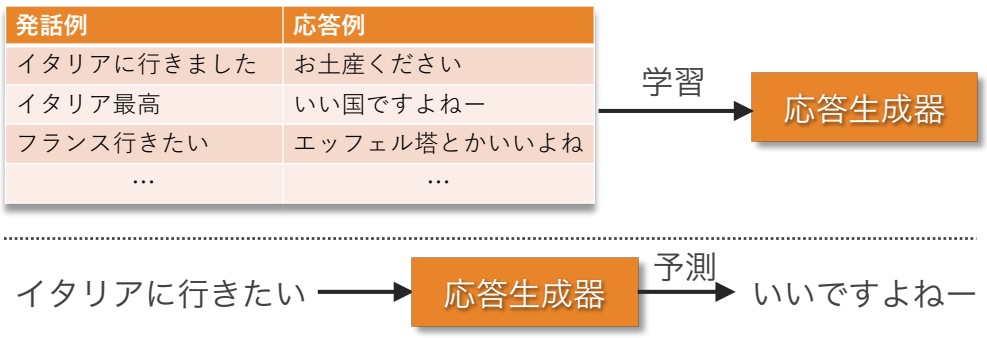

図 1.1　対話応答生成モデルの学習と予測の様子

デルを応答生成に適用したものであり，ニューラルネットワークを用いて発話に対する適切な応答を単語や文字単位で生成する．対話応答生成モデルは図 1.1 のように大規模な対話コーパスを基に発話と応答の対応関係を学習するため，コーパスの構築コストがかかる代わりに，人手によるルール作成コストは発生しないという特徴を持つ．コーパスの構築については，クラウドソーシングを用いて対話例を収集したり，Twitter などの SNS 上からユーザ同士の対話を収集するなど，様々な方法を取ることができるようになっている．こうした背景もあり，近年では対話応答生成モデルに関する研究が盛んに行われている．本研究においても，対話応答生成モデルを研究対象とする．

1.2　対話応答生成モデルが抱える現状の課題と本研究の位置づけ

対話応答生成モデルは多くの課題を抱えている [6]．本節では関連研究分野の俯瞰のために，対話応答生成モデルが抱える現状の課題を 5 つに分けて紹介する．

対話コーパスの構築に関する問題　対話応答生成モデルは大規模な対話コーパスを用いて訓練される．そのため用いた対話コーパスの性質はそのまま対話応答生成モデルにも引き継がれる．近年では SNS の普及やクラウドソーシング等のプラットフォームの登場によって容易に大規模な対話コーパスを獲得できるようになった．しかし，医療や科学技術など専門的な内容に特化した対話コーパスや，特定の目的に特化した対話コーパスを得るのは依然として難しい．また，SNS 等から抽出した大規模なコーパスを用いる場合，対話として成り立っていないような品質の低いデータが多く混ざっている場合がある．そのため，

構築したい対話システムの要件を満たすような性質を持ち，かつ品質が高い対話データを低コストに獲得する技術の実現は重要な課題である．

発話意図の正確な理解に関する問題　対話応答生成モデルはしばしば発話に込められた意図や要求に応えない見当違いな応答を生成することがある．これは，ユーザが入力発話に込めた意図をモデルが正確に理解できていないことに起因すると考えられる．モデルがユーザの発話意図を理解できなければ，ユーザは同じ意図の発話をモデルが理解できるまで様々な表現に言い換えて入力する必要が生じる．ユーザとの円滑なコミュニケーションのためには，発話意図をより正確に推定できるような技術の実現が不可欠である．

具体的な情報を過不足なく含む応答生成に関する問題　対話応答生成モデルは流暢な応答を生成できるが，一方で「そうですね」や「わかりません」などのようにどのような発話に対して応答として成り立つような，具体的でない汎用な応答を頻繁に生成してしまう．例え対話を成立させるような応答を生成できていても，それがユーザが求めるだけの情報を含んだ応答でなければユーザと円滑にコミュニケーションを取ることができない．また，ユーザが求める以上の情報を過剰に含んだ応答もまたコミュニケーションの妨げとなりうる．具体的な情報を過不足なく含んだ応答の生成はユーザとの円滑なコミュニケーションのために必要な能力である．

対話を破綻させるような応答の検出に関する問題　対話応答生成モデルは対話を破綻させてしまうような破綻応答を生成することがある．破綻応答には，自然言語文として成り立っていないような応答や，構文的には正しいが文意が汲み取れないような応答や，発話と全く関係のない応答など様々な種類のものがある．そのため，対話応答生成モデルの実用化にあたってはそのような破綻応答をユーザに出力する前に検出しフィルタリングするような機構の実現が不可欠である．

応答のスタイルに関する問題　ある一連の対話の中では，人間は通常終始一貫したスタイル（その人固有の口調・口癖や，対話相手との関係性に依存した言葉遣いなど）で発話を行う．対話システムもそのシステムのキャラクタに基づいて一貫したスタイルで応答できなければ，ユーザに不自然さを感じさせてしまう可能性がある．しかし，大規模な対話コーパスには様々なスタイルの応答が含まれることから，対話応答生成モデルの応答スタイルは通常一貫しない．ま

た，一貫した応答スタイルを持つ対話コーパスを構築するためには高いコストが必要となる．一貫したスタイルでの対話応答生成を効率よく実現する技術が不可欠である．

本研究においては，人間とより円滑にコミュニケーションを行う対話応答生成モデルの構築を目的とする．円滑なコミュニケーションを行うためには，次の 3 つの要件を共に満たす必要があると考えられる．(1) ユーザは意図を明示しない**抽象的な発話**を行うことがあるため，モデルはそのような発話の意図を汲み取る必要がある．(2) 一方で，システムはユーザが必要な情報を過不足なく含む**具体的な応答**をする必要がある．(3) モデルが生成した応答をユーザに対して出力する前に，その応答が**対話を破綻させるような応答**になっていないことを確認する必要がある．

関連研究分野の中で，(1) は上述の「発話意図の正確な理解に関する問題」，(2) は「具体的な情報を過不足なく含む応答生成に関する問題」，(3) は「対話を破綻させるような応答の検出に関する問題」に属する課題として位置づけられる．対話応答生成モデルに関する残る 2 つの課題については本研究では取り扱わないが，いずれも重要な研究課題である．「対話コーパスの構築に関する問題」については文献 [7, 8, 9] など，「応答のスタイルに関する問題」については文献 [10, 11, 12] など，それぞれ様々な取り組みがなされている．

1.3 本論文の構成と研究内容

本章以降の構成を説明する．まず，第 2 章において，本論文の研究対象である対話応答生成モデルと，その評価方法について詳説する．第 3, 4, 5 章では，先に挙げた (1), (2), (3) の研究目的に沿って行った研究内容について，それぞれ詳説を行う．以下にその概要を示す．

第 3 章：抽象的な発話の具体的な発話への事前言い換えに基づく対話応答生成

対話において人間はしばしば自身の要求や意図について具体的に言及せず，抽象的に表現することがある．人間は抽象的な発話に含まれる言外の意図を推測し，それに基づいて適切な応答を返すことができる．例えば，レストラン等の予約サービスにおいてユーザがオペレータに対して「あまり予算がないのですが」と言った場合，オペレータはその発話には「もっと安い店を提示してくだ

さい」という言外の意図が含まれていると解釈できる．人間と円滑なコミュニケーションを行う対話システムの実現のためには，抽象的な発話の言外に暗示された意図（具体的な発話）を推定する技術が不可欠である．第 3 章では，文献 [13, 14] で公表した研究成果に基づき，対話応答生成において発話を具体的な発話へと事前に言い換える手法を提案し，実験的に評価した．本研究において重要となる抽象的な発話を含むコーパスはこれまで存在しなかった．そこで本研究では，抽象的な発話に関する研究分野の発展に寄与するために，抽象的な発話と具体的な発話の対を含む対話コーパス DIRECT (Direct and Indirect REsponses in Conversational Texts Corpus) を構築した．

実験では，構築したコーパスを用いて抽象的な発話を具体的な発話に言い換えるモデルを構築し，その言い換え性能を検証した．大規模なコーパスを用いて事前学習を行った文エンコーダを用いることで，正確に言い換えを推定できることが確認された．最後に，言い換えモデルを対話応答生成モデルの入力発話に適用し，事前言い換えに基づく対話応答生成モデルの性能の検証を行った．事前言い換えを行うモデルは全ての評価尺度において通常の対話応答生成モデルよりも高い性能を達成した．

第 4 章：応答の具体性を制御可能な対話応答生成モデルの構築

対話応答生成モデルは「そうですね」や「わかりません」などのようにどのような発話に対しても成り立つような応答（dull response）を頻繁に生成してしまうという問題がある．これを防ぐために応答の具体性の向上に取り組んだ研究は多く存在する．しかし，人間は実際の対話において具体性の低い応答と高い応答を必要に応じて使い分けている．そのため，人間との円滑なコミュニケーションのためには，常に具体性の高い応答を生成するのではなく，対話応答生成モデルも必要に応じて応答の具体性を制御できることが望ましい．そこで第 4 章では，文献 [15, 16] で公表した研究成果に基づき，具体性を制御可能な対話応答生成モデルの構築を目指して研究を行った．具体的には，発話とよく共起する単語が含まれる応答は具体性が高いという仮説に基づいて，応答の具体性を測る尺度 MaxPMI を設計し，これを具体性の制御のための変数として用いた．また，与えられた制御変数に応じて発話との共起が強い単語の生成確率を増幅するような機構をモデルに追加した．

実験では日本語と英語それぞれの対話データを用いて，複数の自動評価尺度

と人手評価指標によって評価を行った．自動評価と人手評価の両方において，提案手法が比較手法に比べてより具体的な応答を生成可能であることを確認した．また，具体性の制御性という観点においても，提案手法の方がより幅広く具体性を制御できることを示した．エラー分析においては，提案手法や既存手法の両方において，通常の対話応答生成モデルよりも頻繁に対話を破綻させるような応答を生成してしまうことがわかった．これに対処するために，生成した応答文中の単語頻度情報を用いて破綻応答をフィルタリングするヒューリスティックな手法も提案し，一定の効果が得られることを示した．

第 5 章：アノテータの多様性に頑健な対話破綻検出

対話応答生成モデルはしばしば対話を破綻させるような応答を生成する．特に，第 4 章で構築したような応答の具体性を制御するような手法では破綻応答をより頻繁に生成してしまう．そのため，対話応答生成モデルの実用化にあたってはそのような破綻応答をユーザに出力する前に事前に検知しフィルタリングするような機構の実現が不可欠であると考えられる．そこで第 5 章では，文献 [17, 18] で公表した研究成果に基づき，ある与えられた人間–システム間の対話において，その対話を破綻させるようなシステム応答を検知する対話破綻検出タスクに取り組んだ．対話破綻を感じる基準は人によって異なるため，対話破綻検出のデータはアノテータによってアノテーションの傾向が大きく異なる．本研究では，アノテーション傾向に基づいてアノテータをクラスタリングし，各クラスタの傾向を反映した破綻検出器を構築し，それらをアンサンブルすることで破綻検出性能の向上を目指した．

事前実験においては，他の代表的なアンサンブル手法と提案手法の比較を行い，アノテーション傾向を考慮する提案手法が性能向上に有効であることを示した．その後，対話破綻検出の共同タスクである対話破綻検出チャレンジ 3 の評価データを用いて，提案手法の対話破綻検出性能を検証した．実験では，提案手法が従来の破綻検出手法よりも高い破綻検出性能を持つことを確認した．最後に，第 4 章において構築した対話応答生成モデルが生成した応答に対して提案手法を含めた複数の対話破綻検出モデルを適用したところ，提案手法が最も高い検出性能を持つことが確認できた．

最後に，第 6 章において以上の研究成果をまとめ，今後の展望を議論する．

第 2 章　事前知識

本章では事前準備として，本研究の研究対象である対話応答生成モデルに関する詳説を行う．また，対話応答生成モデルの評価方法についても解説する．

2.1　対話応答生成モデル

本研究における研究対象である対話応答生成モデルについて，その詳細を説明する．昨今の深層学習技術の発展に伴い，ニューラルネットワークを用いて自然言語文を生成する試みがなされるようになった．特に，再帰的ニューラルネットワーク（Recurrent Neural Networks; RNN）を用いて一単語ずつ順に文を生成する RNN 言語モデル [19] の登場により，流暢な自然言語文を生成することが可能になった．RNN 言語モデルは，ある単語列 $y_1, y_2, ..., y_{t-1}$ が与えられたとき，その次の単語 y_t を予測するモデルであり，文の生成確率 $P(\mathbf{Y})$ は以下で定式化される．

$$P(\mathbf{Y}) = \prod_{t=1}^{T+1} P(y_t|y_{t-1}, ..., y_0)$$

Sequence to Sequence (seq2seq) [3, 4] とは，機械翻訳分野において提案された，RNN 言語モデルを文から文への変換モデルへと拡張したモデルである．機械翻訳分野において，seq2seq は従来の統計的機械翻訳手法 [20] に比べて翻訳性能の著しい向上を達成している．seq2seq は図 2.1 のように，エンコーダとデコーダと呼ばれる二つの RNN によって構成される．エンコーダは変換前の単語列を入力として受け取り，その内容を表すような実数値ベクトルを出力する．デコーダはエンコーダの出力を初期状態として受け取り，変換後の単語列を一単語ずつ順に生成するものである．なお，各単語には事前に整数値の ID が割り振られており，その単語の ID に該当する要素のみが 1 で残りの要素が 0 となる離散値ベクトル（one-hot ベクトルと呼ばれる）を用いることでニューラルネットへの入力が可能になる．seq2seq は RNN 言語モデルのデコーダにエンコーダの出力を条件として加えたよ

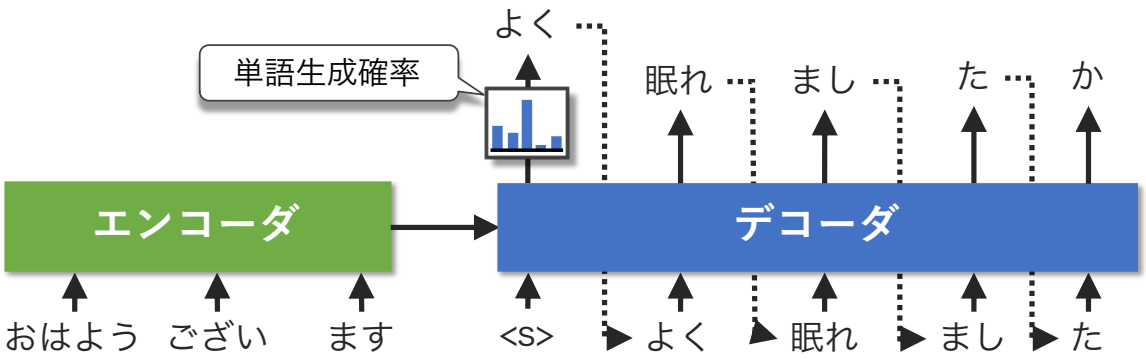

図 2.1　Sequence-to-Sequence モデルの概要図

うなモデルとなっており，ある入力文 $\mathbf{X}$ が与えられたときの出力文 $\mathbf{Y}$ の条件付き確率を予測するモデルとして捉えることができる．よって，$\mathbf{Y}$ の生成確率は以下で定式化できる．

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{T} P(y_t|\mathbf{X}, y_{t-1}, ..., y_0)$$

なお，単純な RNN では言語データのような長期の系列データを入力すると勾配消失と呼ばれる現象が起き，うまく学習ができなくなる問題が生じる．そのため seq2seq の実装においては通常の RNN に代わり LSTM [21] や GRU [4]，Self-Attention Networks (SAN) [5] などが用いられる．LSTM は通常の RNN とは異なり，メモリセル と呼ばれる内部状態を保持するためのセルを入力とは独立に持っており，その内部状態をどのように更新するかを決める入力ゲート，忘却ゲート，出力ゲートと呼ばれる 3 つのゲートを持つ．入力ゲートは現在のタイムステップにおける入力データの情報をどの程度取り込むかを決め，忘却ゲートは次のタイムステップにどの程度現在の内部状態を渡すかを決める．出力ゲート は現在の出力にどの程度メモリセルの情報を反映させるかを決める．この構造によって過去の情報を効率よく保持することができ，勾配消失が起きにくくなる．GRU はメモリセルを持たないが，現在の内部状態の計算に過去の内部状態をどの程度反映させるかを決めるリセットゲートと，現在の内部状態をどの程度次のタイムステップに反映させるかを決める更新ゲートを持つ．LSTM の構造をより簡素化したものとして位置付けられ，LSTM よりも高速に動作する．SAN は RNN とは異なり再帰的な構造を持たず，あるタイムステップの出力を入力系列中の全てのタイムステップ

の内部状態の重み付き線形和として表現する．また，重みの計算自体にも入力系列を用いる．SAN を用いた seq2seq は Transformer と呼ばれる [5].

seq2seq は機械翻訳の他にも様々なタスクへの適用が可能である．例えば，長文をその要約文へ変換する文要約 [22, 23] タスクや，ある文を同じ意味を保ったまま別の文体の文に変換するスタイル変換 [24, 25] タスク，難解な文を文意を保ったまま簡単な文に変換するテキスト平易化 [26, 27] タスクなどへの応用事例がある．対話応答生成もある発話文をその発話文に対応する応答文へ変換するタスクとして捉えることができる．対話応答生成モデル [2] は seq2seq 型のモデルを対話応答生成に適用したモデルの総称であり，入力として発話文を受け取り，応答文を生成するようなモデルアーキテクチャになっている．対話応答生成モデルの登場により，対話コーパスを基に容易に対話システムを構築できるようになった.

近年では BART [28] や T5 [29] のように，Transformer モデルを大規模なデータを用いて様々なタスクで事前訓練したモデルが登場している．これらの事前訓練済みモデルを目的タスクのデータで再訓練することで，対話応答生成を含めた様々なタスクで最高性能を達成している [30, 31, 32, 33, 34].

2.2 対話応答生成モデルの評価

対話応答生成モデルの性能評価において現在一般に用いられている方法について説明する．機械翻訳において最も広く用いられている自動評価尺度として，**BLEU** [35] が挙げられる．BLEU は，ある入力文を与えられた時にモデルが生成した文（生成文）と，その入力文に対して出力すべき正解（参照文）の間の n-gram の一致度を測る尺度となっている．コーパス全体での BLEU は以下の計算によって得られる.

$$\mathrm{BLEU} = \mathrm{BP}_{\mathrm{BLEU}} \cdot \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

一般的に $N = 4$ とすることが多い．ここで,

$$p_n = \frac{\sum_i \text{生成文 } i \text{ と参照文 } i \text{ で一致した } n\text{-gram の種類数}}{\sum_i \text{生成文 } i \text{ 中の全 } n\text{-gram 数}}$$

であり，生成した n-gram の Precision を取る操作と似た定義となっている．ただし，通常の Precision とは異なり，分子においては一度カウントした n-gram は二

度とカウントしない（Modified Precision）．例えば生成文が {the, the, the, the, the, the, the.} で，参照文が {the cat is on the mat.} であったような場合を考える．単純な uni-gram の Precision では $\frac{7}{7}$ となってしまい低品質な生成文でありながらも関わらず高く評価されてしまうのに対し，Modified Precision は $\frac{2}{7}$ となり，より実態に即した評価値を得られることがわかる．また，$\mathrm{BP}_{\mathrm{BLEU}}$ は Brevity Penalty と呼ばれ，生成文の単語数 c が参照文の単語数 r よりも短い場合にペナルティを与える項であり，以下で定義される．

$$BP_{\mathrm{BLEU}} = \begin{cases} 1 & (c > r), \\ \exp^{1-\frac{r}{c}} & (c \leq r), \end{cases}$$

翻訳においては原言語文の翻訳候補に対話ほどの多様性がないため，一般的には BLEU が高ければ高いほど良い翻訳器であると言われる．対話においては発話と応答が多対多の関係にあるため，BLEU によるシステム同士の比較は必ずしも妥当であるとは言えない．しかし，Liu ら [36] や Zhao ら [37] による研究では，BLEU と人手評価の間には中程度の相関があり，他のいくつかの自動評価尺度よりも高い相関を示すことが明らかになっている．

また，**Perplexity** も頻繁に用いられる尺度の一つである．Perplexity は元々言語モデルの流暢性の評価に用いられてきた尺度である．対話応答生成モデルにおける Perplexity は，ある入力発話 $\mathbf{X}$ とその対となる参照応答 $\mathbf{Y} = y_0, y_1, ..., y_T$ が与えられたとき，以下で計算される．

$$\mathrm{Perplexity} = \prod_{t=1}^{T+1} P(y_t|\mathbf{X}, y_{t-1}, ..., y_0)^{-\frac{1}{N}}$$

対話応答生成モデルにおける Perplexity は，直感的には $\mathbf{X}$ が与えられたときに $\mathbf{Y}$ を生成する難しさの度合いを示す．つまり，その値が小さければ小さいほど，モデルにとっては与えられた入力発話に対して正解となる参照応答を生成するのが簡単であるということを示す．よって Perplexity は小さいほど良いとされる．

これらの他にも NIST [38], dist [39] など，様々な自動評価尺度が用いられている．使用するコーパスやタスクに応じて，適切な自動評価尺度を複数併用することが多い．また，近年では人手評価との相関が高い自動評価尺度を深層学習ベースの手法で構築する取り組みが複数なされている [40, 37].

雑談を目的とした対話システムなどの評価においては特に，人手評価を行うことが重要である．人手評価の方法もタスクやモデルの設計目的によって様々である．例えば，文献 [41, 16] では，モデルが生成した応答に対して 3 段階での評価を行い，その平均値によって他のモデルとの比較を行っている．また，文献 [42, 43] などのように，モデルが満たすべき性質を複数定義し，それらに基づいて複数の観点で人手評価を行うものもある．2 つのモデルが生成した応答のうちどちらの方がより望ましい応答か（満たすべき性質をより満たしているか）を選ばせ，その勝敗の数を比較するような方法を取ることもある [44]．この方法には，数段階での絶対評価では比較が難しいような繊細な観点での比較を行うことができるというメリットがある．

本論文においても，BLEU や Perplexity の他に様々な自動評価尺度・人手評価方法を用いる．それぞれの詳細については各章で改めて解説する．

第 3 章　抽象的な発話の具体的な発話への事前言い換えに基づく対話応答生成

3.1　はじめに

対話において人間はしばしば自身の要求や意図について具体的に言及せず，抽象的に表現する [45]．人間は対話相手から抽象的な発話を受け取ったとき，これまでの対話履歴などの文脈に基づいて言外の意図を推測できる．図 3.1 に，レストランの予約に関する対話における抽象的な発話と具体的な発話の例を示す．この例ではオペレータの「A レストランを予約しますか？」という質問に対してユーザは「予算が少ないのですが」という発話を返している（図中の「抽象的な発話」）．この発話は字義通りの意味だけを考慮するとオペレータの質問への具体的な回答にはなっていない．しかし，オペレータは対話履歴を考慮してユーザが A レストランよりも安いレストランを探していると推論し，新たに A レストランよりも安い B レストランを提案している．人間と円滑なコミュニケーションを行う対話システムの実現のためには，ユーザの抽象的な発話に暗示された意図（具体的な発話）を推定する技術の実現が重要である．

大規模な対話コーパス [46, 47, 48] と深層学習技術により，近年では対話応答生成 [49, 50] や対話状態追跡 [51, 30] など様々なタスクにおいて高い性能を誇る手法が提案されている．また，最近では抽象的な発話に関するコーパス [52, 53] も構築されている．しかし，Pragst ら [52] のコーパスはルールベースで半自動的に構築されたコーパスであり，多様性や自然さに欠ける．また，Louis ら [53] のコーパスは Yes/No 型かつ一問一答型の質問応答対のみを扱うコーパスであり，それ以外の抽象的な発話には対応できない．抽象的な発話の意図を解釈する対話システムの実現のためには，より複雑かつ自然な抽象的な発話を含む対話コーパスの構築が必要である．

本研究ではより複雑な抽象的な発話を理解する対話システムの開発のために，71,498 の抽象的な発話と，その意図を明示的に表現した具体的な発話の対からなる

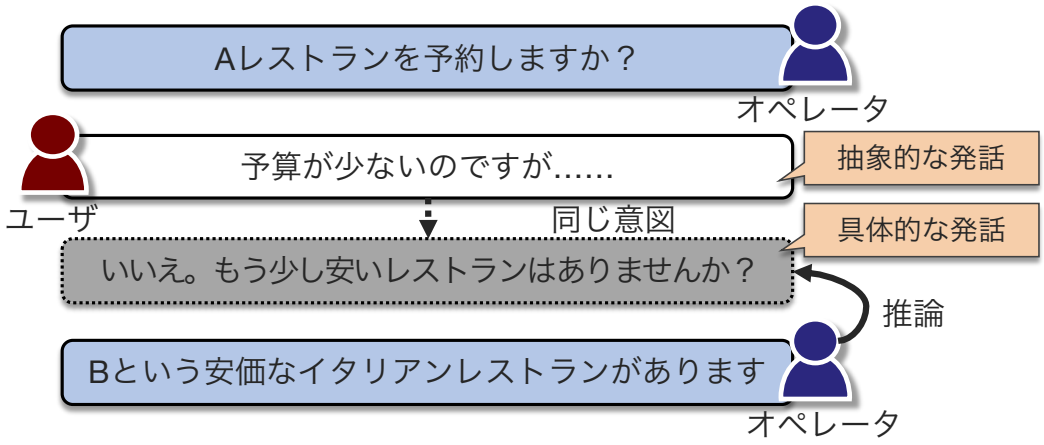

図 3.1　対話における抽象的な発話と具体的な発話の例．これらの発話は字義通りに解釈すると異なる意味を持つが，この対話履歴上においては同じ意図の発話として解釈できる．

対話コーパス DIRECT (Direct and Indirect REsponses in Conversational Text)[1] を構築する．既存の対話システム研究の成果の再利用を容易にするため，本コーパスは既存のマルチドメイン・マルチターンのタスク指向対話コーパス MultiWoZ [48] を拡張して作成する．具体的には，MultiWoZ の各ユーザ発話に対してクラウドソーシングを用いて「ユーザ発話をより抽象的に言い換えた発話」と「ユーザ発話をより具体的に言い換えた発話」の対を収集する．そのため，DIRECT コーパスには元の発話・抽象的な発話・具体的な発話の 3 つ組が収録される．

モデルが抽象的な発話の意図を正確に解釈できるかどうかを検証するために，本研究では DIRECT コーパスを用いて，抽象的な発話を具体的な発話へと言い換えるタスクを設計する．ベースラインとして最先端の事前学習済み言語モデルである BART [28] に基づく言い換えモデルを構築し，性能調査を行った．また，言い換えモデルを用いてユーザの入力発話を事前により具体的な発話に言い換えることで，対話応答生成の性能が向上することを確認した．

本章の構成を記す． 3.2 節では本研究の関連研究を紹介する． 3.3 節では DIRECT コーパスの構築方法について述べたのち，データ例や統計的な分析結果を基にコーパスの特徴について説明する．3.4 節では，抽象的な発話が対話応答生成モデルの応答に及ぼす影響を調査する．3.5 節では DIRECT コーパスを用いて，抽象的な発話を同じ意図を持つより具体的な発話へと言い換えるモデルを構築し，

1） https://github.com/junya-takayama/DIRECT/

その性能を評価する．3.6 節では，入力発話をより具体的な発話へと言い換えることで対話応答生成モデルの性能が向上することを実験的に示す．最後に，本章のまとめを 3.7 節にて述べる．

3.2 関連研究

3.2.1 抽象的な発話に関する研究

本研究において扱う「抽象的な発話」は，間接発話行為 [45] を純粋なテキストベースの人間－システム間対話で起こりうる範囲の問題に限定したものとして位置づけられる．間接発話行為とは，字義通りの解釈とは異なる効力を持つ発話のことである．例えばある駅において A が B に対し「出口はどちらかわかりますか？」と尋ねたとする．これは字義通りに解釈すれば「わかるかどうか」を問うだけの疑問文であるが，B としてはその発話が「出口の場所を教えてほしい」という要請を伝えるものだと判断するのが自然である．間接発話行為とは逆に，発話の字義通りの解釈がそのまま効力を成す発話を直接発話行為と呼ぶ．間接発話行為は対話システム分野においても重要な課題として取り組まれてきた [54, 55, 56] が，具体性の異なる発話間の言い換え関係を直接的に扱ったコーパスの構築に取り組んだのは本研究が初めてである．

相手への要求を抽象的に表現することは，対話において互いの立場や心的距離を損なわないようにするためのポライトネス戦略の一つである [57]．対話システム分野においてもポライトネスは重要な観点であり，ポライトな振る舞いをする対話システムの実現に向けて多くの研究が多くなされている [58, 11, 59]．本研究はポライトネスを直接扱うものではないが，DIRECT コーパスはポライトネス研究においても活用が期待できる．

3.2.2 対話における言い換えに関する研究

本研究では対話における言い換えに着目したコーパスを構築する．そこで本節では，対話システム分野において言い換えや抽象的発話と具体的発話の関係性を扱った関連研究を紹介する．対話における発話の言い換えを扱った事例としては文献 [60, 61] が挙げられる．これらの研究では言い換え生成技術を用いて発話の言い

換えを生成することでデータ拡張を行い，応答生成の性能向上を達成している．しかし，発話の言い換えにおいては字義的な意味のみを考慮しており，抽象的な発話と具体的な発話の言い換え関係には対応していない．文献 [52] は抽象的な発話と具体的な発話からなる言い換え対が含まれた対話コーパスをルールベースで自動構築する手法を提案している．また，再帰的ニューラルネットを用いた発話選択器を用いることで，対話中の抽象的な発話をその発話と同じ意図を持つ具体的な発話に置き換えられることを示している．しかし，ルールベースであるために自動生成可能な対話データのパターンは限られており，多様性に欠けるコーパスとなっている．文献 [53] は，Yes/No 型の質問と，それに対する抽象的な応答の対 34,268 件からなるコーパスを構築している．各質問応答対に対しては，応答が質問に対して肯定と否定のどちらの意図を示すものであるかがアノテーションされている．コーパス構築にあたってはまず著者らが事前に用意した 10 パターンの対話シナリオのいずれかに基づく質問をクラウドソーシングを用いて収集しており，質問数は合計で 3,431 件である．また，それぞれの質問に対して最大で 10 件の応答をクラウドソーシングを用いて収集している．そのため，人間らしくかつ多様な質問応答対からなるコーパスとなっている．しかし，一問一答型かつ Yes/No 型の質問応答対に限定されており，Yes/No で言い換えられないような抽象的応答には対応していない．本研究ではこれらの既存研究とは異なり，人手で作成された対話履歴に対して人手で抽象的発話・具体的発話の言い換え対を作成する．そのため DIRECT コーパスは人間らしく多様な発話が含まれ，かつ複雑な抽象的発話を扱う最初の対話コーパスである．

3.2.3 言い換えに関する言語資源

言い換えに関連する既存のコーパスを紹介し，本研究で扱う抽象的な発話と具体的な発話の言い換えの位置付けを示す．これまでに多くの言い換えコーパス [62, 63] が提供されている．文献 [62] は，Web ニュース上から単語の一致率などに基づくヒューリスティックな手法によって言い換え候補の文対を抽出し，それらの文対が言い換えになっているかどうかを人手でラベル付けすることで，5,801 件の言い換えコーパスを構築している（うち 3,900 件が言い換え対と判定されてい

る）．文献 [63] は Twitter[2)] 上から同じ URL を参照するツイートの対を言い換え候補対として収集し，それらのツイート対が言い換えになっているかどうかを人手でラベル付けしており，51,534 件の言い換えコーパスを構築している（うち 12,988 件が言い換え対と判定されている）．これらの言い換えコーパスは全て対話履歴等の文脈を伴わないものであり，対話履歴を伴う抽象的な発話の言い換えに着目したコーパスは存在しない．

含意関係認識タスクに関するコーパス [64, 65, 66] も本研究との関連が深い．含意関係とは，ある 2 文 t_1 と t_2 があったとき，t_1 が真であればそのことから t_2 も真であることが推論できるような関係のことを言う．含意関係認識コーパスには字義通りの意味だけではなく世界知識や語彙の上位下位関係なども考慮しなければ解けない問題も含まれている．対話における含意関係を扱ったコーパスとしては DialogueNLI [67] がある．このコーパスはシステムの持つペルソナとシステムが生成した発話との間の意味的な一貫性の向上を目的として構築されたものである．ペルソナ情報が付与された対話コーパスである Persona-Chat [12] を基に，ペルソナ情報と各発話との間での含意関係情報を付与している．

既存のコーパスにおいては，文同士が言い換え関係にあるかや文が仮説を含意するかどうかは，その文が用いられる文脈等を考慮せずとも世界知識や文の明示的な意味のみに依存して判断可能である．これに対し本研究で扱う抽象的な発話の言い換えでは，発話同士が言い換え関係にあるかどうかを判断するためには対話履歴等の文脈が重要な要素となる．

3.3 DIRECT コーパス

抽象的な発話と具体的な発話の言い換え対からなる対話コーパスを構築する．データ作成コストの低減と，既存の対話システム研究との互換性の確保のために，本研究では既存の対話コーパスを拡張する形で発話対を収集する．具体的には MultiWoZ2.1（以降は MultiWoZ と表記） [47, 48] コーパスを基に，クラウドソーシングを用いて発話対を収集する．

本節ではまず発話対の収集方法と得られたデータ例について 3.3.1 節で説明す

2） https://twitter.com

表 3.1　抽象的発話と具体的発話の収集のための指示書

Instructions
Read the following dialogue between the USER and the OPERATOR, please rephrase the USER's response written in red letters into two different types of speech, following the instructions below. **Type-1 (Direct) :** a more direct response that expresses the same intention as the original response **Type-2 (Indirect) :** a more indirect but natural response that expresses the same intention as the original response "Indirect response" means, for example, a response to a Yes/No question that does not contain a "Yes" or "No", or a response that does not directly refer to the action you want the other person to do or your desire. If you have trouble rephrasing, click the "Hints" button. You can see the goals that "USER" must achieve in that interaction.

る．また，収集したデータの品質評価を 3.3.2 節にて行う．構築した DIRECT コーパスの特徴や性質について，3.3.3 節では統計的な観点から分析する．3.3.5 節においては，既存の言い換え認識モデルを用いて，DIRECT と既存の言い換えコーパスの比較分析を行う．

3.3.1　抽象的な発話と具体的な発話の対の収集

MultiWoZ は 10,438 対話 からなるマルチドメインかつマルチターンの英語タスク指向対話コーパスであり，対話行為や対話状態など豊富なアノテーションが付属している．各対話はユーザとシステムの二者が交互に発話する形式であり，ユーザ発話は合計で 71,524 件存在する．

本研究ではクラウドソーシングサービスの Amazon Mechanical Turk[3)] を用いて，MultiWoZ を拡張する形で抽象的な発話と具体的な発話の対を収集する．作業者に

3） https://www.mturk.com/

はまず表 3.1 の指示書と作業例を提示する．また，図 3.2 のように，各作業者の作業画面上には MultiWoZ から抜粋された対話履歴が表示される[4]．表示された対話履歴に基づいて，各作業者は指定されたユーザ発話（図 3.2 中の赤字の発話）を抽象的な発話と具体的な発話のそれぞれに言い換え，入力フォームに入力する．なお，対話履歴のみでは話者の意図を十分に理解できない場合に備え，「Hints」ボタンを押すことで MultiWoZ から抜粋したその対話の対話目標を表示できる機能を作業画面上に実装する．

作業者の選定　数単語の置き換えや入れ替えしか行っていないデータや，オリジナルの発話の言い換えにもなっていないようなデータの収集を防ぐため，本番の収集タスクに登用する作業者を事前に選定した．具体的には，2 回のパイロットタスクを実施した．なお，2 回目のタスクにおいては 1 回目のタスクにおいて作業者から受けた質問や得られたデータの品質などを基に指示書を部分的に修正し，本番のタスクにおいて使用したものと同一の指示書を使用した．本番タスクにおいては，抽象的な発話と具体的な発話の間の単語レベルの Jaccard 係数が 0.75 以上のデータを自動的に不採択にした．また，得られたデータの抜き打ち検査も実施した．最終的には，これらの手動評価と自動評価に合格した 655 人中の 536 人の作業者によってデータが作成された．

発話対の収集　1 件あたりの作業時間を平均 1 分と見積もり，平均報酬は 1 件あたり 0.12 米ドル（1 時間あたり 7.2 米ドル．2021 年 6 月 23 日現在のレートで日本円換算すると約 799 円）とした．最終的には 71,498 件[5]の抽象的な発話と具体的な発話の対を収集できた．得られたデータは，MultiWoZ と同じ設定で訓練データと評価データに分割した．

データ例　収集した抽象的発話と具体的発話の例を表 3.2 に示す．上段の例におけるオリジナルのユーザ発話（MultiWoZ に元々収録されている発話）を見ると，ユーザは中間的な価格帯のレストランを希望していることがわかる．この発話を抽象的に言い換えたものとして “I don’t want to overspend but remember its also vacation”（あまり使いすぎたくはないけれど，休暇でもあるということを忘れない

4） 元々の MultiWoZ データは システム（system）役とユーザ（user）役による二者間対話からなる．しかし本研究においては，人間同士の対話において自然に発生するような抽象的な発話を収集するために，システム役の発話を “operator” による発話として表現し，人工的に生成されたものだと認識されないようにする．

5） 自然言語文として成り立っていない 26 件のデータを除外した

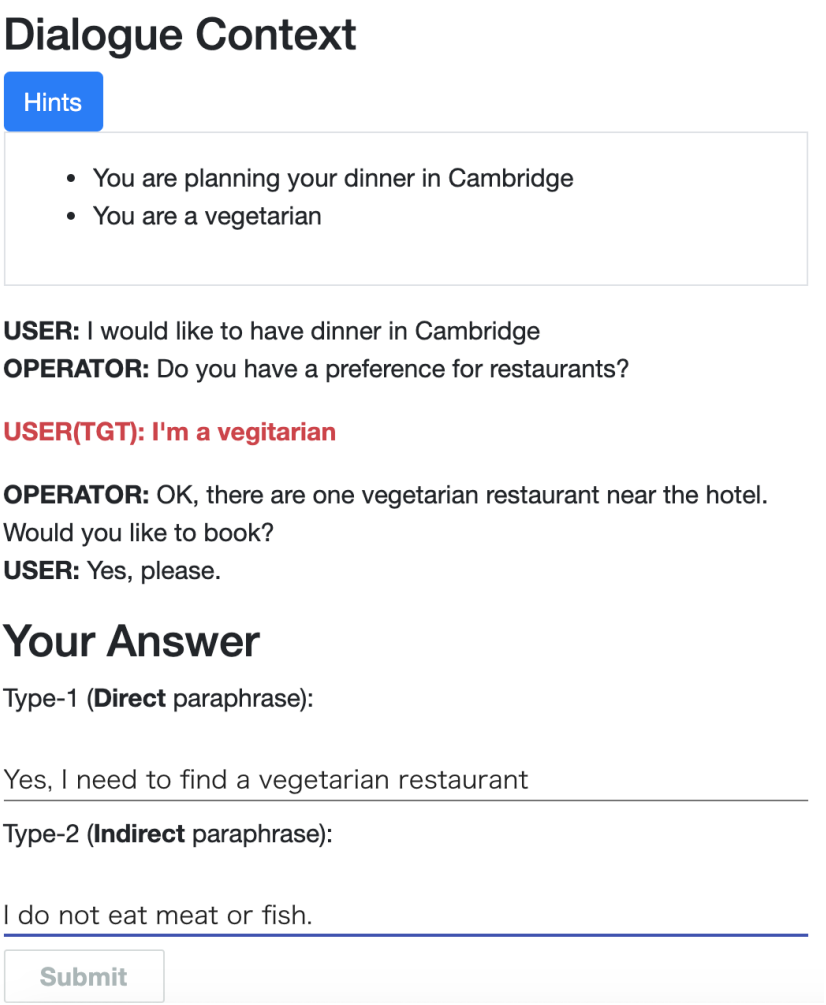

図 3.2　クラウドソーシングを用いた抽象的な発話と具体的な発話の対の収集に用いる作業画面例

で）という発話が得られた．この例では “its also vacation” というフレーズが具体的な発話における “not too cheap”（安すぎない）に対応している．下段の例では，抽象的な発話の中に “Do you know of any in town?” というフレーズがある．対話履歴を考慮すれば，これは単に「何か街にあるもの」を知っているかどうかを問うための発話ではなく，具体的な発話における “Can you find me a guesthouse...?” のように「ゲストハウスを探してほしい」という要求を伝えるための発話であることが汲み取れる．

表 3.2　DIRECT コーパスのデータ例．“USER (抽象的)” と “USER (具体的)” はそれぞれ本研究においてクラウドソーシングによって収集した抽象的発話と具体的発話で，“USER (オリジナル)” は MultiWoZ から抽出したオリジナルの発話である．

話者	発話
	(中略．ユーザはレストランを探している)
SYSTEM	Would you like to pick a different type of food?
USER	Yes, what about British food please.
SYSTEM	What price range are you comfortable with?
USER (オリジナル)	Something in the moderate price range would be good.
USER (抽象的)	I dont want to overspend but remember its also vacation.
USER (具体的)	Can you choose something that is not too expensive and not too cheap.
SYSTEM	Do you have a preference as to what area of town you dine in?

speaker	utterance
USER	I need a place to stay in the north
SYSTEM	OK im seeing alot of choices in hotels is there anything else You need in the hotel that would help narrow it down
USER (オリジナル)	I'd really like to stay in a guesthouse. I heard the ones in Cambridge are very nice.
USER (抽象的)	I am thinking of staying in a guesthouse. Do you know of any in town?
USER (具体的)	Can you find me a guesthouse in Cambridge?
SYSTEM	How about the Acorn Guesthouse? It is rated 4 stars and is in the moderate price range.

3.3.2　品質評価

発話対の収集後，最終的な成果物の品質評価を実施した．具体的には評価セットに含まれる 7,372 件分のユーザ発話に対し，得られた抽象的な発話と具体的な発話の対の品質をクラウドソーシングを用いて評価した．品質評価において作業者に提示した作業画面の例を図 3.3 に示す．作業者には評価対象となる言い換え対とその対話履歴を提示した．その際，言い換え対には “Response-A”, “Response-B”

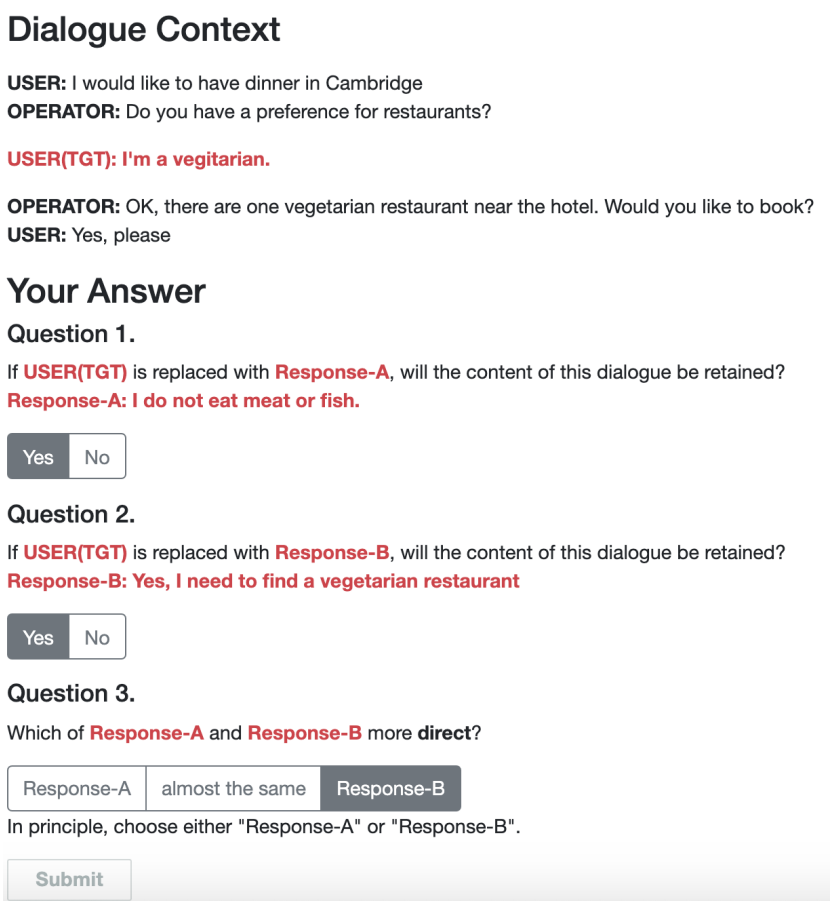

図 3.3　クラウドソーシングを用いたコーパスの品質評価に用いる作業画面例

のラベルをランダムに割り当て，どちらが抽象的（あるいは具体的）な発話として収集されたデータかわからないようにした．作業者にはまず，言い換え対が実際に MultiWoZ のオリジナルの発話と同じ意図を表現できているかどうかを “Yes”, “No” の二値で評価してもらった．次に，Response-A と Response-B のどちらがより具体的かについても評価してもらった．なお，判断に迷った場合のみ “no difference” を選ぶことを許容した．1 件あたりの作業にかかる平均時間を 30 秒と見積もり，平均報酬は 1 件あたり 0.06 米ドル（1 時間あたりでは 7.2 米ドル．2021 年 6 月 23 日現在のレートで日本円換算すると約 799 円）とした．各言い換え対に対して 5 人の作業者による評価を収集した．最終的な評価値はその多数決によって決定した．なお，評価タスクにおいては，収集タスクに参加した作業者が自分が作成したデータを自己評価することがないように作業者の割り当てを行った．

評価結果を表 3.3 に示す．Intention-accuracy は収集した発話のうち，オリジナ

表 3.3　評価データの品質評価結果

Metric	Ratio [%]
Intention-accuracy (抽象的)	95.0
Intention-accuracy (具体的)	99.7
Directness-accuracy (Exact)	81.4
Directness-accuracy (Relaxed)	89.4

表 3.4　収集した言い換え対の統計情報

尺度	値 [単語]
語彙サイズ（抽象的な発話）	6,273
語彙サイズ（具体的な発話）	4,664
平均文長（抽象的な発話）	15.59
平均文長（具体的な発話）	12.38

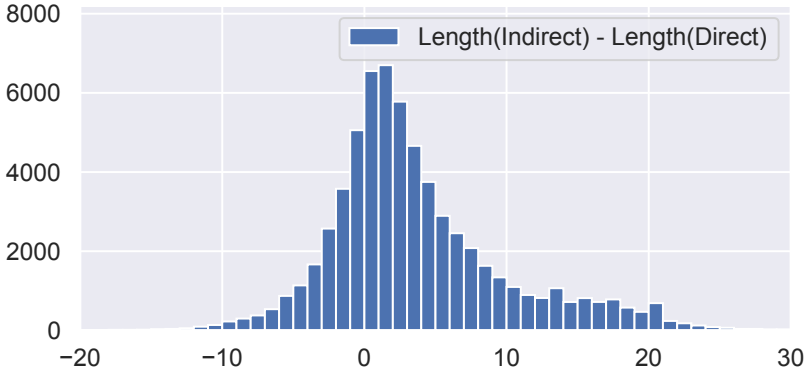

図 3.4　抽象的発話と具体的発話それぞれの文長の分布

ルの発話と同じ意図を表現していると判断されたものの割合である．抽象的な発話への言い換えでは 95.0%，具体的な発話への言い換えでは 99.7% と高い割合のデータがオリジナルと同じ意図を持っていることがわかる．抽象的な発話の Intention-accuracy が具体的な発話の場合より 4.7% 低い結果となった．これは，ユーザの意図を抽象的に表現しているために，人間の評価者にとっても解釈が困難な曖昧な表現が多少含まれているためであると考えられる．Directness-accuracy は，Response-A と Response-B のどちらがより具体的かとの問いに対し，具体的な発話として収集された方の発話が正しく選ばれた割合である．“Exact” は “no difference” と判断されたデータを許容しない設定，“Relaxed” は許容した設定での Directness-accuracy であり，それぞれ 81.4%, 89.4% と高い割合となった．DIRECT コーパスにおいては，これらの評価ラベルも提供する．

3.3.3　統計的分析

DIRECT コーパスにおける抽象的な発話と具体的な発話の言い換え対の特徴を明らかにするために，まずトークン単位での統計的分析を行う．表 3.4 に

単語レベルでの統計情報を示す．なお，単語分割には nltk[6] [68] ライブラリの ‘word_punct_tokenize()’ メソッドを用いた．また，大文字小文字は区別していない．まず，語彙サイズについては，抽象的な発話の方が具体的な発話よりも 1.3 倍程度大きいことがわかる．これは，同じ意図を持った発話であっても，具体的に表現した場合よりも抽象的に表現した場合の方がより多様な表現を取り得ることを示唆している．また，文長（発話に含まれる平均単語数）に関しては，抽象的な発話では 15.59，具体的な発話では 12.38 で，抽象的な発話の方が長い．平均文長について Wilcoxon の検定 [69] を実施したところ，0.1% 有意水準で有意であった．図 3.4 に，対応する抽象的な発話と具体的な発話の文長の差のヒストグラムを示す．図を見ると，文長の差はやや正の側に偏りつつも，負の側にも多く分布していることが読み取れる．このことから，平均文長に有意差はあれど，発話をより具体的に言い換える際に単に文長を短くすることが必ずしも有効というわけではないことが示唆される．

次に，抽象的な発話と具体的な発話において使用されるフレーズの違いを調査する．表 3.5 に，それぞれにおける単語 tri-gram の頻度上位 20 件を掲載する．表を見ると，具体的な発話の tri-gram には “book” や “find” などユーザがオペレータにして欲しいことを具体的に伝える動詞や，“the reference number” のように特定のオブジェクトを指すフレーズが頻繁に含まれていることがわかる（表中太字で表記）．一方，抽象的な発話の tri-gram には，具体的な発話では頻出でない “is there any” や “I think that” などのフレーズが含まれている．以上のことから，抽象的な発話と具体的な発話では，それぞれ頻出するフレーズに違いがあることが読み取れる．

3.3.4 定性的分析

DIRECT コーパスにおいて，具体的発話と比較して抽象的発話にはどのような特徴があるかを調査するため，訓練データから無作為に抽出した 300 件の発話対について類型化を試みる．第一著者による分析の結果，発話対は以下の 5 つの類型に大別できた．以下に各類型の割合とその説明を記す．なお，複数の類型に該当する発話もあったため，割合の和は 100% に一致しない．

要求や行為の曖昧化（54.0%） 具体的発話では明示的に言及されている要求や行

6） https://www.nltk.org/

表 3.5　頻度上位 20 件の tri-gram

抽象的な発話		具体的な発話	
trigram	freq	trigram	freq
i want to	2387	**find me a**	2617
i need to	2223	i want to	2442
would like to	1969	**can you find**	1971
i would like	1903	**please find me**	1924
is there any	1762	all i needed	1807
that would be	1588	thanks for the	1643
you help me	1584	in the centre	1628
thanks a lot	1407	i need to	1623
in the centre	1402	for the help	1539
i think that	1312	**you find me**	1531
i think i	1270	that's all i	1403
a place to	1246	can you get	1376
you have been	1242	give me the	1358
would be swell	1056	i need a	1206
such a great	1042	you get me	1200
i think you	1027	i would like	1145
you have done	1011	please give me	1097
a great help	981	get me a	1094
have been such	979	**book it for**	1086
been such a	979	**the reference number**	1060

為について，抽象的発話ではより曖昧になっているもの.

例 1: (ユーザがレストランを探している文脈で)

具体的発話: Find me a good gastropub in town.

抽象的発話: It would be ideal if there were good gastropubs in town. (「探して」という要求が明示されていない)

例 2: (システムの「Pizza Hut Cherry Hinton serves Italian food on the south part

of town. Would you like their phone number?」という発話に対して)

具体的発話: Yes, you can get me the phone number.

抽象的発話: I will want to call them. (「(電話番号を) 提供してほしい」という要求が明示されていない)

対象の曖昧化（18.3%） 具体的発話では明示的に言及されている物体や概念について，抽象的発話ではより曖昧になっているもの.

例 1: (映画館の予約が済んだ後で)

具体的発話: Yes, I also need the address of the **theatre**?

抽象的発話: I've never been there. I'll need help finding it. (theatre が省略されている)

例 2:（システムの「I'm sorry but I was unable to book you for that time. Would you like to try another day or another time slot?」）という発話に対して

具体的発話: No thanks, any other **restaurant** for a similar time?

抽象的発話: Nah, got anything else like it? (restaurant が省略されている)

対話状況に依存しない言い換え（10.3%） 具体的発話におけるある表現が，対話状況に依存しない形でより抽象的な表現に言い換えられているもの.

例 1: (システムがユーザに立地の希望を聞いた文脈で)

具体的発話: Yea the **north** sounds like it would work out for me.

抽象的発話: I am just hoping to be **as far from the south side as possible**. (対話状況によらず,「north」と「as far from the south side as possible」は言い換え可能)

例 2: (システムがユーザに宿泊先の価格帯の希望を聞いた文脈で)

具体的発話: A **reasonably priced** lodge would be wonderful.

抽象的発話: I only stay in **affordable** lodges, without exception.. (対話状況によらず,「reasonably priced」と「affordable」は言い換え可能)

丁寧さ・モダリティの調整（41.3%） 抽象的発話において，具体的発話に比べてより丁寧な言い回し（"Could you…" や "Would you…" など）や断定を避けるようなモダリティ表現（"I think…" や "maybe" など）が付与されているもの.

例 1:

具体的発話: Please book for 4 people on Tuesday at 12:00

抽象的発話: Can you make a reservation for 4 people at 12:00?

表 3.6　言い換えかどうかの判定に対話履歴の有無が与える影響の調査結果

	言い換えと判定された割合 [%]	
	対話履歴提示あり	対話履歴提示なし
抽象的発話 - オリジナル発話	96.0%	83.6%
具体的発話 - オリジナル発話	99.6%	94.8%

例 2:

具体的発話: French food please. Any price range is fine.

抽象的発話: Can i get some French food? Any price range is fine.

自己開示の追加（7.0%）　発話意図を抽象的に伝えるために，自己に関する何らかの情報を追加しているもの．

例 1:

具体的発話: Yea can you find me a place to eat that is in the expensive category.

抽象的発話: **I want to take my girl** to the nicest place to eat in town.

例 2:

具体的発話: I have to leave Cambridge after 20:30.

抽象的発話: **I will finish my meeting** and can depart from Cambridge only after 20:30.

このうち，「要求や行為の曖昧化」と「対象の曖昧化」については抽象的発話の意図の解釈が対話履歴に依存して変わりうるもの（対話履歴に依存した言い換え）であると言える．これら二つの類型の少なくともどちらか一方に属するデータの割合は 60.3% であった．よって，DIRECT コーパスのうち 6 割程度は対話履歴に依存した言い換えが行われていると考えられる．

次に，発話の意図を人間が解釈する際に，対話履歴の有無がどの程度影響を与えるかを調査する．検証データから無作為に抽出した 500 件の発話対について，Amazon Mechanical Turk を用いて，それらがオリジナルの発話と言い換え関係にあるかどうかを，対話履歴を提示した場合と提示しない場合のそれぞれについて判定してもらった．各発話につき 5 人の作業者による判定を行ったのち，多数決によって最終的なラベルを決定した．表 3.6 に言い換えと判定された割合を示す．対話履歴を提示した場合，抽象的・具体的発話はともにほぼ全ての発話がオリジナル

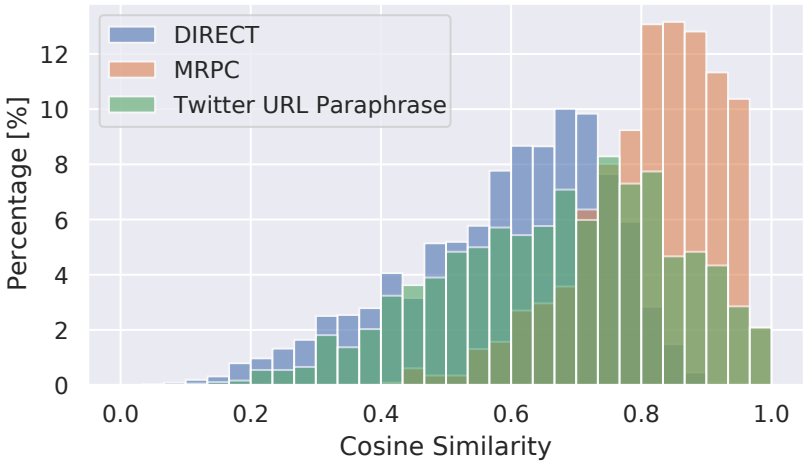

図 3.5　Sentence-BERT を用いて算出した言い換え対のコサイン類似度の分布

発話と同義だと判定されていることがわかる．一方で対話履歴を提示しない場合は，提示した場合に比べて抽象的発話では言い換えと判定された割合が 12.3% 低い．このことから，DIRECT コーパスには対話履歴の情報が重要な言い換え対が少なくとも 12% 程度は含まれていると推定される．しかし，この値は先の類型化において「対話履歴に依存する言い換え」としたデータの割合である 60.3% よりはるかに小さい．これは，“I’m looking forward…” と “Find me…” のように必ずではないがほとんどの状況において慣習的に同じ意図で用いられるような発話対が，対話履歴を提示しなかった場合でも言い換え関係として判定されてしまったことが原因だと考えられる．

3.3.5　モデルを用いた分析

本節では，最先端の文符号化器や言い換え認識モデルを用いて，DIRECT コーパスと既存の言い換えコーパスの比較調査を行う．まず，DIRECT コーパス，MRPC [62]，Twitter URL Paraphrase コーパス [63] のそれぞれの言い換え対（DIRECT コーパスにおいては抽象的発話と具体的発話の対）全てについて，文符号化器 Sentence-BERT[7] [70] を用いてその文間のコサイン類似度を計算する．図 3.5 に示す類似度のヒストグラムを見ると，DIRECT は他の既存のコーパスに比べて類似度が低い言い換え対が多いことがわかる．Sentence-BERT は，文間の類似度を推定するタスクである STSBenchmark [71] を用いて事前訓練されたモデルで

7）`https://www.sbert.net/` にて公開されている ‘stsb-roberta-base’ モデルを用いた．

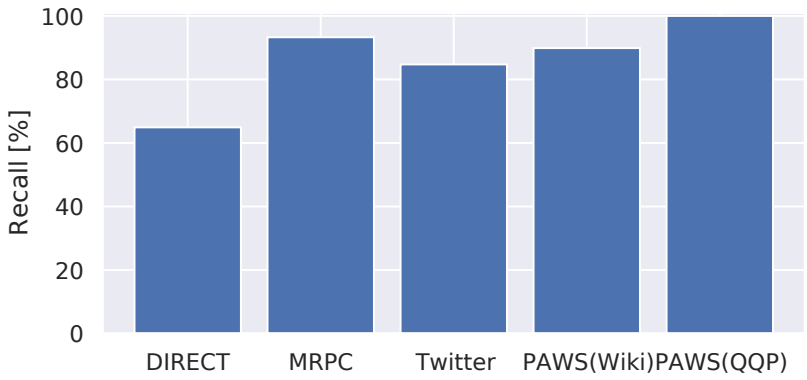

図 3.6　MRPC, Twitter URL paraphrase コーパスでファインチューニングした BERT によって言い換えだと認識された言い換え対の割合

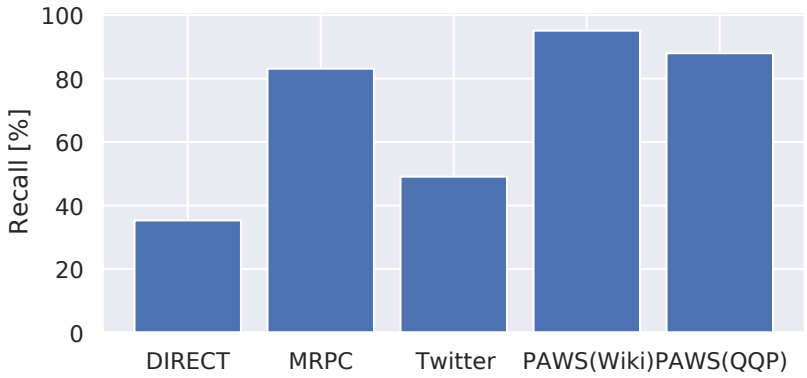

図 3.7　PAWS コーパスでファインチューニングした BERT によって言い換えだと認識された言い換え対の割合

あり，文脈を考慮しないレベルでの文の意味的類似度を扱うモデルとなっている．DIRECT ではコサイン類似度が低い言い換え対が大量に存在することから，文脈を考慮しなければ言い換えだと判断できないような言い換え対が多く含まれていることが確認できる．

次に，既存の言い換えコーパスを用いて訓練された言い換え認識モデルを DIRECT コーパスや既存のコーパスに適用して比較分析を行う．具体的には，BERT[8] [72] を既存の言い換え認識用コーパスでファインチューニングした言い換え認識モデルが，実際に言い換えだと認識できた言い換え対の割合を計算する．また，特性の異なる言い換え認識モデルでの挙動の違いを分析するため，

8）実装には transformer ライブラリ `https://huggingface.co/transformers/` の version 3.5.1 を用いた．また，事前訓練済みモデルとしては “bert-base-cased” を用いた

MRPC と Twitter URL Paraphrase コーパスで訓練したモデルに加え，Paraphrase Adversaries from Word Scrambling (PAWS) [73] で訓練したモデルについても分析を行う．PAWS コーパスは単語の重複度が高い言い換え対を扱うコーパスで，MRPC や Twitter コーパスに比べて構文レベルでの判断が必要なコーパスとなっている．図 3.6 に MRPC と Twitter URL Paraphrase を用いてファインチューニングした場合の結果を，図 3.7 に PAWS を用いてファインチューニングした場合の結果を掲載する．図 3.6, 3.7 を見ると，DIRECT コーパスはどちらのモデルにおいても言い換えと認識される対の割合がそれぞれ 64.9%，35.3% と最も低いことが読み取れる．このことから，既存の語彙や構文レベルでの言い換えに着目したモデルでは，DIRECT に含まれる抽象的・具体的な発話の言い換え対は十分に扱えないことが示される．

3.4 抽象的発話が対話応答生成に与える影響の検証

本研究では，対話応答生成モデルにとっては具体的なユーザ発話の方が抽象的なユーザ発話に比べてより正しい応答を生成しやすいという仮定に基づき，入力発話を事前により具体的に言い換えることで対話応答生成の性能向上を目指す．本節では，事前言い換えに基づく手法の構築に先あたって，ユーザ発話の具体性が対話応答生成に与える影響を調査し，仮定の妥当性を検証する．検証の方法としてはまず，ある発話の具体性を推定するようなモデルを構築する（3.4.1 節）．次に，そのモデルを用いて MultiWoZ の評価データにおけるオリジナルのユーザ発話を抽象的・具体的のいずれかに振り分ける．それぞれに分類された発話に対し，最先端の End-to-End タスク指向対話応答生成モデルである UBAR [74] を用いてシステム応答を生成する．抽象的なユーザ発話に対する応答の品質が具体的な発話に対するものよりも低いことを確認する（3.4.2 節）．

3.4.1 具体性推定モデルの構築

概要　本節では，あるユーザ発話が持つ具体性を推定するようなモデルを構築する．DIRECT コーパスには，それぞれの対話履歴に対して MultiWoZ のオリジナルの発話，抽象的な発話，具体的な発話の 3 つの発話が付与されている．これらの発話は具体性が高い順に並べると具体的な発話，オリジナルの発話，抽象的な発話

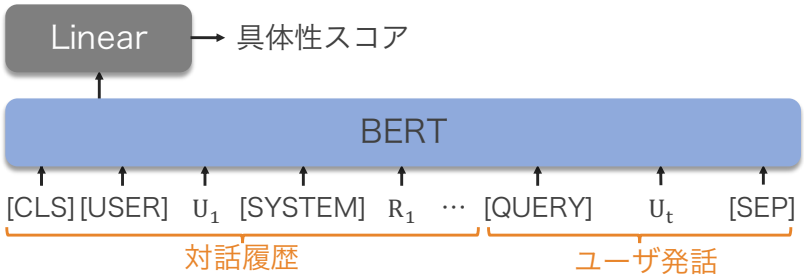

図 3.8　具体性推定モデルのアーキテクチャ

の順になる．モデルが推定した入力発話の具体性スコアを用いて 3 つの発話を降順に並び替えたときに，それが実際の具体性の順番通りに並んでいれば，モデルは良く具体性を推定できていると言える．よって，3 つの発話を正しくランキングできるような目的関数を用いて具体性モデルの設計を行うこととする．

具体性推定モデルの設計　具体性推定モデルを，対話行為推定 [75] や対話状態追跡 [76] などの対話データに対する分類タスクにおいて高性能を誇る事前学習済みモデル BERT を基に構築する．モデルアーキテクチャを図 3.8 (c) に示す．まず具体性の推定対象である発話と対話履歴を特殊トークンを挟んで連結して BERT エンコーダに入力する．“[CLS]” トークンに対応する最終層の出力を線形層に入力してスカラー値に変換し，sigmoid 関数を通して $(0, 1)$ の範囲のスカラー値を得る．最終的な出力を入力された発話の具体性を示すスコアとみなす．

3.3.3 節で議論した通り，具体的な発話と抽象的な発話のそれぞれにおいて用いられている単語やフレーズの頻度には大きな違いがあることがわかっている．そこで，単純な単語レベルの特徴量が具体性推定にどの程度有効かを検証するため，Bag-of-words ベースの線形回帰モデルに関しても比較を行うこととする．

モデルの学習には，ランキング学習 [77] タスクでよく用いられる Pointwise 損失と Pairwise 損失を用いる．Pointwise 損失では，推定値と正解の具体性スコアの間の平均二乗誤差を最小化する．なお，今回は擬似的な正解値として，抽象的な発話に 0.0, オリジナルの発話に 0.5, 具体的な発話に 1.0 のスコアを割り振る．

Pairwise 損失は，より具体的な発話の推定スコアが他方の発話よりも大きくなるような損失関数である．例えば，具体的な発話 A と抽象的な発話 B があって，その推定スコアを s_A，s_B としたとき，Pairwise 損失は以下で定義される:

$$-\log \frac{1}{1 + e^{-(s_A - s_B)}}$$

表 3.7　具体性推定タスクの評価結果

モデル	損失関数	Exact match	Kendall's tau
BERT w/o history	Pointwise	0.785	0.803
BERT w/o history	Pairwise	**0.816**	**0.846**
BERT w/ history	Pointwise	0.784	0.804
BERT w/ history	Pairwise	**0.813**	**0.841**
LR w/o history	Pointwise	0.540	0.616

BERT ベースのモデルの実装には transformers ライブラリを用いた．また，事前学習済みモデルとして "bert-base-cased" を使用した．線形回帰モデルの実装には機械学習実装フレームワークである scikit-learn (version 0.23.1)[9] を用いた．また，対話履歴を入力するモデルとしないモデルを構築した．

推定性能の評価　DIRECT コーパスの評価データを用いて，具体性推定モデルの推定性能を評価する．評価尺度としては，推定スコアに基いたランキングと正解のランキングに対して，それらが完全一致する割合（Exact match）とそれらの間の Kendall's tau [78] を用いる．表 3.7 に実験結果を示す．なお，太字は t 検定の結果 1% 水準で他の群との有意差が認められたものを示す．太字で示された 2 モデル間では有意差を認められなかった．Exact-match と Kendall's tau の両方において，対話履歴を利用しないモデルと利用するモデルではほぼ変わらない値となっていることがわかる．このことから，具体性については対話履歴を用いずともフレーズや単語のみをする手がかりとしてある程度予測可能なのではないかと考えられる．

線形回帰モデル（LR w/o history）の Exact-match は 0.540 で，これはチャンスレートの 0.167 よりもはるかに高い値となっている．これは，BERT ベースのモデルには及ばないものの，単純な Bag-of-Words 特徴量でも具体性をある程度推定可能であることが示された．

3.4.2　対話応答生成における抽象的発話の影響調査

3.4.1 節で構築した具体性推定モデルを用いて，ユーザ発話の具体性がシステム応答の生成に与える影響を調査する．まず，表 3.7 で最も高性能であった BERT

9）https://scikit-learn.org/stable/

w/o history の Pairwise 損失モデルを用いて，MultiWoZ の評価データのユーザ発話を抽象的・具体的のいずれかに振り分ける．ここで，振り分けの閾値は 0.5 に設定し，それより高いものは具体的，それ以下のものを抽象的とする．次に，抽象的なユーザ発話群と具体的なユーザ発話群のそれぞれに対して最先端の End-to-End タスク指向対話モデルである UBAR [74] を用いてシステム応答を生成する．なお，実装には著者らが公開しているスクリプトを用いた [10]．それぞれのユーザ発話群に対するシステム応答の BLEU を計算したところ，抽象的な発話群に対しては 10.25，具体的な発話群に対しては 14.09 となった．この結果から，対話システムにとっては具体的なユーザ発話の方が抽象的なユーザ発話に比べてよりシステム応答を生成しやすいという仮定は妥当であると言える．よって，ユーザ発話を予めより具体的に言い換えて対話応答生成モデルに入力することで，対話応答生成の性能は改善できると考えられる．

3.5 抽象的な発話の具体的な発話への言い換え

本節では対話応答生成モデルにおけるユーザ発話のより具体的な発話への事前言い換えのために，抽象的な発話を具体的に言い換えるタスクを設計し，実験を行う．

3.5.1 言い換えモデルの構築

本タスクは入力された発話を意図を保持したままより具体的なスタイルに置き換えるスタイル変換タスクであると言える．そこで，文平易化 [79] やフォーマル性変換 [80] などのスタイル変換タスクにおいて高い性能を記録している BART [28] をこのタスクのベースラインとして用いる．ベースラインモデルのアーキテクチャを図 3.9 に示す．特殊トークンとして “<user>”, “<system>”, “<query>” を追加する．“<user>” トークン, “<system>” トークンをそれぞれ対話履歴中のユーザ発話の直前，システム発話の直前に付与することで，モデルがそれぞれの発話の話者を区別しやすいようにする．また，言い換え対象である抽象的な発話の直前には “<query>” トークンを付与する．対話履歴中の発話と言い換え対象の発話を時系列

10） https://github.com/TonyNemo/UBAR-MultiWOZ

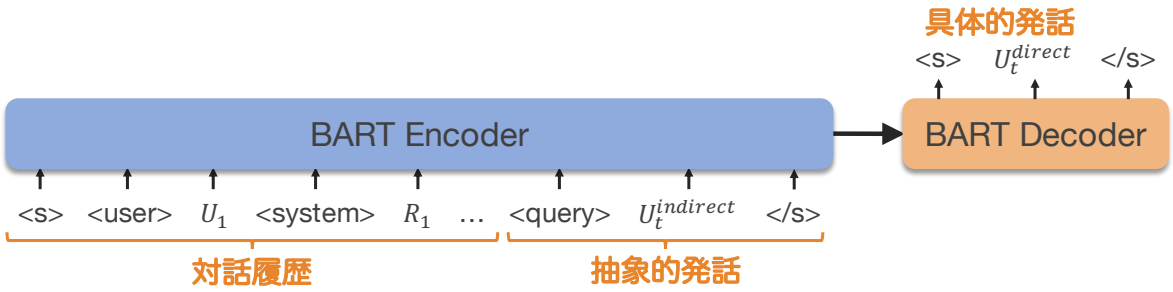

図 3.9　抽象的な発話から具体的な発話への言い換えモデル

表 3.8　抽象的な発話から具体的な発話への言い換えタスクの実験結果

Model	BLEU	Perplexity
Transformer w/ history	25.23	2.66
BART w/o history	32.51	**2.15**
BART w/ history	**33.77**	2.16

順に連結し，BART のエンコーダに入力する．なお，入力データのトークン数が 512 を超えた場合は末尾から 512 トークン目までのみを入力する．モデルのファインチューニングはクロスエントロピー損失を用いて行う．

3.5.2　実験設定

実装には transformers [81] ライブラリを使用した．また，事前学習済みモデルとして “facebook/bart-base”[11] を用いた．オプティマイザとして AdamW [82] を用い，学習率は 2e-5 [12] とした．バッチサイズは GPU メモリ容量上の上限となる 8 を採用した[13]．訓練データのうちランダムに 2,000 件をハイパーパラメータチューニング用の検証データとして抽出し，残りを訓練に用いた．30 エポックの学習を経て，最も検証データでの損失が小さいモデルを評価データでの評価に用いた．

また，対話履歴の影響を調査するために，対話履歴を入力しないモデル（BART w/o 対話履歴）も構築した．事前学習の効果を検証するために，事前学習されてい

11）`https://huggingface.co/facebook/bart-base`
12）検証データにおいて最も損失が小さくなるように選んだ．
13）使用した GPU は GeForce RTX 2080 Ti, 11GB Memory.

表 3.9　抽象的な発話の具体的な発話への言い換えタスクにおける具体的な発話の生成例

USER	Hi, I'm looking for a place to eat some indian food.
SYSTEM	Do you have a price range in mind?
抽象的な発話（入力）	I want it to be the best place in town.
BART	I want it to be an expensive place.
- w/o 対話履歴	Can you find me something expensive?
Transformer	Can you find me an attraction in town?
参照文	I was hoping for a much more expensive place if possible.
	(omitted)
SYSTEM	Yes, The Cambridge Belfry is a cheap hotel in the West.
USER	Do they have free internet?
SYSTEM	Yes, they have internet would you like me to book it for you?
抽象的な発話（入力）	i dont think so on the booking. address and phone number though.
BART	dont need to book it. address and phone number for them though.
- w/o 対話履歴	address and phone number isn't needed.
Transformer	no...just give me the address and phone number.
参照文	address and phone number is all i need right now.

ない Transformer モデルを DIRECT コーパスのみを用いて訓練したモデルについても評価を行った．なお，Transformer モデルについてはエンコーダ・デコーダ共に層数や中間層の出力次元数などの各種ハイパーパラメータを BART と同一の値に設定した．

評価尺度としては BLEU と Perplexity を用いることとした．それぞれの詳細は 2.2 を参照されたい．先にも述べたように，本タスクはスタイル変換タスクの一種として捉えることができる．BLEU はフォーマル性変換 [80] においても用いられているように，スタイル変換における標準的な自動評価尺度の一つである．また，入力文をモデルに入力した時の参照文の生成されやすさを測る Perplexity も採

用する．

3.5.3　実験結果と考察

表 3.8 に各モデルの評価データにおける BLEU と Perplexity を示す．BART を用いたモデルでは，対話履歴を考慮するモデルの方が BLEU スコアが高い．これは，抽象的な発話は対話履歴を考慮しなければ正確に具体的な発話への言い換えができないことを示唆している．

BART と Transformer については BART の方が BLEU，Perplexity 共に大きく上回っていることがわかる．この結果から，他の多くの自然言語処理タスクと同様に，抽象的な発話から具体的な発話への言い換えにおいても事前学習によって得られる知識が効果的に作用することがわかる．

表 3.9 に生成された具体的な発話の例を示す．上段の例では，BART モデルは抽象的な発話の “the best place” というフレーズを価格帯に関する表現だと解釈できていることがわかるが，一方で Transformer モデルはこの解釈に失敗している．下段の例は文脈の考慮が特に重要な例である．対話履歴を利用しない BART（BART w/o 対話履歴）では，参照文とは逆の意図を持つ文を生成してしまっている．一方，対話履歴を利用する 2 つのモデルは共に参照文と同じ意図の文を生成できている．

3.6　対話応答生成への事前言い換えの適用

対話システムにとって具体的なユーザ発話の方が抽象的なユーザ発話に比べてより正しい応答を生成しやすいという仮定に基づき，入力発話を具体的に言い換えた発話を考慮する対話応答生成モデルの構築と評価を行う．具体的には，最先端の対話応答生成モデルである UBAR [74] に対し，入力発話に加えてそれをより具体的に言い換えたものも考慮するようにモデルを改良する．

3.6.1　具体的な発話への言い換えを考慮した対話応答生成器の構築

UBAR は事前学習済みのテキスト生成モデルである GPT-2 [83] を基にして構築された End-to-End 型のタスク指向対話応答生成モデルであり，MultiWoZ データにおける応答生成において高い性能を記録している．UBAR のモデルアーキテク

表 3.10　対話応答生成の評価結果

モデル	入力発話	BLEU	INFORM	SUCCESS	COMBINED
UBAR	オリジナルのみ	15.07	90.6	77.8	99.27
UBAR w/ 具体的な発話	+ 具体的な発話（生成）	**15.39**	91.1	78.8	100.34
UBAR w/ 具体的な発話	+ 具体的な発話（参照文）	15.27	**91.7**	**79.4**	**100.82**

チャを図 3.10 (a) に示す．UBAR では GPT-2 に対して対話履歴中の最初のユーザ発話を入力し，その入力を基に，ユーザが何を求めていて何が解決されていないか等，現状の対話状態を予測する．予測された対話状態を基にホテルやレストランの空き情報等が登録されたデータベースを検索し，条件に合致したレコードを GPT-2 に入力する（図中「DB」）．その後，次にどのような応答を生成するべきかを示す対話行為を推定し，最後にシステム応答を生成する．次のユーザ発話が入力された後も同様の処理を繰り返すことで，対話履歴を考慮しながら応答を生成していくようなモデルとなっている．

本実験では UBAR への入力ユーザ発話に対して，それと同じ意図を持つ具体的な発話を図 3.10 (b) のように連結して入力することで，入力発話を事前に具体的に言い換えた場合の応答生成性能を検証する．なお，予測対象の応答の直前のユーザ発話のみを言い換え対象としている．訓練時には具体的な発話として参照文（DIRECT コーパスに収録されている具体的な発話）を用いる．評価時には，第 3.5 節で構築した具体的な発話への言い換えモデルを用いて入力発話を具体的に言い換えたものを用いる．また，言い換えによる性能向上の上限を調査するために，参考として評価時に参照文を用いた場合の性能についても報告する．

訓練設定やパラメータについては全てのモデルについて Yang ら [74] での設定に準拠した．実装は著者らが公開しているスクリプト[14)]を用いた．MultiWoZ の訓練データと評価データへの分割方法も 3.5 節や Yang ら [74] の設定と同様である．

3.6.2　評価尺度

評価尺度は BLEU, INFORM, SUCCESS, COMBINED の 4 種類を用いる．このうち INFORM, SUCCESS, COMBINED は MultiWoZ [47] 等のタスク指向対話データセットを用いた応答生成実験において代表的に利用されている評価尺度で

14）`https://github.com/TonyNemo/UBAR-MultiWOZ`

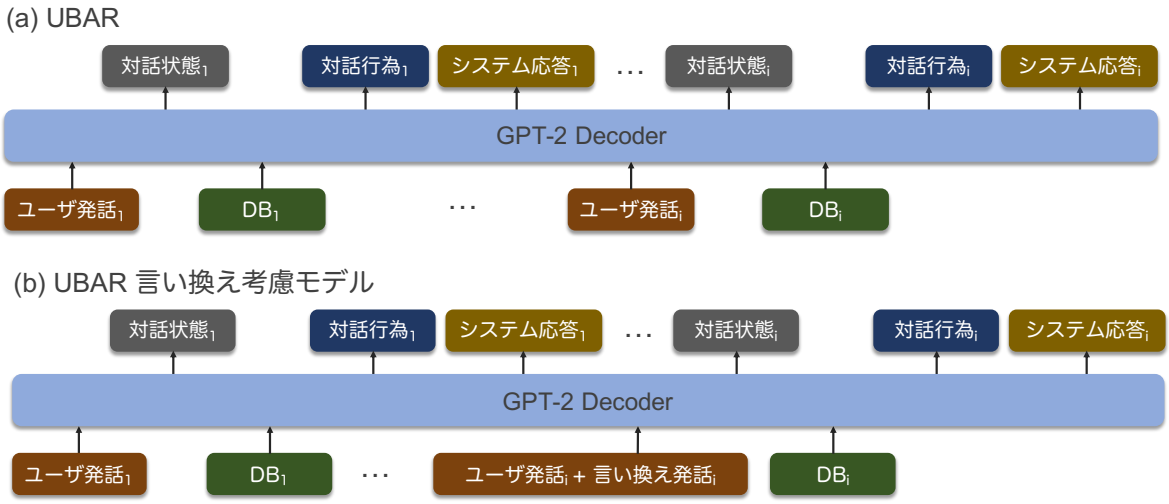

図 3.10　対話応答生成モデルのモデルアーキテクチャ

ある．INFORM は対話全体を通して提示したエンティティ（ホテル名やレストラン名など）が正解と一致した割合であり，ユーザの希望に合致した場所を案内できたかどうかを図る尺度である．SUCCESS は要求された属性値（住所や予約番号，列車番号など）を正確に出力できた割合であり，予約等の成功率を図る尺度である．COMBINED はこれら 3 つの尺度を統合したもので，$\text{COMBINED} = \text{BLEU} + 0.5 \cdot (\text{INFORM} + \text{SUCCESS})$ で計算される．

3.6.3　実験結果と考察

評価データにおける評価結果を表 3.10 に示す．表より，全ての尺度において具体的に言い換えたユーザ発話を考慮したモデルが UBAR のスコアを上回っていることがわかる．INFORM については言い換えモデルによって生成された具体的な発話を用いた場合で 0.5 ポイント，参照文を用いた場合で 1.1 ポイント上昇している．発話を具体的に言い換えたことで，ホテルやレストランに対するユーザの希望をより正確に汲み取り，希望に合致した場所を提示できるようになったことが示唆される．また，SUCCESS においては生成文を用いた場合で 1.0 ポイント，参照文を用いた場合で 1.6 ポイント上昇している．発話をより具体的に言い換えたことで，モデルがユーザの要求をより正確に推定できるようになり，予約の成功率が向上したことが示唆される．以上の結果から，ユーザ発話の具体的な発話への言い換えは対話応答生成の性能向上に寄与することがわかった．また，生成文と参照文を

比較した場合，BLEU 以外の全ての尺度で参照文の場合の方が性能が高いことが読み取れる．このことから，より正確な言い換えモデルを構築することで，さらなる応答生成性能の向上が見込める．

3.7 おわりに

本研究では 71,498 対の抽象的な発話と具体的な発話の対を含む対話コーパスを構築した．また，抽象的な発話を具体的な発話に言い換えるタスクを設計し，最先端の事前学習済み言語モデルの抽象的な発話に対する処理能力を調査した．さらに，対話応答生成において具体的な発話への言い換えモデルを用いて入力発話をより具体的な発話に言い換えることで，応答生成の性能が向上することを確認した．今後はより性能の高い言い換えモデルの構築に取り組む．

本研究ではタスク指向対話システムを対象としてコーパスを構築したが，第 1 章でも述べたように，対話システムには雑談を目的としたものも多く存在する．雑談対話はタスク指向対話に比べて話題が幅広く発話も多様であるため，より解釈の難しい抽象的な発話が発生すると考えられる．近年の雑談対話システムは流暢な応答を生成する一方でしばしば対話として成立しないような応答を生成してしまう問題がある [6]．タスク指向の場合と同様，この問題は抽象的な発話を具体的に言い換えることである程度改善可能であると期待される．雑談対話における抽象的な発話に着目したコーパスの構築も将来的な課題として挙げられる．

第 4 章　応答の具体性を制御可能な対話応答生成モデルの構築

4.1　はじめに

機械翻訳分野において広く用いられてきた Sequence to Sequence (seq2seq) [3] 型のモデルを応答生成に適用した対話応答生成モデル [2] が提案されている．seq2seq は入力文と出力文の対応関係を直接的に学習するモデルであるため，学習データとして大量の発話応答対を用意すれば比較的流暢な応答文を生成することができる．しかし，seq2seq は機械翻訳のように入力文と意味的に類似した文を出力するタスクでは高精度な文生成ができるが，雑談応答のように入力文に対して可能な応答が多様なタスクにおいては「そうですね」や「わかる」などといった，可能な中で最も無難で高頻度な応答（dull response）を生成してしまう問題を抱えている [39]．これは，seq2seq が入力発話文に対する応答として参照文が生成される確率を最大化するように訓練されるためであると考えられる [84]．

この問題を解決すべく，応答の具体性を向上させるような手法がこれまでに多く提案されている [39, 85, 86]．しかし，実際の人間同士の対話においては図 4.1 のように，ある発話に対して取りうる応答として，具体性の低い応答から高い応答まで様々な応答が存在する [7]．つまり，人間は応答の具体性を必要に応じて使い分けていると言える．そのため，人間と自然な対話を行う対話システムの構築のためには，応答の具体性を制御できるような手法が不可欠である．しかし，従来の研究のように単純に応答の具体性を向上させる手法では，具体性を使い分けることができない．

そこで本研究では，応答の具体性を制御可能な応答生成手法を提案する．発話中の単語と共起しやすい単語が含まれる応答ほど具体性が高いという直感に基づき，以下のように対話応答生成モデルを改良する．まず，単語間の正の自己相互情報量（Positive Point-wise Mutual Information; PPMI）を用いて，ある入力発話文に対して共起しやすい単語が出力されやすくなるように対話応答生成モデルのデコー

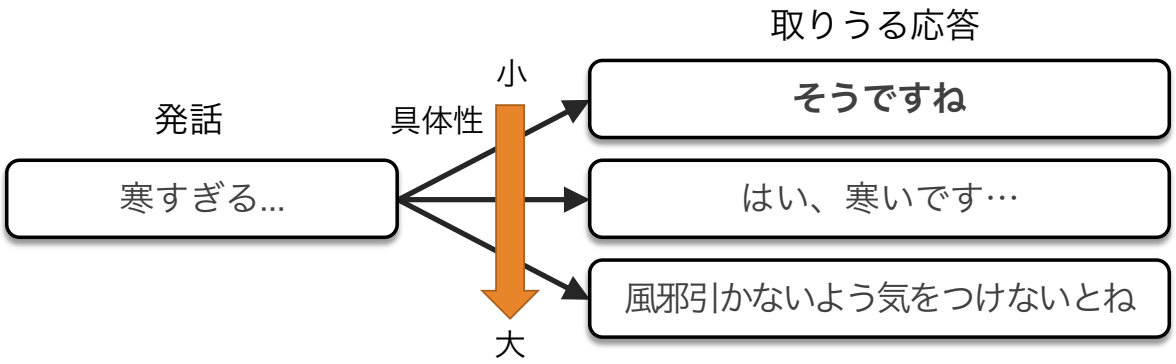

図 4.1　発話と応答の関係性の例．ある発話に対して取りうる応答には具体性の異なるいくつかの候補がある．

ダを改良する．また，PPMI に基づいて発話文と応答文の共起を測る MaxPMI という尺度を設計し，これを用いて学習データにスコアを自動的に付与することで Distant Supervision を導入する．推論時には任意のスコアを入力することで生成文の具体性を制御できるようになる．また，MaxPMI の上限値は入力文によって一意に定まるため，この値を入力すれば入力発話に対して具体性が最大となる文を出力することも可能となる．

Twitter から抽出した日本語対話データと DailyDialog と呼ばれる英語対話データを用いて実験を行い，自動評価と人で評価を行った．自動評価と人手評価の両方において，提案手法は既存手法よりも鋭敏に具体性を制御可能であることを確認した．

4.2　関連研究

対話応答生成モデルにおける dull response の抑制を図った研究はこれまで盛んに行われてきた．MMI [39] では，発話に対する相互情報量が高い応答を返すために，発話文 X と生成文 Y の間の PMI を $(1-\lambda)\log p(\mathbf{Y}|\mathbf{X}) + \lambda \log p(\mathbf{X}|\mathbf{Y})$ で近似し，生成時にこれを用いてリランキングを行っている．しかし，この手法は訓練時に相互情報量を最大化しないため，$p(\mathbf{Y}|\mathbf{X})$ による N-best 生成において妥当性の高い語が出力されなかった場合には効果が期待できない．dull response に対してペナルティを与えるように目的関数を設定してモデルを訓練する手法も複数提案されている．代表的な手法として，深層強化学習を用いた手法 [87, 88] や敵対的生成

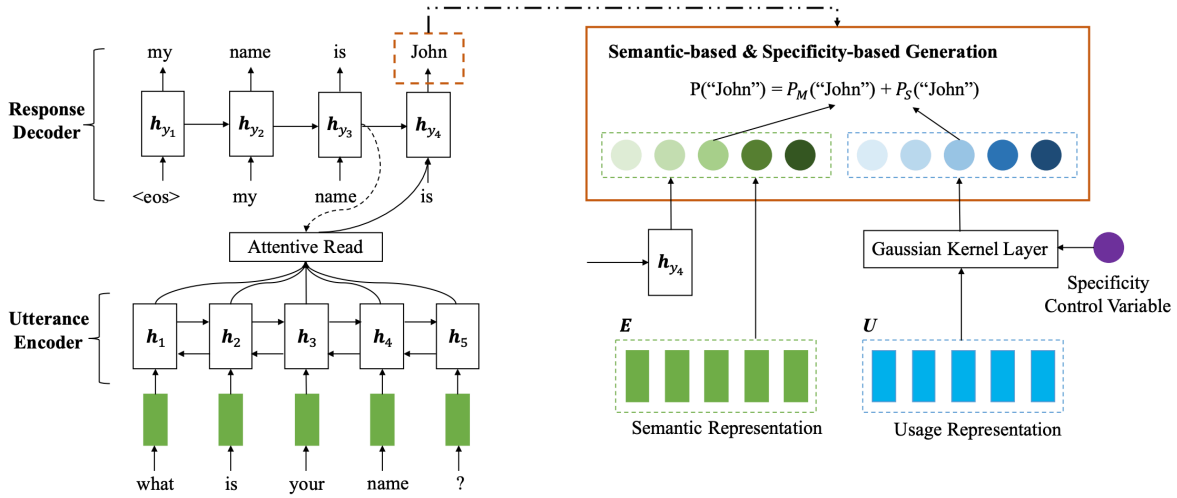

図 4.2　先行研究 SC-Seq2Seq のモデル概要図（Zhang ら [41] より引用）

ネットワークを導入した手法 [89, 85] などがあげられる．また，出現頻度の低い単語が多い応答ほど具体的であるという仮説に基づき，頻度の高さに応じて訓練時の損失関数を重みづけするような手法も提案されている [90, 86, 43, 91]．

応答の具体性の制御を目指した手法としては，Zhang ら [41] の SC-Seq2Seq があげられる．SC-Seq2Seq のモデル概要図を図 4.2 に示す．SC-Seq2Seq では学習データ中の応答文に文の具体性を示すスコアを自動で付与し Distant Supervision による学習を行うことで，出力文の具体性を制御できる．文の具体性の尺度としては，単語逆頻度が高い語を持つ文は具体性が高いという仮説から設計された Normalized Inverse Word Frequency (NIWF) を用いている．ある単語 y の逆頻度 IWF_y はその単語を含む応答文の総数 f_y と全データ数 $|R|$ を用いて以下で表される．

$$\mathrm{IWF}_y = \log(1 + |\mathcal{R}|)/f_y.$$

Zhang らはこれを用いて，応答文 $\mathbf{Y}$ の逆頻度 $\mathrm{IWF}_{\mathbf{Y}}$ を以下で定義している．

$$\mathrm{IWF}_{\mathbf{Y}} = \max_{y \in \mathbf{Y}} (\mathrm{IWF}_y)\,.$$

応答文 $\mathbf{Y}$ の NIWF スコアは $\mathrm{IWF}_{\mathbf{Y}}$ を $[0, 1]$ の範囲に正規化したものとしている．SC-Seq2Seq では NIWF を変数としたガウスカーネル層をデコーダの出力層に追加しており，推論時にはスコアを固定してデコードすることで具体性の制御が可能である．しかし，NIWF は $\mathbf{Y}$ のみによって決定する変数であることや，ガウスカー

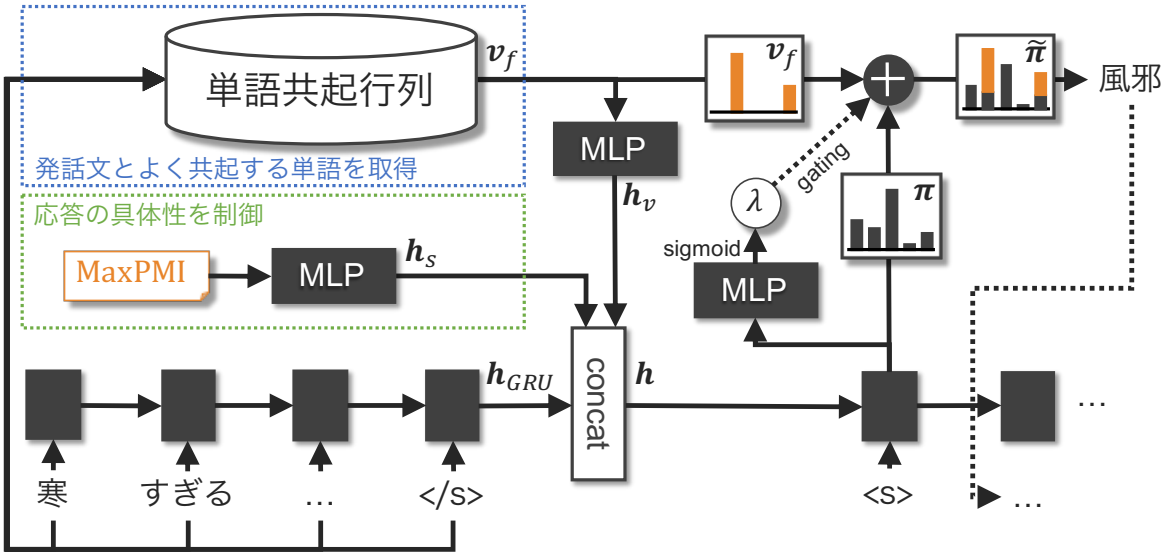

図 4.3　モデルの構造図

ネル層が入力になんら依存しない層となっていることから，具体的ではあっても発話に対する関連性が高いとは言えない応答がしばしば生成される．応答文の制御を目指した手法としては他に文献 [92] や文献 [42] が存在する．これらは対話行為や応答文の長さや具体性など様々な属性に基づいて応答を制御できる．しかし，属性を示すラベルを訓練データに追加でアノテーションする必要がある．

4.3　提案手法

提案手法の概要を図 4.3 に示す．提案手法ではまず PPMI に基づく共起尺度 MaxPMI を用いて入力発話文と応答文の共起度合いを示すラベルを自動でアノテーションする（第 4.3.1 節）．モデルは事前に計算した PPMI と MaxPMI に基づいて文生成を行う（第 4.3.2 節）．学習時は Distant Supervision の枠組みを利用し，発話応答対と事前付与したラベルを基に訓練を行う（第 4.3.3 節）．推論時には任意の固定値をスコアとして入力する方法と，スコアの上限値を推定して入力する方法によって文生成を行う（第 4.3.4 節）．

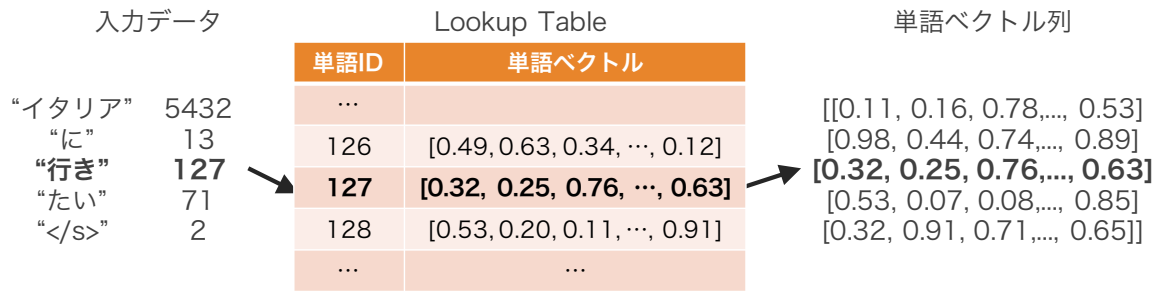

図 4.4　Embedding 層の概略図

4.3.1　発話文に対する応答文の共起度合いの推定

本研究では入力発話文に対する応答文の具体性の度合いはそれぞれに含まれる単語同士の共起の強さに相関するという直感に基づき，PPMI ベースの単純な共起尺度 MaxPMI を提案する．まず，訓練データ全体から単語同士の PPMI を事前に計算しておき，これを単語共起行列に保存する．単語 x が発話文・応答文中に現れる確率をそれぞれ $p_X(x)$ ，$p_Y(x)$ とし，単語 x, y がある発話応答対で同時に現れる確率を $p(x, y)$ としたとき，PPMI は以下で計算される．

$$\mathrm{PPMI}(x, y) = \max\left(\log_2 \frac{p(x, y)}{p_X(x) \cdot p_Y(y)}, 0\right)$$

ある発話文の単語列を $X = \{x_1, x_2, \ldots, x_{|\mathbf{X}|}\}$，それに対する応答文の単語列を $Y = \{y_1, y_2, \ldots, y_{|\mathbf{Y}|}\}$ としたとき，MaxPMI を以下で定義する．

$$\mathrm{MaxPMI}(\mathbf{X}, \mathbf{Y}) = \max_{x \in \mathbf{X}, y \in \mathbf{Y}} \mathrm{PPMI}(x, y)$$

ただし，モデルの学習時にはこれを $[0, 1]$ の範囲に正規化したものを用いる．

4.3.2　モデル構造

提案モデルは seq2seq のアーキテクチャを改良したものであり，エンコーダとデコーダを持つ．

エンコーダ　まず，入力データは単語に対応した one-hot ベクトルが並べられた時系列データ $X = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ として与えられる．Embedding 層はそれぞれの単語に該当する単語ベクトルを Lookup Table から抽出する（図 4.4）ことで $\mathbf{x_i}$ を分散表現 $\bar{\mathbf{x}}_\mathbf{i}$ に変換する．これは単語を表す one-hot ベクトルを Embedding 層の重

み行列 W_e を用いて線形変換する操作に等しい：

$$\bar{\mathbf{x}}_\mathbf{i} = W_e^{Encoder}\mathbf{x}_\mathbf{i}$$

続いて，RNN 層は Embedding 層の出力であるベクトル列 $\bar{X} = \{\bar{\mathbf{x}}_\mathbf{1}, \bar{\mathbf{x}}_\mathbf{2}, ..., \bar{\mathbf{x}}_\mathbf{n}\}$ を入力として受け取る．ここで，RNN 層の各セルには Gated Recurrent Unit (GRU) [4] を用いる．GRU は LSTM と同様，RNN の勾配消失問題の解決を目指したアーキテクチャである．GRU は，あるタイムステップ i における入力を $\bar{\mathbf{x}}_\mathbf{i}$，隠れ状態を $\mathbf{h}_\mathbf{i}$ としたとき，以下の計算を行う．

$$\begin{aligned}
\mathbf{z}_\mathbf{i} &= \text{sigmoid}\left(W_z \cdot [\mathbf{h}_\mathbf{i-1}, \bar{\mathbf{x}}_\mathbf{i}]\right) \\
\mathbf{r}_\mathbf{i} &= \text{sigmoid}\left(W_r \cdot [\mathbf{h}_\mathbf{i-1}, \bar{\mathbf{x}}_\mathbf{i}]\right) \\
\tilde{\mathbf{h}}_\mathbf{i} &= \tanh\left(W \cdot [\mathbf{r}_\mathbf{i} \odot \mathbf{h}_\mathbf{i-1}, \bar{\mathbf{x}}_\mathbf{i}]\right) \\
\mathbf{h}_\mathbf{i} &= (\mathbf{1} - \mathbf{z}_\mathbf{i}) \odot \mathbf{h}_\mathbf{i-1} + \mathbf{z}_\mathbf{i} \odot \tilde{\mathbf{h}}_\mathbf{i}
\end{aligned}$$

ここで，エンコーダの GRU の初期状態 $\mathbf{h_0}$ は $\mathbf{0}$ ベクトルである．また，W_z, W_r, W は学習可能なパラメータであり，$\text{sigmoid}(x)$ はシグモイド関数と呼ばれる $(-\infty, \infty) \to (0, 1)$ の活性化関数で，以下で定義される．

$$\text{sigmoid}(x) = \frac{1}{1 + \exp^{-x}}$$

$\mathbf{r}_\mathbf{i}$ はリセットゲートと呼ばれ，現在のタイムステップの仮の隠れ状態 $\tilde{\mathbf{h}}_\mathbf{i}$ の計算の際に一つ前のタイムステップの隠れ状態 $\mathbf{h}_\mathbf{i-1}$ をどれだけ破棄するかを決めるものである．一方 $\mathbf{z}_\mathbf{i}$ は更新ゲートと呼ばれ，$\mathbf{h}_(\mathbf{i}-\mathbf{1})$ と $\tilde{\mathbf{h}}_\mathbf{i}$ をどれだけの割合で足し合わせるか，つまり，過去の状態をどれだけ取り込むかを決めるためのゲートとして機能する．以上の操作を行い $\bar{\mathbf{x}}_\mathbf{i}$ を用いて内部状態 $\mathbf{h}_\mathbf{i}$ を得る操作を以下で記述する．

$$\mathbf{h}_\mathbf{i} = \text{GRU}^{\text{Encoder}}(\bar{\mathbf{x}}_\mathbf{i}, \mathbf{h}_\mathbf{i-1})$$

この操作を文頭から文末まで再帰的に適用し，得られた最終状態を発話文ベクトル $\mathbf{h}_{GRU}$ とする．

提案手法ではこれに加え，入力された $\text{MaxPMI}(\mathbf{X}, \mathbf{Y})$ を多層パーセプトロン（Multi-layer perceptron; MLP）によって処理する **スコアエンコーダ** を持つ．$\mathbf{h}_{GRU}$

とスコアエンコーダの出力 $\mathbf{h}_s$ を連結したベクトル $\mathbf{h}_e = \{\mathbf{h}_{GRU}; \mathbf{h}_s\}$ をエンコーダの出力とする．これにより，出力すべき文の具体性の目安をデコーダに与えることができる．

デコーダ　デコーダの持つ語彙集合を V としたとき，ある単語 $v \in V$ と入力文 X との単語共起スコア s_v を以下で定義する．

$$s_v = \sum_{x \in X} \mathrm{PPMI}\,(x, v)$$

デコーダはまず，全ての語彙に対する単語共起スコアからなるベクトル $\mathbf{v}_f \in \mathbb{R}^{|V|}$ を受け取り，これを多層パーセプトロン（Multi-layer perceptron; MLP）によって処理したベクトル $\mathbf{h}_v$ を得る．これとエンコーダからの出力ベクトル $\mathbf{h}_e$ を連結したものをデコーダの初期状態 $\mathbf{h}_0 = \{\mathbf{h}_e; \mathbf{h}_v\}$ とする．これにより，デコーダは事前に入力と共起しやすい単語の情報を得ることができる．デコーダは Embedding 層と GRU 層と Projection 層を持つ．Embedding 層は $i-1$ 番目のタイムステップで出力した単語の one-hot ベクトル $\mathbf{y_i}$ を受け取り，その単語の分散表現 $\bar{\mathbf{y}}_\mathbf{i}$ を返す．

$$\bar{\mathbf{y}}_\mathbf{i} = W_e^{Decoder} \mathbf{y_i}$$

RNN 層はその単語ベクトルと前回の隠れ状態 $\mathbf{h_{i-1}}$ を受け取り，次の隠れ状態 $\mathbf{h_i}$ を以下の計算によって返す：

$$\mathbf{h_i} = \mathrm{GRU}^{\mathrm{Decoder}}(\bar{\mathbf{y}}_\mathbf{i}, \mathbf{h_{i-1}})$$

Projection 層は $\mathbf{h_i}$ を受け取り，全結合層を用いて語彙数 N 次元分の実数値ベクトル $\tilde{\mathbf{y}}_\mathbf{i+1}$ に変換する：

$$\tilde{\mathbf{y}}_\mathbf{i+1} = W_p \mathbf{h_i} + \mathbf{b}_p$$

ただし，W_p は学習可能な重み行列で，$\mathbf{b}_p$ は学習可能なバイアス項である．ここで，$\tilde{\mathbf{y}}_\mathbf{i+1}$ の各要素に対して softmax 関数（式 4.1）を適用することで，各語彙の生成確率が計算できる．

$$\mathrm{softmax}(\mathbf{y}) = \frac{\exp(\mathbf{y})}{\sum_{y \in \mathbf{y}} \exp y} \tag{4.1}$$

また，各タイムステップ i におけるデコーダの出力ベクトル $\pi_\mathbf{i}$ に対して $\mathbf{v_f}$ を重み付きで足すことで，入力文との相互情報量の高い単語の出力確率を増幅させる

（**出力増幅器**）．最終的なデコーダの出力 $\tilde{\pi}_{\mathbf{i}}$ は以下の式で表される：

$$\tilde{\pi}_{\mathbf{i}} = (1 - \lambda_i) \cdot \pi_{\mathbf{i}} + \lambda_i \cdot \mathbf{v}_{\mathbf{f}}.$$

ここで，λ_i は，デコーダの現在の中間状態 $\mathbf{h}_i$ に応じて以下のように計算される：

$$\lambda_i = \text{sigmoid}\left(W^{gate}\mathbf{h}_i + \mathbf{b}^{gate}\right).$$

ただし，W^{gate} は学習可能な重み行列であり，$\mathbf{b}^{gate}$ はバイアス項である．

4.3.3 Distant Supervision

学習時には発話応答対 $(\mathbf{X}, \mathbf{Y})$ に対して Distant Supervision を行うためのラベルとして MaxPMI$(\mathbf{X}, \mathbf{Y})$ を計算し，これをモデルに入力することで学習を行う．なお，損失関数としては交差エントロピー損失を用いる．よって，モデルのパラメータを θ とすれば，損失関数は以下で表される．ただし，$\mathcal{D}$ は学習データセットである．

$$\mathcal{L} = \sum_{(\mathbf{X},\mathbf{Y})\in\mathcal{D}} \log P(\mathbf{Y}|\mathbf{X}, \text{MaxPMI}(\mathbf{X}, \mathbf{Y}); \theta).$$

直感的には，この損失関数により，発話と MaxPMI を条件としたときに，どのような応答を生成すべきかを学習することができるようになる．

4.3.4 推論

MaxPMI$(\mathbf{X}, \mathbf{Y})$ は $\mathbf{Y}$ が確定していない推論時には定義できない．そのため，$[0, 1]$ の範囲の実数値 s を入力して文を生成する（**固定値デコード**）．ここで，MaxPMI の性質上 s が 1 に近ければ近いほど共起の高い語が生成されやすくなり，具体性の高い文が出力されると期待できる．ただし，実際には「こんにちは」などの定型的な挨拶のように，具体性の低い応答しか取り得ないような発話も存在しうる．このことから，ある発話に対して取りうる s の値には上限があると考えられる．そこで，入力文 $\mathbf{X}$ に対する MaxPMI の推定上限値 s_{max} を提案する．s_{max} は事前計算した PPMI によって以下のように決定する：

$$s_{max} = \max_{x\in\mathbf{X}, v\in\mathbf{V}} \text{PPMI}(x, v).$$

s_{max} を用いてデコードを行うことで $\mathbf{X}$ に対してあり得る応答の中で具体性が最も大きい応答を生成できると考えられる（**具体性最大化デコード**）.

4.4 実験設定

提案手法が発話との関連性と保ちながら応答の具体性を制御できているかどうかを評価するために，日本語と英語の雑談対話コーパスを用いて応答生成の実験を行う．本節ではその実験設定について説明する.

4.4.1 使用するコーパス

実験においては，日本語対話コーパスである Twitter と英語対話コーパスである DailyDialog を用いる．それぞれの詳細を以下に記載する.

Twitter SNS の一つである Twitter[1] から日本語の対話をクロールしたものである．具体的には，あるツイートとそのリプライの対をそれぞれ発話と応答の対と見立てることで対話コーパスを構築した．訓練/検証/評価データのサイズはそれぞれ $1,383,424/24,123/25,010$ 発話応答対となっている．それぞれの発話応答対は transformers[2] (version = 2.5.1) というライブラリに付属する BertJapaneseTokenizer (bert-base-japanese) を用いてサブワード単位に分割した.

DailyDialog 文献 [46] にて構築された英語対話コーパスである．様々なウェブサイトから人間同士の日常会話をクロールして構築されたものである．このコーパスは 2 人の人間が交互に発話を行うマルチターン形式の対話コーパスとなっているが，本研究においては連続する 2 つの発話ペアを発話応答対と見做して使用する．訓練/検証/評価データのサイズはそれぞれ $76,052/7,069/6,740$ 発話応答対となっている．それぞれの発話応答対は BertTokenizer (BERT-base-uncased) を用いてサブワード単位に分割した.

また，これらのコーパスそれぞれに対し，頻度が 50 以上のサブワードについて発話・応答間での PPMI を事前計算し，単語共起行列を作成した.

1） https://twitter.com
2） https://github.com/huggingface/transformers/

4.4.2 比較手法

実験においては提案手法と既存手法の比較を行う．まずベースラインとして，通常の seq2seq を使用する．seq2seq のモデル構造は，提案手法からスコアエンコーダと出力増幅器を取り除いたものと一致する．

また，具体性の制御を目的とした提案手法に最も関連する手法として，**SC-Seq2Seq** [41] との比較も行う．SC-Seq2Seq は，distant supervision を用いて出力文の具体性の制御を可能にした手法である．SC-Seq2Seq では，文中の単語の頻度が低いほど文の具体性が高くなるという仮説に従い，文の具体性の指標として単語の逆頻度を用いている．さらに，SC-Seq2Seq は，デコーダの出力層に加えて，ガウスカーネル層に基づいた単語予測機構を備える．この単語予測機構は，発話と応答の共起を考慮する我々の手法とは異なり，単語の希少性のみを考慮したものである．推論時には，$\in [0, 1]$ の具体性スコア入力することで応答の具体性を制御する．

4.4.3 自動評価尺度

システムの特性を定量的に評価するために，テストデータでの各システムの出力文に対し，複数の自動評価尺度を用いて評価を行う．本研究では BLEU, NIST, distinct, ent, の四つの評価尺度を用いる．以下ではそれぞれの評価尺度の概要と選定理由について述べる．

BLEU　BLEU [35] は機械翻訳において用いられる最も代表的な自動評価尺度であり，生成文と参照文の n-gram の一致度を測る尺度となっている．詳細は 2.2 節を参照されたい． 2.2 節でも述べたように，対話においては BLEU と人手評価の間には中程度の相関があることが知られている．そのため本研究においても BLEU を用いた評価を行うこととする．

NIST　NIST [38] とは，機械翻訳において用いられる自動評価尺度である．基本的には BLEU と同様に生成文と参照文の n-gram の一致度を測るものだが，その計算式は以下で定義され，BLEU とは異なるものとなっている．

$$\mathrm{NIST} = \mathrm{BP}_{\mathrm{NIST}} \sum_{n=1}^{N} \frac{\sum_i \left(\sum_{\substack{\text{生成文 } i \text{ と参照文 } i \text{ に} \\ \text{共通する単語 } w_1, w_2, ..., w_n}} \mathrm{Info}_i(w_1, w_2, ..., w_n) \right)}{\sum_i \text{翻訳文 } i \text{ 中の全 } n\text{-gram 数}})$$

ただし，

$$\mathrm{Info}(w_1, w_2, ..., w_n) = \log_2 \frac{\text{評価コーパスにおける } w_1, w_2, ..., w_{n-1} \text{ の数}}{\text{評価コーパスにおける } w_1, w_2, ..., w_n \text{ の数}}$$

である．NIST は 0 以上の実数値をとり，値が高いほど良いシステムであると評価されるが，BLEU とは異なり，$\mathrm{Info}(w_1, w_2, ..., w_n)$ によって各 n-gram に情報量に応じた重み付けを行う．$\mathrm{Info}(w_1, w_2, ..., w_n)$ は，ある $(n-1)$-gram $\{w_1, w_2, ..., w_{n-1}\}$ の次に続く単語のバリエーションが評価コーパス中で多ければ多いほど高くなるという特徴を持つ．そのため，同じ bi-gram であっても “is the” や “is a” のような機能語列より “the cat” や “the dog” のように内容語を含んだ単語列の一致の方がより高く評価される．BP_{NIST} は BP_{BLEU} と同様，生成文の単語数 c が参照文の単語数 r よりも短い場合にペナルティを与える項であるが，定義は BP_{BLEU} とはやや異なり，計算式は以下で表される．

$$\mathrm{BP}_{\mathrm{NIST}} = \exp\left\{\beta \log^2 \left[\min\left(\frac{c}{r}, 1\right)\right]\right\}$$

ただし，β は，$c/r = 2/3$ のときに BP_{NIST} が 0.5 となるように設定する．

$$\begin{aligned} 0.5 &= \exp\left\{\beta \log^2 \frac{2}{3}\right\} \\ \log 0.5 &= \beta \log^2 \frac{2}{3} \end{aligned}$$

よって，β は以下のようになる．

$$\beta = \frac{\log 0.5}{\log^2 \frac{2}{3}}$$

NIST は BLEU と同じように生成文と参照文の n-gram 一致度を測る指標であるため，BLEU と同様に人手評価との相関がある程度あると考えられる．また，NIST は情報量の高い n-gram を重視するため，汎用的な応答文よりも具体性の高い応答文の方が評価されると考えられる．そのため本研究においては具体性が高くかつ妥当な応答生成がなされていることを確認するための自動評価尺度として NIST を採用する．

dist　dist とは，Li ら [39] によって提案された，システム応答の語彙の多様性を評価するための尺度であり，生成された応答の n-gram のばらつき度合いを表すような指標である．計算式は以下で定義される．

$$\text{distinct}-n = \frac{\text{生成された } n\text{-gram の種類数}}{\text{生成された } n\text{-gram の総数}}$$

用いられた全ての n-gram の種類数が多ければ多いほど，また各 n-gram が用いられた頻度が小さければ小さいほど高いスコアとなる．

dist は参照文との比較による評価尺度ではなく，また生成された n-gram を全て平等に扱うため，例えば発話文に対して全く関係のない単語を生成した場合でも，多くの種類の単語を用いていれば高い評価となってしまう．そのため，通常は BLEU など参照文との一致度を測る評価尺度と共に用いられる．本研究でも応答の語彙の多様性を測るための尺度として distinct を用いるが，BLEU や NIST との関係を考慮しながら考察を行うものとする．

ent　ent [85] は dist と同様，システム応答の語彙の多様性を測る尺度の一つである．dist とは異なり，生成された n-gram の頻度を考慮するのが特徴である．ある生成文の ent は以下の式で計算できる：

$$\text{ent} = -\frac{1}{\sum_w F(w)} \sum_{w \in Y} F(w) \log \frac{F(w)}{\sum_w F(w)}.$$

ここで，Y は生成文の n-gram の集合，$F(w)$ はある n-gram w の頻度である．

繰り返し率　seq2seq をはじめとする言語生成モデルにおいてはしばしば同じ単語やフレーズを必要以上に繰り返し生成してしまう現象が観測される [93, 94]．繰り返し率 [93] とは，生成文中において同じ単語が繰り返し用いられている度合いを示す尺度であり，以下で計算される：

$$\text{repetition\_rate} = \frac{1}{N} \sum_{i=1}^{N} \frac{1 + r\left(\widetilde{y}_i\right)}{1 + r(Y_i)}.$$

なお，$\widetilde{y}_i$ はテストデータ中 i 番目の生成文，Y_i は i 番目の参照文，N はテストデータの総数であり，

$$r(X) = \text{len}(X) - \text{len}(\text{set}(X))$$

である．

4.4.4　人手評価の設定

自動評価ではそれぞれの評価尺度の性質を基にシステム出力の特性を定量的に評価することはできるものの，システムが生成した応答文そのものの品質を正しく評価することが困難なため，人手での評価も同時に行う必要がある．本研究では評価者に対し，日本語 Twitter コーパスから抽出した 300 個の発話文とその発話文をもとに複数のシステムが生成した応答文を見せ，それぞれの生成文の品質を評価してもらう．なお，評価者の募集はクラウドソーシングプラットフォームである Lancers[3)] で行い，日本語話者でありかつツイッター利用者であるような 6 名を採用している．また，各システムが生成した応答文を事前にシャッフルした状態で評価者に見せることで，どれがどのシステムの生成文か見分けがつかないようにし，評価の公平性を担保する．

評価基準は先行研究である SC-Seq2Seq [41] と同じく，各システム応答に対して以下の 3 つのラベルのうちの 1 つを付与してもらうものとする．

具体的　発話に対する応答として成り立っており，文法的にも正しく，かつ具体的な応答である．

汎用的　発話に対する応答として成り立っており，文法的にも正しいが，「わかりません」などのような具体性の低い応答である．

破綻　発話に対する応答として成り立っていないか，あるいは文法的な誤りなどがあって文意が理解できない応答である．

最終的な評価値は，他の評価者とのラベル一致率が著しく低かった 1 名の評価者を除外した 5 名分のラベルの多数決によって決定する．

4.4.5　モデル設定

全ての比較手法のモデルについて，Adam [95] オプティマイザを用いて訓練を行った．また，学習率は 0.0002 とした．勾配爆発を防ぐために，閾値を 5 に設定して勾配クリッピングを行った．全てのモデルについて，GRU の隠れ状態の次元数を 512，Embedding の次元数を 256 に設定した．Twitter コーパスにおいては 40

3）　https://www.lancers.jp/mypage

表 4.1　日本語 Twitter コーパスにおける自動評価結果

	BLEU-2	NIST	dist-1	dist-2	ent-4	rep	length
提案手法 ($s = s_{max}$)	**4.22**	**0.66**	**0.063**	**0.19**	**8.47**	2.68	6.08
提案手法 ($s = 0.5$)	4.09	0.64	0.057	0.17	8.26	2.90	6.51
SC-Seq2Seq ($s = 0.8$)	4.00	0.62	0.010	0.02	5.65	1.90	5.45
seq2seq	3.53	0.41	0.008	0.02	4.00	**1.56**	4.08
参照文	100.00	16.85	0.110	0.51	11.17	1.00	6.11

表 4.2　英語 DailyDialog コーパスにおける自動評価結果

	BLEU-2	NIST	dist-1	dist-2	ent-4	rep	length
提案手法 ($s = s_{max}$)	**17.62**	**2.87**	0.083	**0.41**	**10.77**	1.46	11.89
提案手法 ($s = 0.5$)	17.41	2.85	0.085	0.41	10.74	1.41	11.63
SC-Seq2Seq ($s = 0.5$)	8.18	1.40	**0.098**	0.36	10.34	1.29	10.09
seq2seq	9.00	1.54	0.096	0.37	10.31	**1.26**	9.70
参照文	100.00	16.70	0.127	0.54	10.91	1.00	11.67

エポックまで，DailyDialog においては 200 エポックまで訓練を行い，そのうち検証データにおける BLEU が最も高くなったモデルを評価に用いた．

SC-Seq2Seq はガウスカーネル層における分散を決定するためのハイパーパラメータ σ^2 を持つ．Twitter コーパスにおいては σ^2 を 0.1 に，DailyDialog においては 0.2 に設定した．0.1, 0.2, 0.5, 1.0 の候補の中から，検証データにおける BLEU が最も高くなるような σ^2 を選んだ．

モデルは全て深層学習フレームワークである PyTorch[4] [96] (version = 1.0.0) を用いて実装した．訓練・評価の両方において，単一 GPU[5] 環境にて実験を行った．

4.5　実験結果と考察

4）`https://pytorch.org/`

5）NVIDIA Tesla V100 SXM2, 32 GB memory

4.5.1 自動評価結果

評価データにおける自動評価の結果を表 4.1 (Twitter) と表 4.2 (DailyDialog) に示す．なお，表における “length” は応答の平均サブワード数を示す．提案手法の具体性最大化デコード（$s = s_{max}$）を用いたものが BLEU, NIST, dist, ent のすべてにおいて最も高い性能を達成していることがわかる．これらの結果から，具体性最大化デコードを用いることで，ある発話に対して最適な具体性スコア s を推定できることがわかる．また，DailyDialog において，提案手法は比較手法と比べて BLEU と NIST を大きく向上していることがわかる．これは，提案手法では単語の共起情報をモデルに明示的に与えるため，規模の小さいコーパスにおいては seq2seq の学習を補完する効果が大きいのではないかと考えられる．

SC-Seq2Seq は Twitter コーパスにおいては提案手法と同等の BLEU, NIST 値を示したが，dist と ent は seq2seq 以上に低い結果となった．一方，DailyDialog コーパスにおいては SC-Seq2Seq は高い dist, ent 値を示しており，また BLEU, NIST に関しては seq2seq よりも低いという結果になっている．これらの結果から SC-Seq2Seq の有効性はドメインに依存することが示唆される．これは，SC-Seq2Seq では発話と応答の関係を考慮せず，単語の頻度に基づいて具体性を推定していることから，発話との関連性が低いような低頻度語を出力しやすいためであると推測できる．

提案手法の弊害として，両方のコーパスにおいて単語の繰り返し率（表中 “rep”）が高いことが挙げられる．提案手法の繰り返し率が大きい原因として，デコーダの状態は各タイムステップにおいて変化するのに対して各単語の PPMI 値は一定であるため，PPMI が高い単語が常に生成されやすくなってしまっているためだと考えられる．デコーダの状態や既に生成した単語に応じて PPMI 値を割り引くような機構を追加することでこの問題を軽減できると考えられる．

4.5.2 応答の制御可能性の検証

応答の具体性をどの程度鋭敏に制御できるかを自動評価尺度を用いて検証した．検証データのを用いて，SC-Seq2Seq と提案モデルのそれぞれについて，s の値を $[0.0, 0.2, 0.5, 0.8, 1.0, s_{max}]$ のいずれかに設定して応答を生成させた．

結果を表 4.3 (Twitter), 表 4.4 (DailyDialog) に示す．提案手法では SC-Seq2Seq に

表 4.3　日本語 Twitter コーパスにおける具体性の制御可能性の評価結果

		BLEU-2	NIST	dist-1	dist-2	ent-4	rep	length
提案手法	$s = 0.0$	0.03	0.00	0.007	0.03	2.39	0.94	1.28
	$s = 0.2$	2.96	0.36	0.035	0.10	6.39	1.81	4.09
	$s = 0.5$	4.26	0.68	0.058	0.17	8.15	2.93	6.54
	$s = 0.8$	3.45	0.56	0.046	0.15	8.12	3.97	8.25
	$s = 1.0$	3.23	0.53	0.039	0.13	8.09	4.23	8.72
	$s = s_{max}$	**4.46**	**0.70**	**0.064**	**0.19**	**8.41**	2.68	6.06
SC-Seq2Seq	$s = 0.0$	2.68	0.13	0.013	0.04	6.06	0.98	2.99
	$s = 0.2$	2.88	0.17	0.013	0.04	6.01	0.99	3.11
	$s = 0.5$	3.65	0.40	0.013	0.03	5.48	1.42	3.89
	$s = 0.8$	4.16	0.66	0.011	0.03	5.71	1.82	5.36
	$s = 1.0$	3.86	0.57	0.009	0.02	5.66	2.85	6.36

表 4.4　英語 DailyDialog コーパスにおける具体性の制御可能性の評価結果

		BLEU-2	NIST	dist-1	dist-2	ent-4	rep	length
提案手法	$s = 0.0$	1.83	0.02	0.057	0.25	8.26	0.85	3.97
	$s = 0.2$	5.70	0.39	0.073	0.33	9.44	0.93	5.97
	$s = 0.5$	15.66	2.54	0.078	**0.39**	10.73	1.40	11.63
	$s = 0.8$	11.97	1.95	0.068	0.38	11.06	1.89	14.56
	$s = 1.0$	10.53	1.72	0.065	0.37	**11.16**	2.13	15.83
	$s = s_{max}$	**16.04**	**2.60**	0.076	**0.39**	10.76	1.45	11.77
SC-Seq2Seq	$s = 0.0$	7.49	1.18	0.089	0.35	10.13	1.13	8.58
	$s = 0.2$	7.75	1.28	0.090	0.35	10.22	1.17	9.08
	$s = 0.5$	7.92	1.35	0.095	0.35	10.32	1.25	9.68
	$s = 0.8$	7.80	1.32	0.102	0.36	10.30	1.23	9.48
	$s = 1.0$	7.60	1.26	**0.105**	0.37	10.26	1.19	9.11

比べて s の変化に対して鋭敏に評価値が変動していることがわかる．特に，$s \leq 0.5$ の範囲では s の増加に合わせて全ての評価値が増加している．一方，$s \geq 0.5$ の範囲では逆に s の増加に合わせて全てのスコアが減少している．これらの傾向から，

表 4.5　日本語 Twitter コーパスにおける人手評価結果

Models	割合 (%)			Kappa
	具体的	汎用的	破綻	
提案手法 ($s = s_{max}$)	24.8	19.8	55.4	0.42
提案手法 ($s = 0.5$)	**26.6**	17.9	55.5	0.41
提案手法 ($s = 0.0$)	0.6	53.8	45.6	0.02
Seq2Seq	10.1	59.7	30.3	0.42
SC-Seq2Seq ($s = 1.0$)	11.5	33.7	54.8	0.56
SC-Seq2Seq ($s = 0.8$)	9.8	54.7	35.5	0.50
SC-Seq2Seq ($s = 0.0$)	10.9	56.5	32.6	0.44
提案手法 (hybrid)	22.0	38.2	39.8	–
SC-Seq2Seq (hybrid)	12.1	50.2	37.7	–

入力された具体性スコアが発話に対して取りうる範囲を超えている場合に，提案手法は適切な応答を生成できないことが示される．また，s を大きくすることで繰り返し率も増加していることがわかる．これは，デコーダは s が大きいほど発話に共起する単語を優先するため，そのような単語が繰り返し生成されるようになってしまったものと考えられる．

提案手法に具体性最大化デコードを適用したもの（$s = s_{max}$）は，BLEU, NIST, dist, ent の全ての尺度において最も良い値を示している．また，繰り返し率に関しても $s = 0.5$ のときよりも低くなっている．この結果から，具体性最大化デコードによって，入力発話に対する最適な s を推定できていることが示唆される．DailyDialog コーパスにおいても同様の傾向が確認できる．

4.5.3　人手評価結果

日本語 Twitter コーパスの評価データにおける人手評価結果を表 4.5 に示す．まず，評価者間でのラベル一致度合いを示す Fleiss' Kappa [97] の値に関しては，$s = 0.0$ のときの提案手法を除いて 0.4 を超えている．これは先行研究である Zhang ら [41] において報告されている人手評価の Kappa 値と同様の値である．$s = 0.0$ のときの提案手法において Kappa 値が 0.02 と著しく低い理由として，こ

表 4.6　Twitter コーパスの評価データにおける応答生成例

発話	暇さえあれば猫動画観てる。他所の猫かわいく見える
提案手法 ($s = s_{max}$)	猫カフェの動画見てるだけで可愛い
提案手法 ($s = 1.0$)	猫可愛いですよね
提案手法 ($s = 0.8$)	猫カフェの動画見てるだけで可愛い
提案手法 ($s = 0.5$)	猫カフェの動画見てるだけで可愛い
提案手法 ($s = 0.2$)	可愛い
提案手法 ($s = 0.0$)	ありがとう
SC-Seq2Seq ($s = 1.0$)	それな
SC-Seq2Seq ($s = 0.8$)	私のことが気になるの?
SC-Seq2Seq ($s = 0.5$)	それな
SC-Seq2Seq ($s = 0.2$)	それな!
SC-Seq2Seq ($s = 0.0$)	それな!
seq2seq	可愛い
発話	でもあした祝日だからやばそう
提案手法 ($s = s_{max}$)	祝日なんだから仕方ない
提案手法 ($s = 1.0$)	祝日は関係ないからな
提案手法 ($s = 0.8$)	祝日は祝日だから仕方ないね
提案手法 ($s = 0.5$)	祝日だから仕方ない
提案手法 ($s = 0.2$)	がんばれがんばれ
提案手法 ($s = 0.0$)	ありがとう
SC-Seq2Seq ($s = 1.0$)	それは無理だわ
SC-Seq2Seq ($s = 0.8$)	俺は今からバイトだから
SC-Seq2Seq ($s = 0.5$)	それは無理だわ
SC-Seq2Seq ($s = 0.2$)	明日は仕事だよ
SC-Seq2Seq ($s = 0.0$)	おはよー!
seq2seq	それな

の設定では “?” や “は？” などの非常に短い応答が頻繁に生成される[6)]ため，発話

6)　$s = 0.0$ のときに提案手法が生成した応答の平均文長は 1.46 語であった

に対する応答として成り立っているかどうかの判断が割れてしまったためだと考えられる．

提案手法は $s = 0.5$ のときと $s = s_{max}$（具体性最大化デコード）のときにおいて $s = 0.0$ よりも “具体的” の割合が非常に大きい．このことから，提案手法は s の増加に応じて応答の具体性を向上できていることがわかる．また，SC-Seq2Seq と比較すると，s の変化に対する “+2” の割合の変化は提案手法の方が大きいことがわかる．このことから，提案手法は SC-Seq2Seq よりも鋭敏に具体性を制御できていることが示唆される．しかし，提案手法と SC-Seq2Seq のいずれにおいても，s を増加させることで “破綻” の割合も大きく増加してしまっていることがわかる．これは，特定の単語の生成確率を増幅させることでデコーダの言語生成能力に悪影響を与えてしまい，応答の流暢性が悪化してしまったためだと考えられる．特に，4.5.2 節で示された単語の繰り返しの増加は，応答の流暢性の悪化に大きく影響しているものと考えられる．

この課題に対処するため，提案手法の生成文と seq2seq の生成文を簡単なヒューリスティックに基づいて切り替えるような手法を提案する．具体的には，提案手法が生成した応答文に含まれるユニークな単語の割合がある閾値 T（今回は $T = 0.95$ とする）を下回った場合（つまり，提案手法の応答文が繰り返しを多く含む場合）に，その応答の代わりに seq2seq の応答を使用する．このヒューリスティックを提案手法（$s = s_{max}$）と SC-Seq2Seq（$s = 1.0$）に適用した結果を，それぞれ表 4.5 の “提案手法 (hybrid)” と “SC-Seq2Seq (hybrid)” に示す．元々の評価値と比べ，提案手法においても SC-Seq2Seq においても “具体的” の割合をほとんど減らさずに “破綻” の割合を 15% 以上減少させることができた．

4.5.4 事例分析

表 4.6 に Twitter コーパスの評価データにおける応答の生成例を 2 例示す．表を見ると，$s \geq 0.5$ の範囲において，提案手法は発話に対して具体的な応答を生成できていることが確認できる．しかし，s の値が非常に大きいとき（特に，2 例目における $s = 0.8$ の場合）において，提案手法は同じフレーズを繰り返し生成してしまっているのが見て取れる．4.5.1 節でも言及したとおり，これは発話との共起が大きい単語の生成確率を増幅したことによる弊害であると考えられる．一方で，具体性最大化デコード（$s = s_{max}$）では，発話ごとに最適な s の値が決定されること

により，この問題が解決されていることがわかる．

SC-Seq2Seq は 2 例目においては seq2seq に比べて具体的な応答を生成できている．しかし，1 例目 を見ると，s を増加させても応答がより具体的になるような傾向は確認できない．特に，1 例目の SC-Seq2Seq $s = 0.8$ は発話に対して成り立たないような応答になってしまっている．これは，SC-Seq2Seq では具体性の計算の際に発話と応答の関係を考慮していないためであると考えられる．これに対し，提案手法は “猫”，“映画”，“かわいい” など，発話によく関連する単語を生成できていることがわかる．

4.6 おわりに

発話と応答中の単語の間の共起情報を用いることで，先行手法と比べてより鋭敏に応答の具体性を制御できることを実験的に示した．また，先行手法では具体性の高い応答を生成させると発話との関連性が低い単語が多く生成されてしまっていたが，提案手法ではこれを解決できることも確認した．しかし，提案手法と先行手法の両方において，通常の seq2seq を用いた対話応答生成モデルよりも破綻した応答の割合が増加してしまった．対話応答生成モデルの実用化にあたっては，このような破綻した応答をユーザに提示する前に検出し，フィルタリングするような機構の実現が必要であると考えられる．

第 5 章　アノテータの多様性に頑健な対話破綻検出

5.1　はじめに

第 3 章でも議論したように，対話応答生成モデルはしばしば対話を破綻させるような，一貫性の低い応答を生成する．この現象は特に，応答の具体性の向上を図ったモデルにおいて顕著であり，具体性と一貫性にはトレードオフの関係があることが知られている [98, 99]．そのため，対話システムの実用にあたっては，対話システムが出力した一貫性の低い応答をユーザに提示する前に事前に検知してフィルタリングするような機構が重要となる．対話破綻検出 [100] タスクは，ユーザと対話システムの対話データの中から，対話破綻を引き起こすようなシステム応答を検出するタスクである．

対話破綻検出タスクのデータは，ユーザとシステムの対話のうち，各システムの応答に複数人のアノテータによって破綻の有無を示すラベルがアノテーションされている．ラベルは O (破綻していない), T (破綻の可能性がある), X (破綻している) の三種類からなる．対話破綻を感じる基準は人によって異なるため，しばしばアノテータごとにラベル付与の判断が割れる場合がある．実際，対話破綻検出の開発データ [100] において，アノテータ間のアノテーション一致率を示す Kappa 値は 0.14 とかなり低い．対話破綻検出の開発データから抽出した実際のデータ例を表 5.1 に示す．一番上の例ではユーザ発話に対して全く関係ないトピックのシステム応答となっており，アノテーションはほとんど割れず，“X” に集中している．しかし，他の 2 例については共にアノテーションが大きく割れていることがわかる．例えば 2 例目は発話が “動物園に行ったことがあるかどうか” を尋ねるものであるのに対し，応答はその問いに明示的には答えていない．この応答をもって暗示的に “いいえ” と答えていると捉えるかどうかによって判断が割れるものと考えられる．このように，対話破綻を感じる基準は人によって異なるため，アノテーションの傾向には多様性がある．

表 5.1　対話破綻検出タスクのデータにおける発話応答とその破綻ラベルの例

ユーザ発話	システム応答	アノテーション O	T	X
私も眠くなってきました	チキンライス！あれ？そんなメニューなかったなぁ。	0	1	29
よく動物園に行ったりするんですか？	水族館だね。	7	17	6
あだ名、教えてください	職場で同僚をあだ名で呼ぶのは、どうかと思うよ。	8	13	9

従来の手法 [101, 102] では多数決などで一つの値に集約したラベルをもとに学習を行なっており，アノテーション傾向の多様性を考慮していない．似たようなアノテーション傾向を持つアノテータのグループごとに，その傾向を再現するような検出器を構築することで，より高性能な対話破綻検出が実現できると考えられる．

そこで本章では，アノテーション傾向の多様性を考慮した対話破綻検出手法の構築を行う．提案手法ではまず，アノテータをそれぞれのアノテーション傾向に基づいて k-means を用いてクラスタリングする．次に，各クラスタごとにそのクラスタのアノテーション傾向に基づいて対話破綻検出器を構築する．最後に，すべてのクラスタの検出器をアンサンブルすることで，最終的な対話破綻確率を決定する．対話破綻検出器としては，2 つの LSTM エンコーダを直列に繋ぎそれぞれを用いてユーザ発話・システム応答をエンコードするモデルと，2 つの畳み込みニューラルネット（Convolutional Neural Networks; CNN）を用いたエンコーダを並列に繋ぎそれぞれを用いてユーザ発話・システム応答をエンコードするモデルを提案する．そして LSTM モデル，CNN モデルをアンサンブルしたモデルの合計 3 種類を用いる．

対話破綻検出タスクのコンペティションである DBDC3 [103] の日本語タスクにおいて，提案手法は条件付き確率場（Conditional Random Field; CRF）を用いたベースラインを F 値において 5.6%，Accuracy において 3.5% 上回る結果を得た．アノテーション傾向の偏りを考慮せずにクラスタリングした場合や，アンサンブルを行わなかった場合との比較も行い，アノテーション傾向を考慮したクラスタリン

グの有効性も確認した．また，第 4 章 にて構築した対話応答生成モデルの生成結果への適用実験も行った．訓練データとのドメインの異なりから全体的には低い検出性能となったが，この実験においても提案手法はベースラインを上回る検出性能を記録した．

5.2 関連研究

対話破綻検出の手法はこれまで多く研究されている．DBDC3 において最も高い検出性能を達成した Sugiyama ら [104] の手法では，決定木に基づくアンサンブル学習手法の一種である Extra Trees Regressor [105] を用いて検出器を構築している．この手法では深層学習を用いずに高い検出性能を達成しているが，対話破綻検出タスクのデータの他に特徴量抽出器の構築のための外部データを要求する．

近年では深層学習を用いた手法も多く提案されている．Inaba ら [101] は RNN を用いてユーザ発話とシステム応答をそれぞれエンコードし，それに基づいて破綻ラベルを分類するモデルを構築している．Kubo ら [102] は seq2seq モデルを用いて，ユーザ発話からシステム応答を予測するタスクと対話破綻ラベルを予測するタスクのマルチタスク学習を行っている．

いずれの先行研究においても，アノテーション傾向の多様性を考慮するような工夫はなされていない．よって本研究は対話破綻検出においてアノテーション傾向の多様性に着目した初めての研究であると言える．

5.3 提案手法

図 5.1 の概要図に示す通り，提案手法ではまずアノテータ毎のラベルのアノテーション傾向に基づいて訓練データをクラスタリングする（5.3.1 節）．次に，各クラスタ毎の対話破綻検出器を構築する（5.3.2 節）．推論時には，各クラスタ毎の検出器が出力した対話破綻確率をアンサンブルすることで，アノテーション傾向の多様性を考慮した対話破綻検出を行う（5.3.3 節）

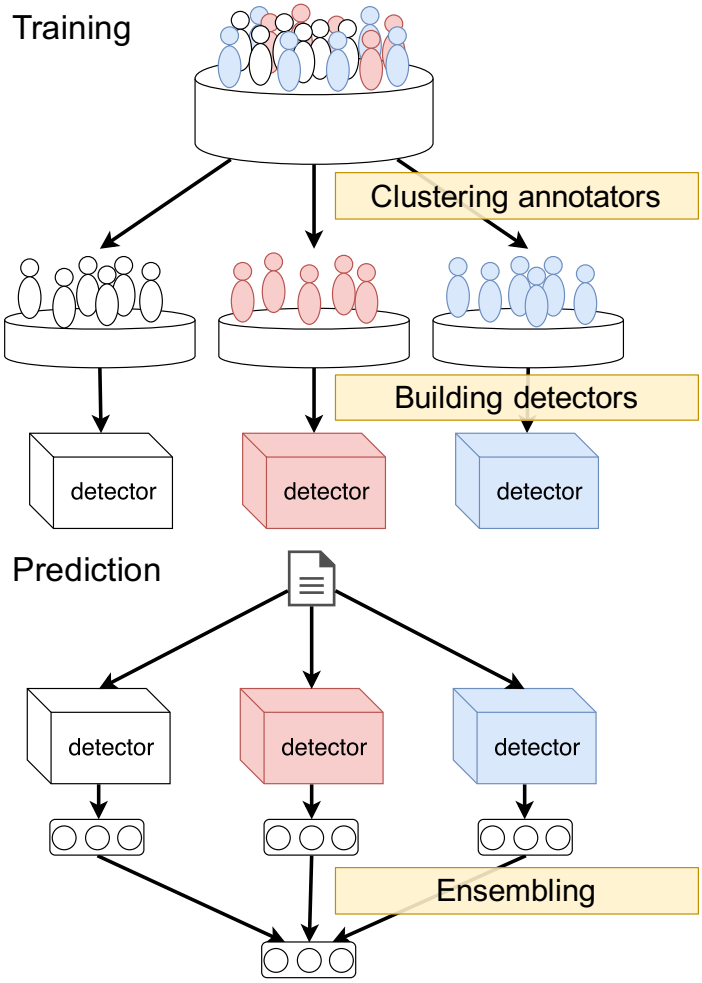

図 5.1　提案手法の概要図

5.3.1　アノテータのクラスタリング

対話破綻検出タスクのデータにおける対話破綻ラベルは複数のアノテータによってアノテーションされている．先行研究 [101, 102] では各発話応答対に付与された複数のラベルを多数決によって一つに集約し，そのラベルを基に対話破綻検出器を訓練している．提案手法ではアノテータ毎のアノテーション傾向の違いに着目し，異なる分類傾向を持つ検出器を複数構築する．本節では，似たアノテーション傾向を持つアノテータをクラスタリングする方法を説明する．

提案手法では図 5.2 に示すように，各アノテータが付与したラベルの分布を特徴量として k-means++ [106] 法でアノテータをクラスタリングする．具体的には，あ

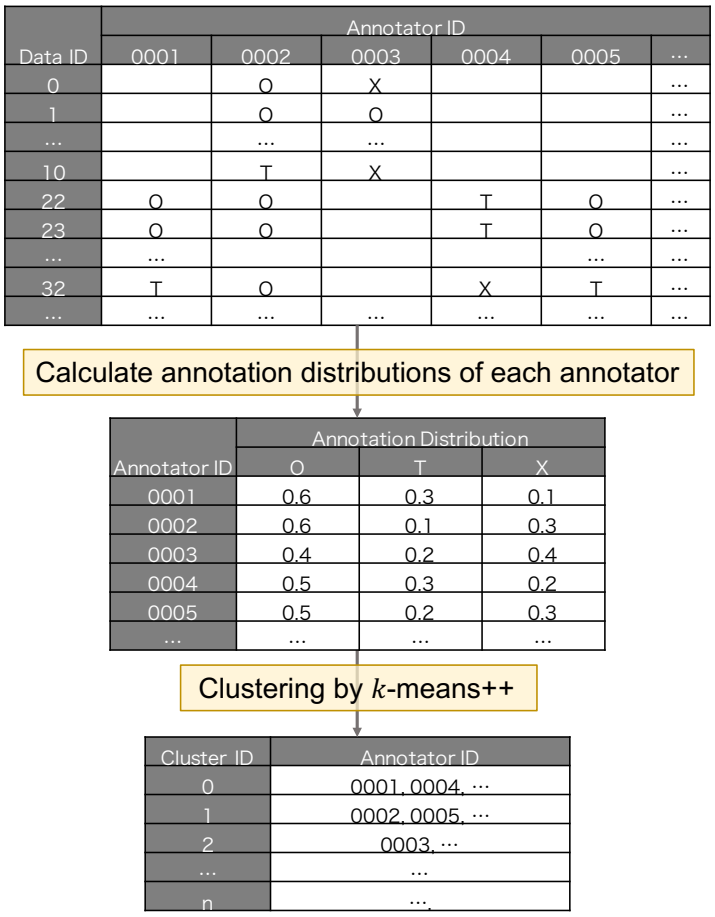

図 5.2　アノテータのクラスタリング方法の概要図

るアノテータを表す特徴量は，そのアノテータが付与した “O”, “T”, “X” ラベルの頻度を，そのアノテータがアノテーションしたデータの総数で割ったものとして表される．クラスタリングによって，あるアノテータはただ一つのクラスタに属することになる．

アノテータをクラスタリングした後，各発話応答対に対するラベルをそれぞれのクラスタ毎に多数決によって一つの値に集約する．この操作によって，同じ発話応答対であってもクラスタによって異なる破綻ラベルを持ち得るようになる．

5.3.2　対話破綻検出モデルの設計

対話破綻には様々な類型が存在する [107]．本研究では単語やフレーズレベルで判断可能な破綻（例えば，話題を示すようなフレーズが発話と応答で食い違う場合など）と文脈レベルでの判断が必要な破綻（例えば，発話における質問にシステム

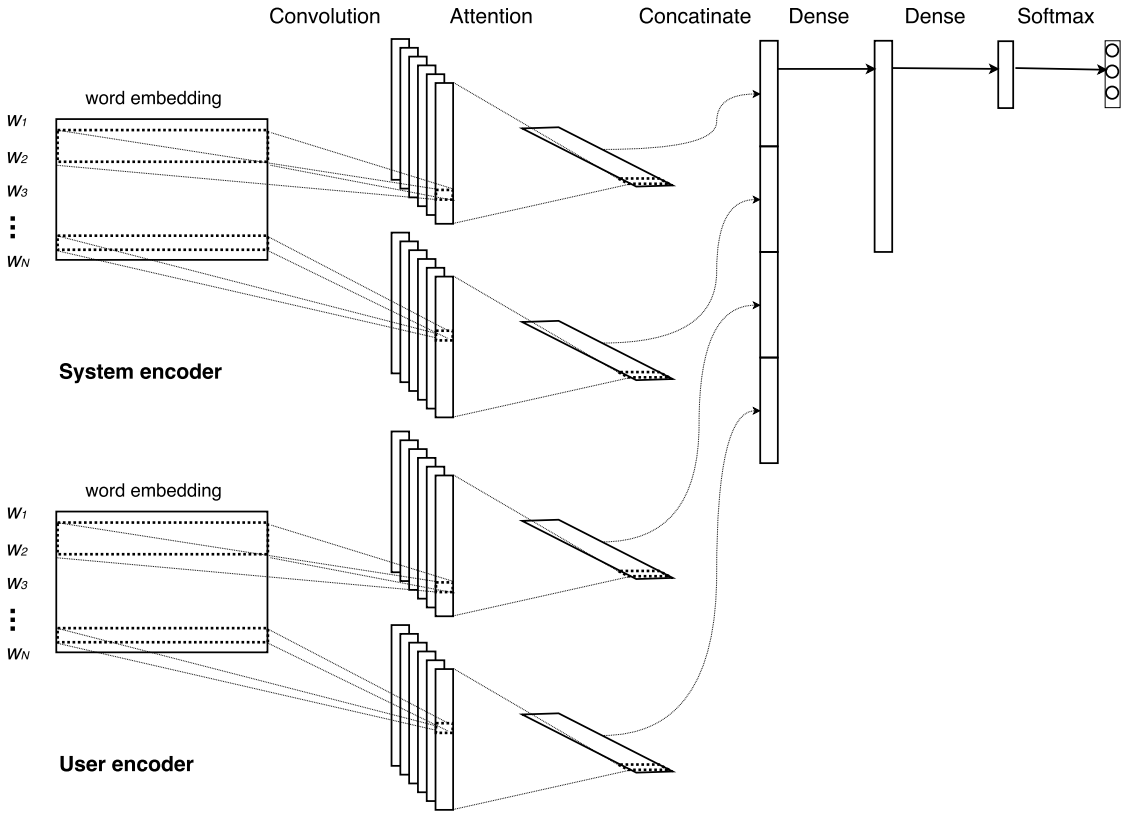

図 5.3　Parallel-CNN モデルのアーキテクチャ

応答が答えられていない場合など）の二つに着目し，それぞれに特化した 2 種類のモデルを構築する．単語やフレーズレベルの破綻に着目した検出器として，CNN エンコーダを用いたモデルを設計する．また，文脈レベルでの破綻に着目した検出器として，LSTM を用いたモデルを設計する．

各検出器に共通の処理として，まず各発話・応答をそれぞれ単語をベクトル空間に射影するニューラルネットワークモデルである CBoW [108] を用いて単語ベクトルの列に変換する．検出器はある発話・応答対に対応する単語ベクトル列の対を入力として受け取り，その対に各破綻ラベル（O: 破綻していない, T: 破綻の可能性がある, X: 破綻している）が付与される確率を予測し出力する．以降，各モデルの詳細を説明する．

Parallel-CNN モデル

CNN は時間非依存な畳み込み処理によって入力系列から局所的な特徴を反映した特徴量を抽出することができる．CNN はその利点を生かして近年ではトピック分類 [109, 110] や関係抽出 [111, 112] などのタスクで活用されている．

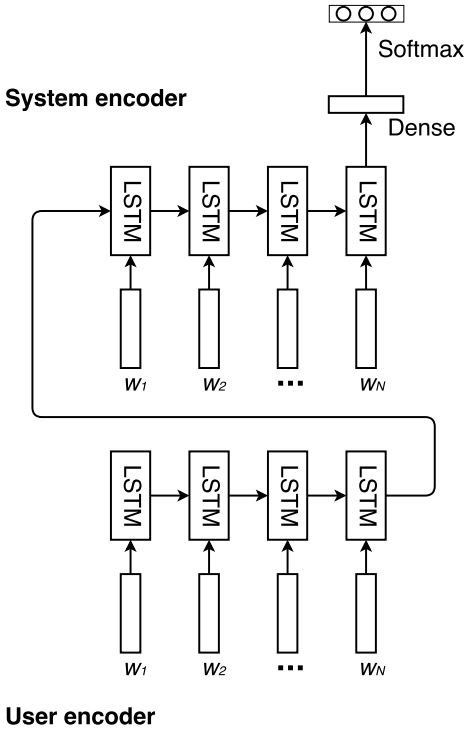

図 5.4　Series-LSTM モデルのアーキテクチャ

本研究では CNN エンコーダを二つ並列に繋いだ Parallel-CNN モデル（図 5.3 ）を設計し，対話破綻検出器を構築する．先行研究 [110] に倣い，ユーザ発話・システム応答それぞれについて畳み込み操作を行った後，アテンション機構によってベクトル化を行う．また，異なる窓幅を持つ CNN エンコーダを複数用意することで，短いフレーズから長いフレーズまで考慮できるようにする．すべてのエンコーダから出力されたベクトルを連結したものを 3 層の多層レイヤーパーセプトロン（Multi layer perceptron; MLP）に入力する．MLP からは 3 次元のベクトルが出力される．これに対して softmax 関数を適用し，各破綻ラベルの確率を計算する．

Series-LSTM モデル

RNN は系列データを先頭から順に，自身の過去の状態と共に再帰的に処理するネットワークであり，時系列を考慮することができる．LSTM [21] は単純な RNN に比べてより長い系列を考慮することができる

本研究では LSTM エンコーダを二つ直列に繋いだ Series-LSTM モデル（図 5.4 ）を設計し，対話破綻検出器を構築する．まず，ユーザ発話を LSTM エンコーダ（図中 “User encoder”）によってエンコードする．その出力をもう一つのエンコー

ダ（図中 “System encoder”）の初期状態として，システム応答もエンコードする．システム応答をエンコードして得られたベクトルを全結合層に入力し，3 次元のベクトルを得る．これに対して softmax 関数を適用し，各破綻ラベルの確率を計算する．

5.3.3 アンサンブルによる対話破綻ラベルの予測

各クラスタごとに構築した対話破綻検出モデルの出力をアンサンブルし，最終的な対話破綻ラベルの予測結果を得る．具体的には，各モデルが出力した各ラベルの予測確率の平均を最終的な予測確率として採用する．最終的な予測確率が最も高いラベルを対話破綻ラベルの予測結果として用いる．

5.4 事前実験

適切なクラスタ数の決定のために，事前実験を行う．また，複数のアンサンブル法との比較を行い，アノテーション傾向を考慮した提案手法の有効性を調査する．

5.4.1 ベースライン

アノテーション傾向を考慮したアンサンブル手法の有効性を検証するために，以下の 2 つのベースラインモデルを構築する．

Without-Clustering
: 同じ訓練データを使用して訓練した複数の対話破綻検出モデルを用いて，その出力をアンサンブルする．各検出モデルの違いは初期化時のパラメータのみとなる．検出モデルは Parallel-CNN モデルを用いて構築する．

Random-Split
: アノテータをランダムに複数のクラスタに分配し，そのクラスタごとに対話破綻検出モデルを構築する．訓練データをランダムにサンプリングして複数のモデルを学習する Bagging [113] に似た手法となる．検出モデルは Parallel-CNN モデルを用いて構築する．

5.4.2 データセット

DBDC3 においては複数の開発用データセットが提供されている．その一覧と説明を以下に示す．

雑談対話コーパス (1,146 対話)

NTT Docomo が提供する雑談対話 API (DCM) を用いて構築したコーパス[1)]．最初の 100 対話は 24 名のアノテータによってアノテーションされており，残りの 1,046 対話は 2, 3 名のアノテータによってアノテーションされている．

DBDC1 開発データ (200 対話)

DCM を用いて構築されたコーパス．対話破綻検出チャレンジ[2)] における開発データとして公開されているもの．各対話は 30 名のアノテータによってアノテーションされている．

DBDC1 評価データ (20 対話)

DCM を用いて構築されたコーパス．対話破綻検出チャレンジにおける評価データとして公開されているもの．各対話は 30 名のアノテータによってアノテーションされている．

DBDC2 開発データ (150 対話)

DCM を用いて構築されたコーパスに加え，Denso IT Laboratories (DIT) 社の対話システムを用いたコーパスと，Ritter [114] らの検索ベース対話システム（IRS）を用いたコーパスから構成されている．対話破綻検出チャレンジにおける開発データとして公開されているもの[3)]．各対話は 30 名のアノテータによってアノテーションされている．

DBDC2 評価データ (150 対話)

DCM, DIT, IRS の各システムを用いて構築されたコーパス．対話破綻検出チャレンジにおける評価データとして公開されているもの．各対話は 30 名のアノテータによってアノテーションされている．

事前実験においては，DBDC2 の評価データを用いて評価を行い，残りのデータ

1） https://sites.google.com/site/dialoguebreakdowndetection/chat-dialogue-corpus
2） https://sites.google.com/site/dialoguebreakdowndetection/
3） https://sites.google.com/site/dialoguebreakdowndetection2/

全てをモデルの学習用に用いた．

単語のベクトル化に用いる CBoW の訓練には，日本語 Wikipedia のダンプデータ[4)]を用いた．Wikipedia データにおける文章は口語体でないため，対話破綻検出タスクのデータのような雑談調のデータへの適用には不十分であると考えられる．そこで，ウェブ上の演劇台本等からクロールした 76,974 文の台詞も訓練データに追加して用いた．

5.4.3 評価指標

対話破綻検出ではラベル一致系統と分布距離系統の 2 種類の観点に基づく評価指標が用いられる [100, 103]．ラベル一致系統では，対話破綻検出タスクを ‘O’, ‘T’, ‘X’ の 3 クラスへの分類タスクと捉え，Accuracy と X ラベルの F 値 を用いて性能を評価する．各データには複数のアノテータによってラベルが付与されているが，最頻のラベルを正解として用いて評価を行う．

一方，分布距離系統は対話破綻を判断する基準が主観的であることを考慮した評価尺度である．具体的には，対話破綻検出器が出力したラベルの確率分布とアノテーションされたラベルの頻度分布の距離を用いて評価を行う．評価尺度としては平均二乗距離（Mean Squared Error; MSE）や JS ダイバージェンス（Jensen-Shannon divergence; JSD）を用いる．これらは共に値が小さいほど検出性能が高いことを意味する．

事前実験においてはラベル一致系統と分布距離系統のそれぞれから F 値と MSE を代表的な尺度として用いる．

5.4.4 実験設定

各発話応答対は日本語形態素解析器である MeCab [5)] [115] を用いて単語単位に分割した．単語のベクトル化に用いる CBoW の訓練時には，訓練データ上で頻度 50 未満の単語をその品詞名に置き換えた．置き換え後の語彙サイズは 13,806 であった．単語ベクトルの次元数は 512 とした．

Parallel-CNN は窓幅 3, 4, 5 の 3 種類の畳み込み層を用いて構成し，それぞれの特徴マップの次元数は 256 に設定した．Series-LSTM モデルにおける LSTM エン

4） https://dumps.wikimedia.org/jawiki/20170920/jawiki-20170920-pages-articles.xml.bz2
5） http://taku910.github.io/mecab/

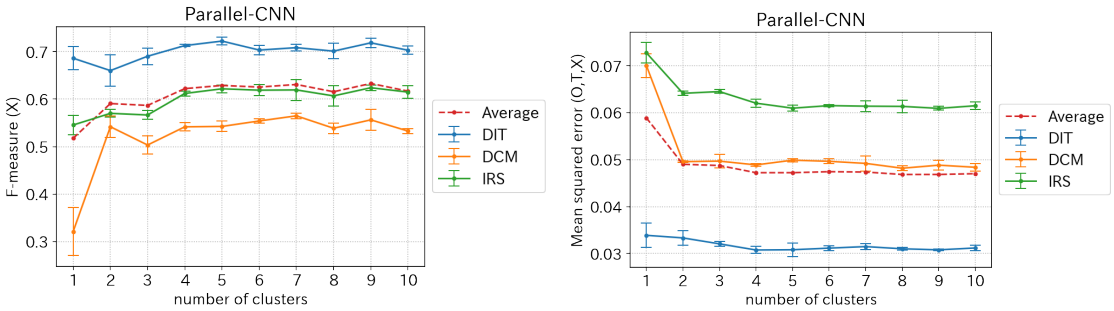

図 5.5　Parallel-CNN モデルにおける，クラスタ数と F 値（左側），MSE（右側）の関係

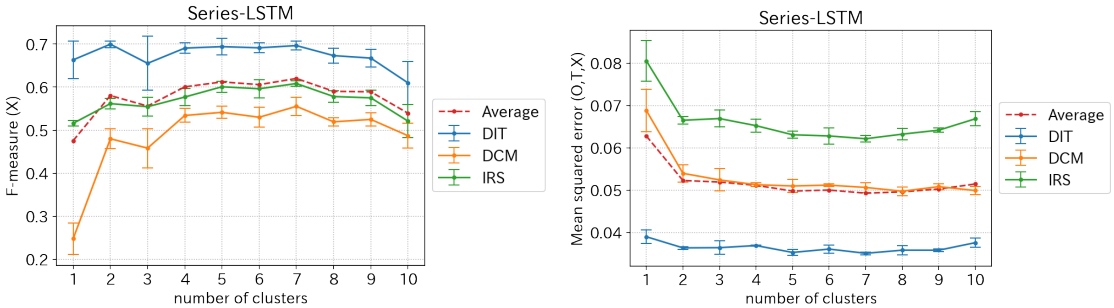

図 5.6　Series-LSTM モデルにおける，クラスタ数と F 値（左側），MSE（右側）の関係

コーダの隠れ状態の次元数は 512 とした．モデルの訓練においては，損失関数として MSE を用い，オプティマイザとしては Adam [95] を用いた．

Parallel-CNN の実装においては深層学習の実装用ライブラリである keras[6] を，また Series-LSTM の実装においては Chainer[7] [116] を用いた．

事前実験では，k-means の k ならびにベースライン手法における検出器の数を 1 から 10 まで変化させ，その性能を検証する．また，モデルのパラメータの初期値のランダム性の影響を考慮し，各モデルにつき 3 回ずつ実験を行い，その平均値と 95% 信頼区間を用いて評価を行う．

5.4.5　結果

Parallel-CNN と Series-LSTM のそれぞれにおける，クラスタ数を増加させたときの F 値と MSE の変化の様子を図 5.5 と図 5.6 に示す．Parallel-CNN と Series-LSTM の両方において，クラスタ数を 2 以上に増やしてアンサンブルすることで F 値と

6）　https://keras.io/ja/
7）　https://tutorials.chainer.org/ja/

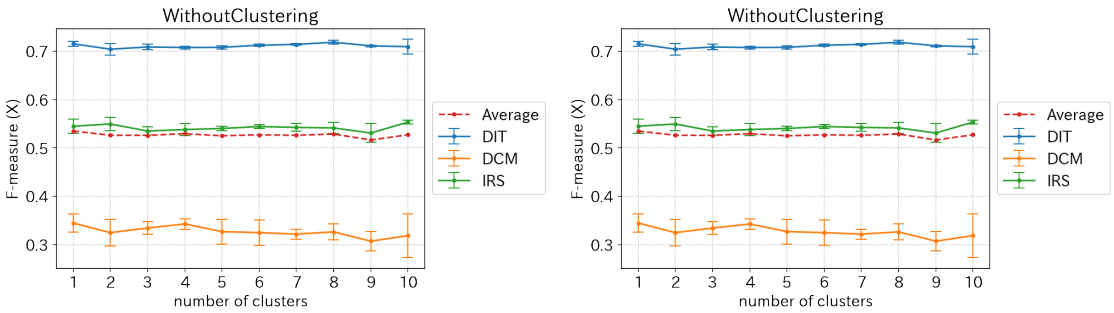

図 5.7　Without-Clustering ベースラインにおける，クラスタ数と F 値（左側），MSE（右側）の関係

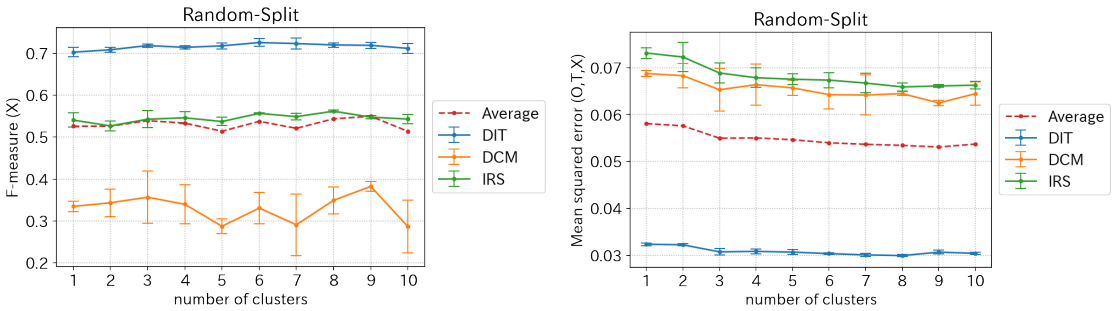

図 5.8　Random-Split ベースラインにおける，クラスタ数と F 値（左側），MSE（右側）の関係

MSE が改善することが確認できる．この傾向は特に DCM において顕著である．Parallel-CNN と Series-LSTM の比較においては有意差は認められなかった．

図 5.7 に，Without-Clustering ベースラインにおける検出モデルの数と F 値，MSE の関係を示す．F 値と MSE の両方において，検出モデルの数を増加させても一貫した性能向上効果は得られなかった．図 5.8 は Random-Split ベースラインにおける実験結果である．Random-Split では，クラスタ数を増加させることでわずかに F 値や MSE が改善することがわかる．しかし，提案手法に比べるとその効果は低い．

表 5.2 に，各モデルにおいて最も F 値が高かったクラスタ数における，F 値と MSE の値を示す．表を見ると，提案手法である Parallel-CNN と Series-LSTM がベースラインの 2 つの手法の結果を上回る値となっていることがわかる．これらの結果から，アノテーション傾向を基にアノテータをクラスタすることで，単純なアンサンブル手法に比べて対話破綻検出性能を向上させられることが確認できる．

表 5.2　事前実験におけるそれぞれのモデルの最良の F 値と MSE とそのときのクラスタ数

モデル	F 値	MSE	クラスタ数
Parallel-CNN	**0.633±0.013**	**0.047±0.001**	9
Series-LSTM	**0.620±0.013**	**0.049±0.001**	7
Without-Clustering Baseline	0.535±0.013	0.057±0.001	9
Random-Split Baseline	0.550±0.008	0.053±0.001	9

表 5.3　DBDC3 における評価結果（X ラベルの F 値）

Runs	平均	DCM	DIT	IRS
CRF (Baseline)	0.5796	0.4502	0.6634	0.6252
Run1	0.6316	**0.4785**	0.7165	0.6998
Run2	0.6230	0.4413	0.7307	0.6968
Run3	**0.6364**	0.4629	**0.7323**	**0.7140**

表 5.4　DBDC3 における評価結果（Accuracy）

Runs	平均	DCM	DIT	IRS
CRF (Baseline)	0.5322	**0.4727**	0.5563	0.5676
Run1	0.5613	0.4690	0.6072	0.6075
Run2	0.5514	0.4600	0.6145	0.5942
Run3	**0.5669**	0.4454	**0.6200**	**0.6208**

5.5　対話破綻検出チャレンジ 3 データセットによる評価実験

対話破綻検出チャレンジ 3（DBDC3）の評価データは 1,650 対のユーザ発話とシステム応答で構成される．内訳は DCM, DIT, IRS のそれぞれの対話システムから構築したものがそれぞれ 550 件ずつとなっている．タスクは事前実験と同様，各システム応答に対して対話破綻ラベル O, T, X が付与される確率をそれぞれ予測するものとなっている．評価尺度は 5.4.3 節で説明したとおり，F 値，Accuracy，MSE，JSD の 4 種類である．

事前実験の結果に基づき，DBDC3 には以下で説明する 3 つのモデルの予測結果

を提出した．

Run1　事前実験において最も高い F 値を達成したときのクラスタ数における Parallel-CNN モデル

Run2　事前実験において最も高い F 値を達成したときのクラスタ数における Series-LSTM モデル

Run3　Run1 のモデルと Run2 のモデルの予測確率を平均することでアンサンブルしたもの

本節では，3 つの Run の結果について分析し，議論を行う．

5.5.1　結果

表 5.3, 5.4, 5.5, 5.6 のそれぞれに，F 値，Accuracy，MSE，JSD のそれぞれにおける評価結果を示す[8)]．DBDC3 ではベースラインとして，ユーザ発話とシステム応答の単語集合を特徴量とした条件付き確率場（Conditional Random Field; CRF）[117] ベースの対話破綻検出器を提供している．その評価結果も掲載する．

評価結果より，DCM, DIT, IRS の平均値ではすべての評価尺度において提案手法が CRF を上回る評価値を達成している．特に，ラベル一致系統（F 値, Accuracy）では Run1 と Run2 のアンサンブルである Run3 が最も良い性能となっている．このことから，検出特性の異なる Series-LSTM と Parallel-CNN をアンサンブルすることで対話破綻検出性能の向上に寄与することがわかる．分布距離系統（MSE, JSD）の評価においては，提案手法は CRF ベースライン大きく上回っている．これは，提案手法はアノテーション傾向の偏りを反映するように分類器を複数構築しているため，アノテーションに対する近似性能が高いのだと考えられる．

5.5.2　エラー分析

DBDC3 における提案手法の出力結果を用いてエラー分析を行う．まず，Parallel-CNN と Series-LSTM の検出特性の比較を行う．その後，両方のモデルが誤った事例について議論する．

8）　DBDC3 に参加した全チームの評価結果は文献 [103] を参照

表 5.5　DBDC3 における評価結果（MSE）

Runs	平均	DCM	DIT	IRS
CRF (Baseline)	0.1996	0.2169	0.1870	0.1950
Run1	**0.0465**	**0.0519**	**0.0352**	0.0525
Run2	0.0498	0.0545	0.0407	0.0543
Run3	0.0469	0.0525	0.0365	**0.0517**

表 5.6　DBDC3 における評価結果（JSD）

Runs	平均	DCM	DIT	IRS
CRF (Baseline)	0.3851	0.4100	0.3741	0.3713
Run1	**0.0891**	**0.0994**	**0.0699**	**0.0979**
Run2	0.0968	0.1048	0.0801	0.1054
Run3	0.0912	0.1011	0.0731	0.0994

Parallel-CNN と Series-LSTM の特性の違い　5.3.2 節でも言及したように，Series-LSTM は文脈レベルでの判断が必要な破綻の検出に有利という仮定をおいて構築したモデルであり，Parallel -CNN は単語や短いフレーズレベルでの判断が必要な破綻の検出に有利という仮定をおいたモデルである．本節では，2 つのモデルのうちどちらかのみが誤判定した発話応答対を分析し，仮定の妥当性を調査する．

Parallel-CNN だけが正解した事例は 88 例，Series-LSTM だけが正解した事例は 72 例であった．DBDC3 の評価結果からも明らかであった通り，CNN ベースの方が検出性能は高いことがわかる．図 5.9 に，応答文の文長（応答文内の単語数）と各モデルの誤検出の割合の関係を示すヒストグラムを示す．図より，Parallel-CNN は長い応答文に対して誤検出しやすく，また Series-LSTM は短い応答に対して誤検出しやすいという傾向が見て取れる．実際，どちらかのシステムのみが誤検出した応答の平均単語数は Parallel-CNN では 15.25 ，Series-LSTM では 11.27 で，これは両側 t-検定における p 値が 0.023 であることから 5% 有意水準で有意差が認められる．以上の結果から，Series-LSTM は Parallel-CNN よりも長い文脈の対話破綻に頑健であることが確認できる．

Parallel-CNN と Series-LSTM の両方が誤検出した事例　Parallel-CNN と Series-LSTM の両方が誤検出した事例について，Out-of-Vocabulary (OOV) 率，文長，品

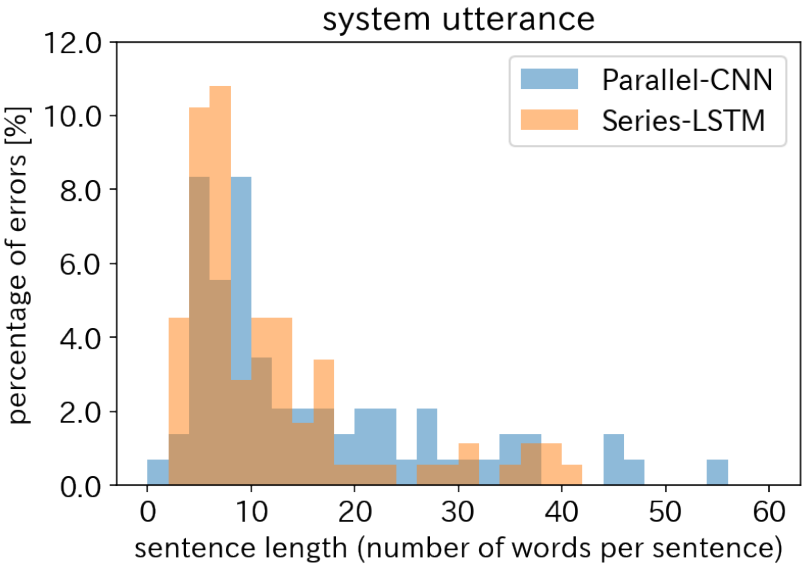

図 5.9　応答文の文長とエラー率の関係

表 5.7　予測ラベル・正解ラベル別の平均文長の傾向

	予測ラベル		正解ラベル		予測結果	
	O	X	O	X	正解	不正解
ユーザ発話	13.3	10.2	**6.7**	**9.7**	**7.9**	**9.6**
システム応答	27.0	20.5	**15.3**	**19.9**	18.2	19.2
絶対誤差	17.3	13.1	**11.1**	**12.4**	12.1	12.8

詞の種類などの観点から分析した．

まず，OOV 数の影響を分析する．OOV とは，訓練データにおいて出現しない単語や頻度が著しく低い単語などのように，モデルが扱える語彙に含まれない単語を指す．直感的には OOV 率が高い事例は破綻を正しく検出するのが難しいと考えられる．モデルが正解した事例（778 発話応答対）における平均 OOV 数は 1.83 語，誤検出した事例（613 対）における平均 OOV 数は 2.01 語であった．両側 t 検定における p 値は 0.08 であり，有意差があるとは言えない．よって，提案手法は OOV に対して特に敏感であるとも頑健であるとも言えない．

次に文長の影響を調査する．ユーザ発話とシステム応答のそれぞれについて，正解した事例と誤検出した事例の平均単語数，またそれらの差の絶対値を算出した．表 5.7 に，予測ラベルと正解ラベルのそれぞれについて，‘O’ ラベルと ‘X’ ラベル

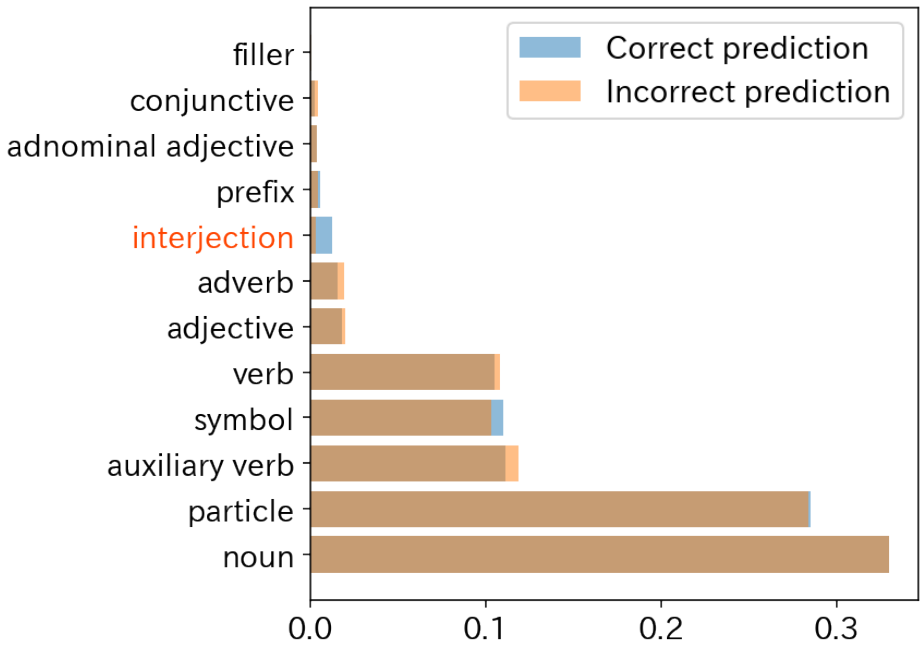

図 5.10　システム応答文中の品詞ごとの，検出成功率と誤検出率の比較

それぞれの事例の平均単語数を示す．また，予測結果が正解だった場合と不正解だった場合のそれぞれにおける平均単語数も掲載する．‘O’ の場合と ‘X’ の場合の値に 5% 有意差が認められるものを太字で示した．正解ラベルに関する統計値を見ると，(1) ユーザの発話が長い場合，(2) システムの応答が長い場合，(3) 発話と応答のいずれかが他方よりも著しく長い場合に対話破綻が起きやすいことがわかる．これは，ユーザの発話は長ければ長いほどその理解が難しくなることや，システム応答は長ければ長いほど矛盾なく生成するのが難しいという直感からも理解しやすい．しかし，予測ラベルに関する統計値を見ると，正解ラベルの場合の傾向とは逆に，モデルは長い発話や応答に対して ‘O’ (破綻ではない) ラベルを付与しやすいことがわかる．発話や応答の文長は対話破綻検出の判断基準として有用であると考えられるため，モデルの特徴量として用いることで検出性能が改善される可能性がある．

最後に，品詞の影響を調査する．応答文における品詞の出現確率をモデルが正解した場合，誤検出した場合のそれぞれについて調査した．その結果のヒストグラムを図 5.10 に示す．図より，提案手法は間投詞（interjection）を除いて，特定の品詞に対して敏感に反応するわけではないことがわかる．「Yes」や「No」のような間投

詞は単体で文を構成することができるため，これらが含まれる応答は短い文長であることが多いと考えられる．そのため，破綻の検出が容易なのだと推察される．

5.6 対話応答生成モデルへの適用

本節では，第 4 章において構築した対話応答生成モデルの生成結果に対して提案手法がどの程度正確に対話破綻検出できるか検証を行う．

5.6.1 検証方法

第 4 章における提案手法（$s = s_{max}$; 具体性最大化デコード）が生成した応答のうち，4.5.3 節において人手評価ラベルを付与した 300 事例を用いて検証を行う．人手評価における「破綻」ラベルを ‘X’ ラベルとして，それ以外のラベルについては ‘O’ として扱い，評価を行う．ここで，‘X’ ラベルと ‘O’ ラベルの出現頻度の比が 6:4 と偏っているため，5:5 になるように ‘X’ ラベルのデータをサンプリングする．結果として，事例数は 230 となった．評価指標としては X（破綻）ラベルの F 値と Accuracy を用いる．なお，事例数が少ないため，これら全てを評価データとして用いる．訓練は 5.4.2 節で用いたデータで行う．

比較に用いるモデル名の一覧とその説明を以下に示す．

Parallel-CNN 単一モデル　アノテータのクラスタリングを行わずに，単一の Parallel-CNN モデルを全ての訓練データを用いて訓練したベースライン．

Series-LSTM 単一モデル　アノテータのクラスタリングを行わずに，単一の Series-LSTM モデルを全ての訓練データを用いて訓練したベースライン．

Parallel-CNN Random-Split　アノテータをランダムにクラスタリングし，各クラスタごとに複数の Parallel-CNN を訓練してアンサンブルを行うベースライン．事前実験の結果より，クラスタ数は 9 とした．

Series-LSTM Random-Split　アノテータをランダムにクラスタリングし，各クラスタごとに複数の Series-LSTM を訓練してアンサンブルを行うベースライン．検証データでの F 値 を基に，クラスタ数は 10 とした．

Parallel-CNN　提案手法によって訓練した Parallel-CNN モデル．DBDC3 に提出した Run1 モデルと同じ設定．

表 5.8　対話応答生成モデルの生成応答への対話破綻検出の適用結果

モデル	F 値	Accuracy
Parallel-CNN 単一モデル	0.302	0.557
Series-LSTM 単一モデル	0.291	0.570
Parallel-CNN Random-Split	0.197	0.539
Series-LSTM Random-Split	0.114	0.526
Parallel-CNN	0.622	0.587
Series-LSTM	0.573	0.565
アンサンブル	**0.643**	**0.594**

Series-LSTM　提案手法によって訓練した Series-LSTM モデル．DBDC3 に提出した Run2 モデルと同じ設定．

アンサンブル　提案手法によって訓練した Parallel-CNN モデルと Series-LSTM モデルのアンサンブル．DBDC3 に提出した Run3 モデルと同じ設定．

5.6.2　検証結果

実験結果を表 5.8 に示す．F 値，Accuracy 共に，提案手法がベースラインの手法を上回る結果となった．特に，Parallel-CNN と Series-LSTM をアンサンブルしたものが最も性能が高い（F 値は 0.643，Accuracy は 0.594）．検出特性の異なる 2 モデルを組み合わせることで対話応答生成モデルが生成した多様な応答に対応できるようになったものと考えられる．Accuracy は最高でも 6 割程度という結果に留まったが，これは対話破綻検出モデルの訓練データと対話応答生成モデルの訓練データのドメインの違いによるものと考えられる．対話破綻検出チャレンジの各開発データの構築に用いられている対話システムの種類は少なく，対話応答生成モデルのような深層学習ベースのモデルは用いられていない．今後より汎用的な対話破綻検出器を構築するためには，対話応答生成モデルを含む様々な種類の対話システムを用いて対話破綻検出データセットを構築する必要があると考えられる．

表 5.9　対話応答生成モデルへの対話破綻検出モデルの適用例

例番号	ユーザ発話	システム応答
	予測ラベル = “O”, 正解ラベル= “O”	
例 1	うちの庭から海みえるの最高だと思うの	綺麗なのだよね
例 2	カレーうどん付きお得セット	カレーうどん食べたい
	予測ラベル = “O”, 正解ラベル= “X”	
例 3	もしぼくが山奥で遭難したらどうするん	山奥にして
例 4	リア友フォローしてるの?	リア友と友だちしてる
	予測ラベル = “X”, 正解ラベル= “X”	
例 5	早いね夏らしいことは何かした?	夏の夏には夏の夏になるらしいです
例 6	熱計ってねなさい!	熱は委員長だよ
	予測ラベル = “X”, 正解ラベル= “O”	
例 7	イカの踊り食いとか美味しいよ	エビフライの方がいい
例 8	今日はもともとお休み	おはよー!体調は大丈夫かな?

5.6.3　事例分析

最も高い検出性能であったアンサンブルモデルについて，その予測例を用いて事例分析を行う．表 5.9 に，予測ラベルと正解ラベル別にそれぞれ 2 例ずつの発話応答対を提示した．例 1 と例 2 は破綻していないことを正しく検出できた例である．例 2 における「カレーうどん」のように，発話中で話題にしている単語（主に名詞や形容詞）が応答でもそのまま出現している場合に予測ラベルが “O” となることが多かった．そのため，例 3 や例 4 のように，正解ラベルが “X” であるにも関わらず “O” を予測してしまった事例が多く観測された．発話と応答の間の名詞・形容詞[9)]の一致度合いを Jaccard 係数を用いて計測したところ，予測ラベルが “O” の場合は平均 0.213，“X” の場合は平均 0.147 で，Welch の t 検定の結果 10% 有意水準で有意な差が見られた．また，正解ラベル別での比較においては “O” の場合と “X” の場合のそれぞれで平均 0.198, 0.147 となり，こちらも 10% 有意水準で有意差があった．このことから，名詞や形容詞の一致度合いは対話破綻検出の手がかりとして有用ではあるものの，モデルはそれに過度に依存してしまっていると考えられる．

9）　形容詞については全て原形に戻した

例 5 のように単語やフレーズを過度に繰り返してしまっているような応答に関しては正しく “X” ラベルを予測できている傾向が見られた．また，例 6 のように発話とは関係なく応答文そのものの文意が取れないような例に関しても “X” ラベルを正しく予測できていた．例 7 のように，発話と応答でそれぞれ話題にしている単語が関連はするものの全く別の単語であるような場合には，正解ラベルによらず “X” を出力してしまう傾向が観測された．例 8 のように，発話中に挨拶の意図を持つフレーズが含まれていないにも関わらず応答中に挨拶が含まれているような場合も “X” が出力されやすい傾向が見られた．これは，挨拶を含む発話には挨拶を含む応答が共起しやすいという傾向に過適合してしまったためであると考えられる．

5.7 おわりに

対話破綻を感じる基準は人によって異なることに着目し，対話破綻ラベルのアノテーション傾向に基づいてアノテータをクラスタリングし，クラスタごとに対話破綻検出モデルを構築してアンサンブルする対話破綻検出手法を構築した．事前実験では，ベースラインとして採用した他のアンサンブル手法に対する提案手法の優位性を示した．対話破綻検出の共同タスクである対話破綻検出チャレンジ 3 においては，提案手法はベースラインを大きく上回る検出性能を記録した．また，第 4 章にて構築した対話応答生成モデルの生成結果への適用実験も行った．この実験においても，提案手法はベースラインモデルの検出性能を大きく改善することを確認した．事例分析においては，発話と応答の間で話題は一貫していても用いられている単語が異なるような事例において，検出性能に改善の余地があることを明らかにした．これは，モデルが十分に幅広い話題の知識を獲得できていないことが原因だと考えられる．

今後は，より多くの話題を網羅した対話破綻ラベル付きデータの構築や，ラベルなしデータを用いた効率的な事前訓練などを行うことで，より汎用的な対話破綻検出モデルを構築する必要があると考えられる．

第 6 章　結論

対話応答生成モデルは従来の対話システムに比べて流暢な応答を生成できるという利点を持つ一方で，抽象的な発話に対してその意図に適切に応える応答を生成できないという問題がある．また，しばしば「そうですね」や「わかりません」といったどのような発話に対しても成り立つ汎用かつ具体的でない応答を生成してしまう問題，さらに対話を破綻させてしまうような応答を生成してしまうという問題がある．これらの問題は人間との円滑なコミュニケーションを妨げるものとなる．本論文では発話や応答の具体性に関するこれらの課題を解決するために以下の 3 つの研究に取り組んだ．

第 3 章では，抽象的な発話の意図を理解し適切な応答を生成する手法の構築に取り組んだ．まず抽象的な発話と具体的な発話の対を含む 7 万件規模の対話コーパス DIRECT (Direct and Indirect REsponses in Conversational Text corpus) を構築した．また，対話応答生成モデルへの入力発話を事前により具体的な発話に言い換える手法を提案し，実験においてその有効性を確認した．

第 4 章では，対話応答生成モデルがしばしば具体性の低い発話を生成してしまうという問題に対し，具体性の高い応答を生成可能な技術の実現を目的として研究を行った．人間は常に具体性の高い応答を行うのではなく必要に応じて応答の具体性を使い分けていることに着目し，具体性を制御可能な対話応答生成手法の構築に取り組んだ．発話に対する応答の具体性を測る簡潔な尺度 MaxPMI を提案し，そのスコアを制御変数として訓練データに付与して訓練することで，具体性の制御を可能にした．評価実験では，提案手法が先行研究に比べてより応答の具体性を制御できることを，自動評価・人手評価等様々な観点から検証・確認した．

第 5 章では対話システムが生成した破綻応答を検知する対話破綻検出タスクに取り組んだ．対話破綻を感じる基準は人によって異なることに着目し，破綻ラベルのアノテーション傾向に基づいて訓練データのアノテータをクラスタリングし，様々なアノテーション傾向を再現する複数の検出モデルを構築してアンサンブルする手法を提案した．実験においては，他のアンサンブル手法や従来の対話破綻検出

手法に比べて提案手法が高い検出性能を達成できることを確認した．また，第 4 章の対話応答生成モデルの生成結果に対して対話破綻検出モデルを適用した場合の性能比較も行い，提案手法の優位性を示した．

本研究では，抽象的な発話の言外に暗示された要求を推定し，適切な具体性を持つ応答を生成する対話応答生成技術の開発と，対話応答生成モデルが生成した破綻応答を検知する対話破綻検出モデルの技術の開発に取り組んだ．研究全体を通して，人間との円滑なコミュニケーションが可能な対話システムの実現に貢献した．一方で，今後の課題も浮き彫りになった．例えば第 3 章では抽象的な発話と具体的な発話の対からなるコーパスを構築することで発話の事前言い換えを実現したが，雑談対話などの幅広いドメインへの対応は現状では対応できない．また，第 5 章では構築した対話破綻検出モデルを第 4 章のデータに適用する実験を行ったが，ドメインの違いから全体的には低い検出性能であった．これらの課題の解決のために，より広いドメインからなる対話コーパスをそれぞれのタスクについて構築する必要があると考えられる．

また，本研究では人間との円滑なコミュニケーションが可能な対話システムの実現に向けて 3 つの要素技術の開発に取り組んだが，これらの要素技術を統合して実用的な対話システムを構築するにあたってさらに新たな課題が生じると考えられる．例えば，対話破綻検出モデルで破綻応答を検出した場合に破綻していない応答を生成する技術の開発を行う必要がある．単純な方法としては，ビームサーチやサンプリング [118] などを用いてあらかじめ複数の応答候補を生成しておき，破綻応答だと判定されなかったもののみを用いるような方法が考えられる．しかし，応答候補の中には必ずしも破綻していない応答が含まれているとは限らないため，より確実に破綻していない応答が含まれるような手法を構築する必要があると考えられる．今後はより実用的な対話システムの構築のために，各要素技術の統合に取り組む．

謝辞

博士前期課程から後期課程の 5 年間にわたり研究内容に関するご指導や研究生活に関するご支援をいただきました荒瀬由紀先生，鬼塚真先生，佐々木勇和先生に深く感謝申し上げます．また，本論文の第 3 章における共同研究者としての議論のみならず，博士課程での研究生活に関するアドバイス等様々な機会でお世話になりました愛媛大学大学院理工学研究科の梶原智之先生にも厚く感謝申し上げます．また，大阪大学大学院情報科学研究科の下條真司先生，原隆浩先生には本論文の内容に関して貴重なご助言を賜りましたことを深く感謝いたします．

博士前期課程への入学以来 5 年間を過ごした大阪大学ビッグデータ工学講座・鬼塚研究室の先輩・同期・後輩の皆様にも感謝いたします．研究に関する議論はもちろん，日々の雑談や休日の外出など，皆様のおかげで充実した研究生活を送ることができました．特に，第 5 章の研究を共同研究者として一緒に進めてくださった野本英梨子さん，頼りないながらも共著者として研究に携わらせてくださった近井厚三さん，田中昂志さん，大橋空さんには深く感謝申し上げます．

最後に，博士後期課程への進学に際し快く背中を押してくれた母と父，いつでも深い理解をもって私を支えてくれた長姉，兄，次姉，妹を始めとする家族・親類の皆に感謝し，謝辞といたします．

参考文献

[1] Joseph Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, Vol. 9, pp. 36–45, 1966.

[2] Oriol Vinyals and Quoc V Le. A neural conversational model. In *Proceedings of the International Conference on Machine Learning*, July 2015.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 3104–3112, December 2014.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, July 2014.

[5] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems*, December 2017.

[6] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems*, Vol. 38, No. 3, apr 2020.

[7] Richárd Csáky, Patrik Purgai, and Gábor Recski. Improving neural conversational models with entropy-based data filtering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 5650–5669, July

2019.

[8] Reina Akama, Sho Yokoi, Jun Suzuki, and Kentaro Inui. Filtering noisy dialogue corpora by connectivity and content relatedness. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 941–958, November 2020.

[9] Andrei Malchanau, Volha Petukhova, and Harry Bunt. Towards continuous dialogue corpus creation: writing to corpus and generating from it. In *Proceesings of the Language Resources and Evaluation Conference*, May 2018.

[10] Ze Yang, Wei Wu, Can Xu, Xinnian Liang, Jiaqi Bai, Liran Wang, Wei Wang, and Zhoujun Li. StyleDGPT: Stylized response generation with pre-trained language models. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 1548–1559, November 2020.

[11] Tong Niu and Mohit Bansal. Polite dialogue generation without parallel data. *Transactions of the Association of Computational Linguistics*, Vol. 6, pp. 373–389, 2018.

[12] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 2204–2213, July 2018.

[13] Junya Takayama, Tomoyuki Kajiwara, and Yuki Arase. DIRECT: Direct and indirect responses in conversational text corpus. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 1980–1989, November 2021.

[14] 高山隼矢, 梶原智之, 荒瀬由紀. 対話における間接的応答と直接的応答からなる言い換えコーパスの構築と分析. 言語処理学会論文誌 自然言語処理, Vol. 29, No. 1, March 2021. to appear.

[15] Junya Takayama and Yuki Arase. Relevant and informative response generation using pointwise mutual information. In *Proceedings of the Workshop on NLP for Conversational AI*, pp. 133–138, August 2019.

[16] Junya Takayama and Yuki Arase. Consistent response generation with controlled specificity. In *Proceesings of the Conference on Empirical Methods in Natural*

Language Processing: Findings, pp. 4418–4427, November 2020.

[17] Junya Takayama, Eriko Nomoto, and Yuki Arase. Dialogue breakdown detection robust to variations in annotators and dialogue systems. *Computer Speech and Language*, Vol. 54, pp. 31–43, 2019.

[18] Junya Takayama, Eriko Nomoto, and Yuki Arase. Dialogue breakdown detection considering annotation biases. In *Proceedings of Dialog System Technology Challenge 6 Workshop*, December 2017.

[19] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association*, pp. 1045–1048, September 2010.

[20] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 48–54, 2003.

[21] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, November 1997.

[22] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceesings of the Conference on Natural Language Learning*, pp. 280–290, August 2016.

[23] Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. Global encoding for abstractive summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 163–169, July 2018.

[24] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pp. 10–19, September 2017.

[25] Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3168–3180, June 2019.

[26] Xingxing Zhang and Mirella Lapata. Sentence simplification with deep reinforcement learning. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 584–594, September 2017.

[27] Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. Complexity-weighted loss and diverse reranking for sentence simplification. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3137–3147, June 2019.

[28] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, July 2020.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020.

[30] Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 3391–3405, November 2020.

[31] Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 1823–1840, November 2020.

[32] Han Huang, Tomoyuki Kajiwara, and Yuki Arase. Definition modelling for appropriate specificity. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2499–2509, November 2021.

[33] Mihir Kale and Abhinav Rastogi. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the International Conference on Natural Language Gen-*

eration, pp. 97–102, December 2020.

[34] San Kim, Jin Yea Jang, Minyoung Jung, and Saim Shin. A model of cross-lingual knowledge-grounded response generation for open-domain dialogue systems. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 352–365, November 2021.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, July 2002.

[36] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2122–2132, November 2016.

[37] Tianyu Zhao, Divesh Lala, and Tatsuya Kawahara. Designing precise and robust dialogue response evaluators. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 26–33, July 2020.

[38] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the International Conference on Human Language Technology Research*, p. 138–145, March 2002.

[39] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, June 2016.

[40] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1116–1126, July 2017.

[41] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. Learning to control the specificity in neural response generation. In *Proceedings*

of the Annual Meeting of the Association for Computational Linguistics, pp. 1108–1117, July 2018.

[42] Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. Linguistically-Informed Specificity and Semantic Plausibility for Dialogue Generation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3456–3466, June 2019.

[43] Yahui Liu, Wei Bi, Jun Gao, Xiaojiang Liu, Jian Yao, and Shuming Shi. Towards less generic responses in neural conversation models: A statistical re-weighting method. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2769–2774, October-November 2018.

[44] Tianxing He and James Glass. Negative training for neural dialogue response generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 2044–2058, July 2020.

[45] John R. Searle. Indirect speech acts. In *Expression and Meaning: Studies in the Theory of Speech Acts*, p. 30–57. Cambridge University Press, 1979.

[46] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the International Joint Conference on Natural Language Processing*, pp. 986–995, November 2017.

[47] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5016–5026, October-November 2018.

[48] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceesings of the Language Resources and Evaluation Conference*, pp. 422–428, May 2020.

[49] Yufan Zhao, Can Xu, and Wei Wu. Learning a simple and effective model

for multi-turn response generation with auxiliary tasks. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 3472–3483, November 2020.

[50] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 270–278, July 2020.

[51] "Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher". A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems*, Vol. 33, December 2020.

[52] Louisa Pragst and Stefan Ultes. Changing the level of directness in dialogue using dialogue vector models and recurrent neural networks. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue*, pp. 11–19, July 2018.

[53] Annie Louis, Dan Roth, and Filip Radlinski. "I'd rather just go to bed": Understanding indirect answers. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 7411–7425, November 2020.

[54] C. Raymond Perrault. A plan-based analysis of indirect speech act. *American Journal of Computational Linguistics*, Vol. 6, No. 3-4, pp. 167–182, 1980.

[55] Sabrina Wilske and Geert-jan Kruijff. Service robots dealing with indirect speech acts. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 4698–4703, October 2006.

[56] Antonio Roque, Alexander Tsuetaki, Vasanth Sarathy, and Matthias Scheutz. Developing a corpus of indirect speech act schemas. In *Proceesings of the Language Resources and Evaluation Conference*, pp. 220–228, May 2020.

[57] Penelope Brown, Stephen C. Levinson, and John J. Gumperz. *Politeness: Some Universals in Language Usage*. Studies in Interactional Sociolinguistics. Cambridge University Press, 1987.

[58] Swati Gupta, Marilyn A. Walker, and Daniela M. Romano. POLLy: A conversational system that uses a shared representation to generate action and social language. In *Proceedings of the International Joint Conference on Natural Lan-*

guage Processing, January 2008.

[59] Mauajama Firdaus, Asif Ekbal, and Pushpak Bhattacharyya. Incorporating politeness across languages in customer care responses: Towards building a multilingual empathetic dialogue agent. In *Proceesings of the Language Resources and Evaluation Conference*, pp. 4172–4182, May 2020.

[60] Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1234–1245, August 2018.

[61] Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 639–649, July 2020.

[62] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*, October 2005.

[63] Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1224–1234, September 2017.

[64] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 1–9, June 2007.

[65] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the International Workshop on Semantic Evaluation*, pp. 1–8, August 2014.

[66] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, September 2015.

[67] Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue natural language inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 3731–3741, July 2019.

[68] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Interactive Poster and Demonstration Sessions*, July 2004.

[69] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, Vol. 1, No. 6, pp. 80–83, 1945.

[70] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pp. 3982–3992, November 2019.

[71] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the International Workshop on Semantic Evaluation*, pp. 1–14, August 2017.

[72] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, June 2019.

[73] Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1298–1308, June 2019.

[74] Yunyi Yang, Yunhao Li, and Xiaojun Quan. Ubar: Towards fully end-to-end task-oriented dialog systems with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, February 2021.

[75] Dian Yu and Zhou Yu. MIDAS: A dialog act annotation scheme for open domain HumanMachine spoken conversations. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1103–

1120, April 2021.

[76] Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 917–929, November 2020.

[77] Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, Vol. 13, No. 1, pp. 1–126, December 2018.

[78] M. G. Kendall. A new measure of rank correlation. *Biometrika*, Vol. 30, No. 1/2, pp. 81–93, 1938.

[79] "Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot". Muss: Multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*, 2021.

[80] Kunal Chawla and Diyi Yang. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 2340–2354, November 2020.

[81] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, October 2020.

[82] "Ilya Loshchilov and Frank Hutter". Decoupled weight decay regularization. In *Proceesings of the International Conference on Learning Representations*, May 2019.

[83] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[84] Shaojie Jiang and Maarten de Rijke. Why are sequence-to-sequence models so dull? understanding the low-diversity problem of chatbots. In *Proceedings of the*

EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI, pp. 81–86, October 2018.

[85] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the Conference on Neural Information Processing Systems*, December 2018.

[86] Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. Improving neural response diversity with frequency-aware cross-entropy loss. In *Proceedings of the Web Conference*, p. 2879–2885, May 2019.

[87] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, and Kam-Fai Wong. An attentional neural conversation model with improved specificity. *arXiv preprint arXiv:1606.01292*, 2016.

[88] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, November 2016.

[89] Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, Xiaolong Wang, Zhuoran Wang, and Chao Qi. Neural response generation via gan with an approximate embedding layer. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 617–626, December 2017.

[90] Ryo Nakamura, Katsuhito Sudoh, Koichiro Yoshino, and Satoshi Nakamura. Another diversity-promoting objective function for neural dialogue generation. In *Proceedings of The AAAI Workshop on Reasoning and Learning for Human-Machine Dialogues*, February 2019.

[91] Ayaka Ueyama and Yoshinobu Kano. Diverse dialogue generation with context dependent dynamic loss function. In *Proceesings of the International Conference on Computational Linguistics*, pp. 4123–4127, December 2020.

[92] Can Xu, Wei Wu, Chongyang Tao, Huang Hu, Matt Schuerman, and Ying Wang. Neural response generation with meta-words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 5416–5426, July 2019.

[93] An Nguyen Le, Ander Martinez, Akifumi Yoshimoto, and Yuji Matsumoto. Im-

proving sequence to sequence neural machine translation by utilizing syntactic dependency information. In *Proceedings of the International Joint Conference on Natural Language Processing*, pp. 21–29, November 2017.

[94] Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4715–4728, July 2020.

[95] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proceesings of the International Conference on Learning Representations*, May 2015.

[96] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pp. 8024–8035. Curran Associates, Inc., 2019.

[97] Joseph L. Fleiss and Jacob Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, Vol. 33, No. 3, pp. 613–619, 1973.

[98] Jianing Li, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. On the relation between quality-diversity evaluation and distribution-fitting goal in text generation. In *Proceedings of the International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, pp. 5905–5915, July 2020.

[99] Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems*, pp. 25–33, April 2021.

[100] Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, and Michimasa Inaba. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In *Proceesings of the Language Resources and Evaluation*

Conference, pp. 3146–3150, May 2016.

[101] Michimasa Inaba and Kenichi Takahashi. Dialogue breakdown detection using RNN encoders. 人工知能学会 言語・音声理解と対話処理研究会, October 2016.

[102] Takahiro Kubo and Hiroki Nakayama. Learning dialog and its breakdowns simultaneously by neural conversational model. 人工知能学会 言語・音声理解と対話処理研究会, October 2016.

[103] Ryuichiro Higashinaka, Kotaro Funakoshi, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi, and Nobuhiro Kaji. Overview of dialogue breakdown detection challenge 3. In *Proceedings of Dialog System Technology Challenge 6 Workshop*, 2017.

[104] Hiroaki Sugiyama. Dialogue breakdown detection based on estimating appropriateness of topic transition. In *Proceedings of Dialog System Technology Challenge 6 Workshop*, December 2017.

[105] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machile Learning Journal*, Vol. 63, pp. 3–42, April 2006.

[106] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, January 2007.

[107] Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. Fatal or not? finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2243–2248. Association for Computational Linguistics, September 2015.

[108] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceesings of the International Conference on Learning Representations*, May 2013.

[109] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, October 2014.

[110] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN:

Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association of Computational Linguistics*, Vol. 4, pp. 259–272, 2016.

[111] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1298–1307, August 2016.

[112] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 2124–2133, August 2016.

[113] Leo Breiman. Bagging predictors. *Machine Learning*, Vol. 24(2), p. 123–140, August 1996.

[114] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 583–593, July 2011.

[115] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing*, pp. 230–237, July 2004.

[116] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems in The Annual Conference on Neural Information Processing Systems*, 2015.

[117] John Lafferty, Andrew Mccallum, Fernando C.N. Pereira, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, pp. 282–289, July 2001.

[118] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proceesings of the International Conference on Learning Representations*, April 2020.