

ユーザとの対話による発話・応答データ収集機構の検討

五十川真生[†], 近井厚三[‡], 荒瀬由紀[‡],

[†] 大阪大学工学部電子情報工学科, [‡] 大阪大学大学院情報科学研究科

{isogawa.mao, chikai.kozo, arase}@ist.osaka-u.ac.jp

1 はじめに

りんな^{*1} に代表される人間とコンピュータが雑談するような非タスク指向型対話システムが登場している。しかし不自然な応答により対話がすぐに破綻してしまう [6], 定型的な発話が多く対話における楽しみに欠ける, 等の改善すべき課題は多い。

本論文では, 非タスク指向対話システムの実現に用いられる用例ベース対話システムに注目する。用例ベース対話システムとは, 発話に対する応答を予め収集し, そのデータを元にユーザの発話に対して応答するシステムである [4]。本論文ではこのような発話・応答からなるデータを対話データと呼ぶ。ユーザの新たな発話は必ずしも対話データ内にあるとは限らない。よってユーザが対話データにない発話をした場合, 対話データから類似した発話を検索し, その発話に対する応答を返すという機構が必要である。用例ベース対話システムの応答は, 類似した対話の検索, 及び対話データの規模に依存する。類似した発話の検索手法は多数のアプローチが研究されているが, 公に利用できる対話データの規模は限定的である。

そこで本研究は対話データを対話システムのユーザからオンラインで収集することでデータの量および質を向上することを目的とする。具体的には, シンプルな用例ベースの対話システムを試作し, 対話データに無い発話がユーザから入力された場合には, その発話を他のユーザにシステムが提示することでユーザからの応答を得る。発話と得られた応答を対話データに保存することを繰り返すことで対話データを充実させていく。さらに, ユーザとの対話におけるユーザの反応を収集することでどの応答がより適切と期待できるかをスコアとして保持することで, より良い応答を選別する際に利用できるようにする。

本システムを研究室内で 11 日間運用し, 14 名の学生から対話データを収集した。ユーザとの対話によって対話データを収集した場合, 収集しない場合それぞれ

で同様の用例ベース対話システムを構築し, ユーザに使用してもらいアンケート調査を行った。その結果, シンプルな用例ベースの対話システムにおいてもユーザから収集した対話データを用いることで応答の自然さやユーモア性が向上することを確認した。

2 関連研究

過去の対話データを用いて対話を実現する用例ベースの対話システムに関する研究はさかに行われており [3], 性能を競うコンペティションも開催されている [4]。対話データの収集については, 柴田ら [5] は WWW 上の文章から応答を取得するシステムを提案している。東中ら [2] は発話と応答のペアを用い, 隠れマルコフモデルにより対話をモデル化している。発話 A とその応答の発話 B, 発話 B' とその応答の発話 C があるとき, 発話 B, B' の内容が類似しているとする, $A \rightarrow \{B, B'\} \rightarrow C$ として対話履歴をモデル化できる。本研究ではこの考え方を用いて, 収集したユーザとシステムとの対話履歴から対話データを作成する。

別所ら [1] は対話データに無い発話の応答をクラウドソーシングを用いて取得するシステムを提案している。ユーザ A が対話データにない発話 A をシステムに入力したとき, システムが別のユーザ B に発話 A を発話し, その応答をユーザ A に発話するものである。その結果, 対話の面白さが増加するが, システムの応答は不自然になる傾向であることが示されている。本論文で提案するシステムは, 対話データにない発話 A に対する応答の取得方法は共通しているが, 応答の即時性を考慮しユーザ B の応答を待つことなく, 現状所持している対話データからユーザ A に応答をする点や, 応答だけでなくユーザの反応に基づいて応答をスコアづけする点で別所らのシステムと異なる。

^{*1}<http://rinna.jp/>

3 システムの設計

3.1 システムの概要

本論文で試作するシンプルな用例ベース対話システムおよび対話データ収集機構を図1に示す。実装にはLINE^{*2}のMessaging API^{*3}を使用している。各コンポーネントの説明を以下に記す。

セッション管理コンポーネント ユーザとの対話の開始, 終了を判定し対話ログをユーザ情報に記録する。システム或いはユーザが最初に発話した際に対話が始まったとする。システムが発話してから15分以内にユーザが発話しない時, 対話は終了したとみなす。ユーザとの対話の終了時に, 対話ログから対話データを更新する。LINE API から取得するユーザ発話, 応答選択コンポーネントから取得するシステム応答を用いてユーザ情報を管理する。応答選択コンポーネントから対話データに無い発話を受理した場合, 対話を開始していないユーザにLINE API を通じてその発話を送り, 対話を開始する。

ユーザ情報 LINEにおいて「友達」として登録しているユーザ情報, また現在行われているユーザごとの対話のログ, ユーザの発話ログ, システムの発話ログを記録する。

応答選択コンポーネント セッション管理コンポーネントから受理したユーザの発話に対する応答を対話データ, ユーザ情報から決定し, 応答する。応答の選択手法は3.4節で述べる。決定した応答を対話管理コンポーネントに通知する。

3.2 シードとなる対話データのマイニング

ユーザとの対話から対話データを収集するためには, システムが対話可能でなくてはならない。そのためには予め対話データを準備しておく必要がある。本研究はTwitter^{*4}上から対話データを収集する。収集のためにTwitterが公開しているREST API^{*5}及びStreaming API^{*6}を使用する。具体的な方法を以下に説明する。

まずStreaming APIを用いて, Twitterの公式タイム

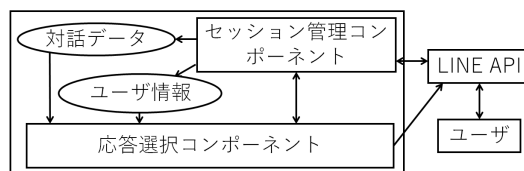


図1: 対話収集システムの模式図

ラインからユーザIDを取得する。次に, REST APIとそのユーザIDを用いて, 各ユーザが発話したツイートの中で別のツイートに対して返信しているツイート T_0 を検索する。REST APIから, T_0 の返信元であるツイート T_1 を取得する。再びREST APIから T_1 の返信元であるツイート T_2 を取得する...という処理を対話の始まりであるツイート, あるいは返信先ツイートが特定不能^{*7}なツイート T_N にたどり着くまで繰り返す。 T_0, \dots, T_N を一連の対話(本論文では対話シーケンスと呼ぶ)として保存する。このようにして2016年6月4日から12月20日までツイートをクロールし, 約500万のツイートと, 約120万の対話シーケンスを収集した。対話シーケンスの平均長は4ツイートであった。対話データには3名以上のグループで対話しているものも含まれているが, ユーザの区別はせず, 対話と応答の系列のみにもとづいてシーケンスを決定する。

3.3 対話データ構造

対話データの構築においては, 東中らのモデル[2]に基づき, 対話シーケンスを保持できる利点のある有向グラフを用いる。形態素解析器MeCab^{*8}を用いて発話から名詞, 動詞, 形容詞, 形容動詞, 副詞, 連体詞, 感動詞, 助動詞(です, ます, だ, で, ある, は例外)を品詞とする形態素を抽出し, 抽出した形態素のセットが完全一致する発話を同一の発話とみなす。東中らはエッジに対話の遷移確率を重みとして付与しているが, 本システムではユーザの反応に基づくスコアと対話シーケンスの識別子をラベルとして付与する。付与したラベルはある発話に対する応答が複数対話データ内にある際に, どの応答を選択するかを決定するために使用する。

ユーザが応答した回数 ユーザがある発話Aに対して発話Bを応答した回数をA→Bのエッジに付与する。

^{*2}<https://line.me/ja/>

^{*3}<https://business.line.me/ja/services/bot>

^{*4}<https://twitter.com/>

^{*5}<https://dev.twitter.com/rest/public>

^{*6}<https://dev.twitter.com/streaming/overview>

^{*7}鍵アカウントやツイート情報・ユーザ情報が削除されていることが原因である

^{*8}<http://mecab.sourceforge.net/>

応答率 システムがある発話 A に対して発話 B を応答として選択した時、ユーザから何らかの応答 C が得られた割合を $A \rightarrow B$ のエッジに付与する。応答率はユーザが応答した回数を R 、ユーザが応答せず無視した回数を I とすると、

$$ResRate = \frac{R}{R+I} \quad (1)$$

と計算する。ユーザとの対話によって応答率を逐次更新するため、 R 及び I をエッジに記録しそれをもとに応答率を計算する。

対話のシーケンス 対話の流れを情報として保持できるよう、対話のシーケンスをエッジにラベルとして付与する。図 2 において、「漫画読んだ」という発話 A に対し、「どうだった?」という発話 B が応答として選択され、その応答として「面白かった」という発話 C が選択される。また、「晩御飯食べた」という発話 D に対し、発話 B が応答として選択され、その応答として「おいしかった」という発話 E が選択される。これら 2 つの異なる対話が発話 B において交差するが、それぞれのシーケンスを再現できるよう、図 2 に示すように識別子をラベルとして付与する。発話 B の応答を選択する時、直前の発話が A なら C を、D なら E を選択する必要がある。直前の発話・応答間のエッジに付与されたラベルと同一のラベルがついたものを応答することで、対話の流れにそった発話を選択できる。

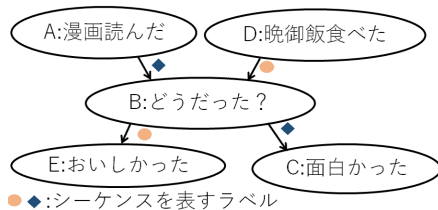


図 2: 対話シーケンスの保存

3.4 応答の選択

応答選択コンポーネントでは以下の手法によりユーザの発話に対する応答を選択する。ユーザの発話と同一の発話 (3.3 節) が対話データ内に存在する場合は、ユーザが応答を返す確率の高い発話を応答として選択することでユーザからより多くの対話データを収集できると期待できる。そこで応答率 (式 (1)) が最も高い

ものを応答候補とする。応答率が同値のものが複数ある場合は、ユーザによる発話回数が多いものを選択する。いずれの値も同値のものがある場合は、ランダムに一つを選択する。ただし、「どうだった?」等の対話の文脈に依存する発話の場合は、保存したシーケンスのラベルを優先する。本システムは、7 文字以下または代名詞以外の名詞を持たない発話は対話の文脈に依存すると仮定する。直前に行われた応答のシーケンスを表すラベルと同じ値のシーケンスを表すラベルを持つ応答を選択する。

ユーザの発話と同一の発話に対話データに含まれていない場合は、その発話に対する応答をデータとして収集するため、セッション管理コンポーネントを通じて他のユーザに提示する。一方で、現在のユーザに迅速に応答するため、現状の対話データから最も類似した発話を選択し、それに付随する応答を返す。類似度の判定のため、まず発話から 3.3 節同様品詞を抽出し、抽出した品詞からその形態素が存在すれば 1、存在しなければ 0 として発話の特徴づけるベクトルを作成する。ここで、発話 A から作成したベクトルを v_a 、発話 B から作成したベクトルを v_b とし、コサイン類似度を使用して文発話 AB 間の類似度を計算する。

$$Sim(A, B) = \frac{v_a \cdot v_b}{|v_a| \cdot |v_b|} \quad (2)$$

4 システムの運用実験

4.1 実験設定

システムは Python と PHP を用いて実装し LINE Messaging API を使用しユーザの LINE に発話をメッセージとして送信する。研究室に所属する 20 代の学生 14 名に 11 日間対話システムと対話してもらい、対話データを収集した。その後、収集した対話データを用いたシステムと、Twitter の対話データのみを用いた対話システムを 11 名の学生 (内 9 名はデータ収集にも参加していた) に使用してもらった。参加者にはどちらが収集した対話データを利用したシステムか分からないよう、匿名化している。それぞれのシステムを 11 日間使用した後、対話システムの応答について、以下の観点について 5 段階で評価してもらった。

- 応答の自然さ
- 応答のユーモア性
- 対話の楽しさ
- 対話の不快さの度合い

4.2 結果

11日間の運用でユーザから合計876個の発話があり、新たな発話に対する応答として625個の発話を収集した。対話データは随時更新しており、運用期間内のシステムの応答1021個を半分に分け、運用期間の前半・後半それぞれでシステム発話の後にユーザが応答を返さなかった割合である無視率を求めた(図3)。Studentのt検定(両側検定)を実施し、 $p = 0.029$ であり、後半の応答は前半の応答と比較して無視率が有意に下がることが示された。このことから、対話データを随時収集、更新していくことで対話システムの応答の質が改善され、ユーザが対話を続ける割合が高まることで将来的にデータ収集量が増加することが期待できる。

収集したデータを用いた対話システム、Twitterからマイニングした対話データ(3.2節)のみ用いた対話システムそれぞれを利用してもらった後実施したアンケート調査の結果を図4に示す。Twitterデータのみで対話システムを作成した場合と比較して、応答の自然さ、ユーモア性、対話の楽しさともに改善していることが分かる。別所らの実験結果ではユーモア性と応答の自然さはトレードオフの関係にあることが示されたが、本実験の結果ではそのような傾向は見られなかった。実験が短期間であった影響も考えられるため、今後長期間システムを運用し検証する予定である。

5 まとめ

大規模な対話データ収集を目的とし、ユーザからオンラインで対話データを獲得するシステムを試作し、運用した。11日間という短期間の運用であったが、14名のユーザから新たに625個の対話データを獲得した。ユーザから収集した対話データを用いることで、シンプルな用例ベースの対話システムにおいても、応答の自然さやユーモア性が改善し、ユーザからの応答率も高まる傾向があることが示された。

今後の課題として、多人数が利用できるようなシステムを頑健にし、一般公開して大規模な対話データ収集に取り組む。また収集したデータは研究利用のため公開する予定である。

参考文献

[1] F. Bessho, T. Harada, and Y. Kuniyoshi. Dialog system using real-time crowdsourcing and

	無視率*	誤差範囲
前半	0.390	± 0.042
後半	0.325	± 0.041

*: $p < 0.05$

図3: 前半と後半の無視率の平均

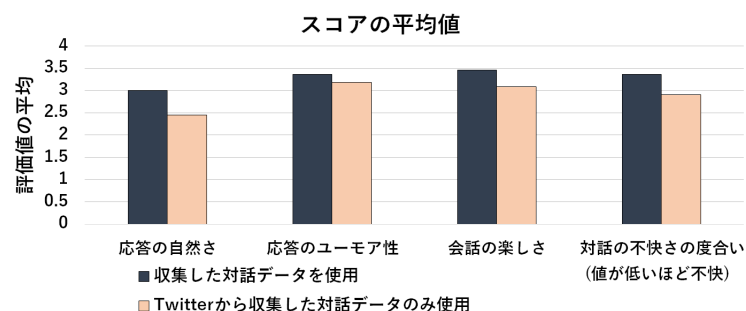


図4: アンケート調査の結果

Twitter large-scale corpus. In *Proceedings of SIGDIAL*, pp. 227–231, September 2012.

- [2] R. Higashinaka, N. Kawamae, K. Sadamitsu, Y. Minami, T. Meguro, K. Dohsaka, and H. Inagaki. Building a conversational model from two-tweets. In *Proceedings of ASRU*, pp. 330–335, December 2011.
- [3] H. Murao, N. Kawaguchi, S. Matsubara, Y. Yamaguchi, and Y. Inagaki. Example-based spoken dialogue system using WOZ system log. In *Proceedings of SIGDIAL*, pp. 140–148, September 2003.
- [4] L. Shang, T. Sakai, Z. Lu, H. Li, R. Higashinaka, and Y. Miyao. Overview of the NTCIR-12 short text conversation task. *Proceedings of NTCIR-12*, pp. 473–484, June 2016.
- [5] 柴田雅博, 富浦洋一, 西口友美. 雑談自由対話を実現するためのwww上の文書からの妥当な候補文選択手法. 人工知能学会, Vol. 24, No. 6, pp. 507–519, September 2009.
- [6] 東中竜一郎, 船越孝太郎, 稲葉通将, 荒瀬由紀, 角森唯子. 対話破綻検出チャレンジ2. 人工知能学会, Vol. B5, No. 02, pp. 64–69, October 2016.