

ユーザの発話履歴を考慮した会話文の自動生成

西尾 友佑 堀場 美那 宮本 善成 村井 翼 韓 東力

日本大学文理学部 情報システム解析学科

1. はじめに

近年、様々な場面において人間と円滑にコミュニケーションを取ることを目的とされている非タスク指向型会話システムへの期待が高まっている。しかし、ユーザの興味・関心に対応し、かつ、人間らしい柔軟な会話文をユーザに提供することは難しい。

昨年、本研究室では Twitter[1]をデータソースとして Web 上から抽出した辞書を用いて、マルコフ連鎖の特徴である「確率による形態素間の繋がり」という概念を用いて応答文を生成する研究が行われた[2]。テンプレート化されていない柔軟な会話文の生成には成功したが、会話文間の前後関係を考慮していないため、話題の流れが断ち切られてしまう事例も数多く見られた。また、一つのキーワードから確率のみを指標として次の形態素を選定していたため、一つの会話文において日常生活で散見される会話文とは異なる形態素間の繋がりが生じ、一つの会話文中の焦点がぼやけてしまう事例がたくさん見られた。

そこで本研究では、直近の会話との繋がりを考慮し、そのうえ焦点の定まった会話文を生成することに重点を置いた会話システムの研究開発を行った。本システムでは、直近の複数のユーザの発話文に対して「窓」という概念を適用することによって、直近のユーザの発話文中の名詞・動詞・形容詞に重みを付与した後、「窓」内における「相互情報量」の概念を用いて単語間の関連度を求め、関連度の高い単語対を抽出することにより直近の会話の繋がりを考慮した。また、生成した単語対のそれぞれを始点と終点とする断片を含む Tweet を Twitter から検索し、そして始点から前へ、終点から後ろへマルコフ連鎖の考え方をもとに文生成を行うことによって、会話文中の焦点をぼやけさせないだけでなく、特徴のある会話文の生成を試みた。

2. システム全体の処理フロー

本研究で構築したシステム全体の流れを図 1 と図 2 に示す。図 1 はユーザの発話文数が窓の範囲未満の時を、図 2 は発話文数が窓の大きさ以上の時を表している。詳細は、次章以降で述べる。

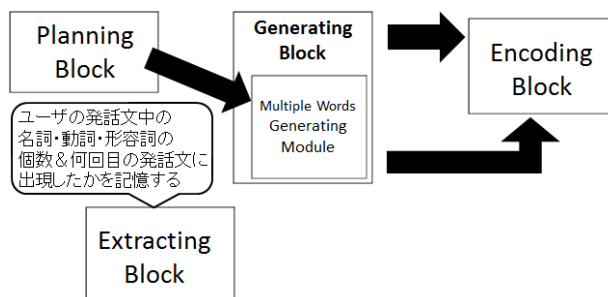


図1 ユーザの発話文数が窓の範囲未満の時の
処理フロー

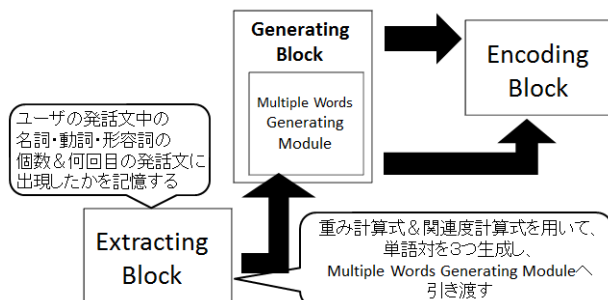


図2 ユーザの発話文数が窓の範囲以上の時の
処理フロー

昨年に関連された会話システムでは、まず Planning Block でユーザの入力文に対して返答と成り得る関連度の高い単語(以下、「キーワード」と呼ぶ)を Web 検索を用いて抽出する。次に、Generating block で文生成を行う。文頭から文末にかけての形態素の並びを記録した順方向マルコフ連鎖辞書と、文末から文頭にかけての形態素の並びを記録した逆方向マルコフ連鎖辞書、「キーワード」と成り得る「名詞、動詞、形容詞、副詞」の前後に出現する形態素を記録したキーワード想起双方向性マルコフ連鎖辞書をユーザの入力文より作成し、それらの辞書を用いて、「キーワード」を起点に、順方向と逆方向のマルコフ連鎖をそれぞれ文末と文頭に向かって行い、会話文を生成する。最後に Encoding Block で会話文に対してキャラクター性の付与を行う。本研究では、先述した問題点を解決することを最優先課題と位置づけ、話題の繋がりを意識した会話を行うために Extracting Block を、焦点の定まった会話文を作成するために Multiple Words Generating Module を新たに

開発することを目指した。

3. 話題の繋がり

この章では、会話の繋がりを考慮した会話文生成を行うための Extracting Block について述べる。

昨年度の研究において、双方向型マルコフ連鎖辞書を用いることにより、柔軟な会話文の生成には成功した。しかし、ユーザの興味を断続的には考慮しておらず、会話文一つを生成する度に、Planning Block にて逐次「キーワード」を抽出しているため、会話文間の繋がりを一切考慮していないという問題点が存在している。

そこで、我々は「窓」の概念を導入し、ユーザが直近に発した複数の発話文を「窓」という枠でくくることにする。「窓」内におけるユーザの発話文中に含まれる単語に関しては、ユーザの興味という概念を導入し、名詞・動詞・形容詞に重みを付与した後、「窓」内における単語間の関連度計算を行い、関連度の高い単語対を生成する。重みの計算に関しては 3.1 節から 3.3 節にて、関連度計算に関しては 3.4 節にて詳しく説明する。

3.1. 窓の概念を導入する理由

ユーザの総発話文数が少ない時、Planning Block を利用し、発話から応答文生成にふさわしい単語を抽出することで話題を膨らませることが望ましいが、総発話文数が一定量を超えた時、過去の発話全てを考慮してキーワードを探すことはシステムに負荷が掛かる上、あまり古い発話文を利用して単語を抽出しても話題が転換している可能性が高い。よって、古すぎる発話文を利用することは応答文生成にふさわしくない単語を抽出してしまうことがある。

我々はこの問題の解決に向けて、黒橋らの研究[3]に注目した。この研究では、あるキーワードの出現密度分布をテキスト内で調べ、そのキーワードの重要説明箇所を特定するという手法を提示しているが、その出現密度分布を調べる際に高密度な出現位置を取り出す為、「窓の幅」というものを考慮してキーワードに重み付けを行っている。ここで言う「窓の幅」とは重み付けを行う範囲のことであり、キーワードの検索範囲を示す。我々の研究ではこの「窓の幅」を参考にし、「窓」の概念を取り入れることにした。利点としては、図3で示すようにユーザの発話文履歴を“窓”という枠でくくることにより、ユーザの興味を精度高く追跡することが可能になる点や、窓の大きさを可変長にすること

で、発話文量の多さによって場合分けができる点などが挙げられる。

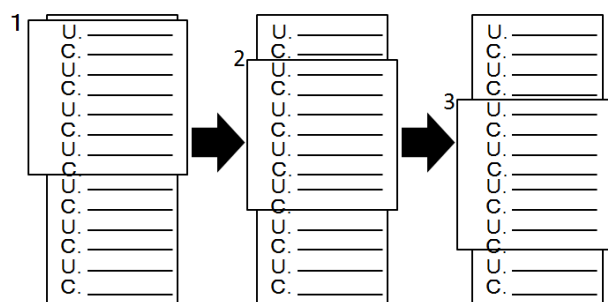


図3 「窓」のイメージ

また、会話文の内容・応答のタイミングの両面においても、以下の様な性能向上が期待出来る。会話文の内容面においては、窓内における全てのユーザの発話文の内容を考慮して応答文を生成することが可能となる。応答のタイミングの面においても、ユーザの発話文数が窓の範囲未満の時に実行時間の長い Planning Block が起動されるので、初対面の状態から時間が経っていない人間同士が会話をする時に新しい話題を考えながら話題を広げようとするイメージと偶然にも合致する。一方、初対面の状態から時間が経った人間同士が会話をする時は、会話の速度も上昇し、内容も固定化されてくるといった、日常生活において散見される会話の状況をユーザに提供することが可能となるからである。

3.2. ユーザの興味

我々が会話を行う際、自分の興味のある話に移り変わった時、自分側が長く話したり、その話題に関連のある単語を繰り返して使うことがある。この節では会話に参加しているユーザが何に興味を持っているか、そしてその興味はどのような形で会話に現れているのかに焦点を絞る。そこで我々は堀内らの研究[4]を参考し、ユーザ興味の概念とそれに基づく興味認識項と時間減衰項という二つの考え方を導入している。

興味認識項とは、ユーザの発話が長い文であればあるほど、その文に含まれる名詞に重みを付与するという考え方である。ユーザが長い文を発話するほど、その発話中にある名詞にユーザが興味を持っていると考えられるからである。逆にユーザの発話に含まれなかったキーワードに対しては、興味が小さいと考えて重みを減らしていく。また、時間減衰項とは、過去の発話より最近の発話にあったキーワードほどユーザの興味を示していると考え、過去に出たキーワ

ードのすべてに対しては重みを減らしていくという考え方である。これらの考え方をもとに、窓内にあるキーワード間の関連度をユーザの興味を考慮しながら数値化する仕組みを 3.3 節に示されるように考案した。

ユーザの興味の概念をシステムに導入することで、システム自体が会話の流れを汲みとりながら発話を行うことを目標としている。

3.3. 窓内におけるキーワードへの重み付け

窓内にあるユーザの発話文中に含まれるキーワード(名詞・動詞・形容詞)に対して、3.2 節で述べられた考え方をもとに、話題の展開に対する影響度に応じた重みを付与する。窓内にあるキーワードの異なり数を j と定義し、 W_j が j 番目の単語を表す。窓の範囲(文数・可変長)を l 、窓内に出現する発話文の順番を i と定義する。また、 C_{ij} とは i 行目のキーワード W_j の出現回数であり、 S_{ij} とは、 i 行目における単語 W_j の総得点であり、 T_j とは窓内における W_j の総得点である。

$$S_{ij} = \frac{l-i}{l} \times C_{ij}$$

$$T_j = \sum_{i=0}^{l-1} S_{ij}$$

以上の式を適用し、Extracting block に格納された名詞・動詞・形容詞に対して、重み付けを行う。

3.4. キーワード間の関連度

我々は、関連度の高いキーワード対を窓内から抽出し、Multiple Words generating Module へと引き渡すことによって、焦点の定まった会話文生成を行おうと考えた。単語関連度は、相互情報量の概念[5]をもとに考案している。以下に、関連度計算式と計算に用いられた変数の順に記述する。

$$SR_c(W_a; W_b) = \frac{\sum_{i=1}^{f(W_a)} \sum_{j=1}^{f(W_b)} \left(\frac{1}{|N_i(W_a) - N_j(W_b)| + 1} \right)}{f(W_a)f(W_b)}$$

$f(W_a)$: 窓内における W_a の出現回数

$f(W_b)$: 窓内における W_b の出現回数

W_a と W_b のペア数

$N_i(W_a)$: i 個目にカウントした W_a が出現している文番号

$N_j(W_b)$: j 個目にカウントした W_b が

出現している文番号

c : 窓内に存在する W_a と W_b のペア数

3.5. キーワード対の選定

窓内に存在する全ての名詞と動詞のペアおよび名詞と形容詞のペアに 3.4 節で述べた関連度計算を適用し、上位 3 つのキーワード対を Multiple Words Generating Module へと引き渡す。その際に名詞の重みが上位にあるペアを最優先とする。また、同一の名詞を使っているペア同士の場合は、動詞を用いているペアが形容詞を用いているペアよりも優先されることとする。その他の場合においても、重み計算で求められた重みを同率時の優先規則として採用する。

4. 焦点の定まった会話文生成

この章では、焦点の定まった会話文生成を行うための Multiple Words Generating Module の機能について述べる。先行研究では、一つのキーワードから双方向にマルコフ連鎖を用いて、文章を作成していたが、生成した会話文において、日常生活で散見される会話文とは異なる態素間の繋がりや、一つの会話文中の焦点がぼやけてしまうような事例が数多く見られた。そこで我々は、第 3 章で話題の繋がりを考慮して選定されたキーワード対を検索クエリとして抽出した Tweet をデータソースとし、さらにマルコフ連鎖を用いることによって、焦点の定まった会話文の生成を行った。

4.1. 会話文生成

まず、第 3 章において、選定されたキーワード対 3 つのうち、最高得点を保有するキーワード対にある名詞を W_{ord1} (以下、 $W1$ と呼ぶ)、もう一方を W_{ord2} (以下、 $W2$ と呼ぶ) と定義する。定義した $W1 \cdot W2$ を検索クエリとして、 $W1 \rightarrow W2$ の順で 2 単語が共に出現する Tweet 群を Twitter API[6] の検索機能を用いて検索する。この処理で抽出された Tweet 群をデータソースにして、キーワード想起双方向性マルコフ連鎖辞書(以下、Twitter マルコフ辞書)を作成する。次に、抽出した Tweet 群から、先端が $W1$ 、終端が $W2$ となるように $W1$ 以前、 $W2$ 以後の内容を削除し、断片を生成する。最後に生成した断片から、 $W1$ 以前では逆方向マルコフ連鎖辞書を用いて、 $W2$ 以後では順方向マルコフ連鎖辞書を用いて、断片前後の内容を追加し、会話文を生成する。なお、ここまで説明した一連の処理に関してまとめた図を、図 4 に示す。

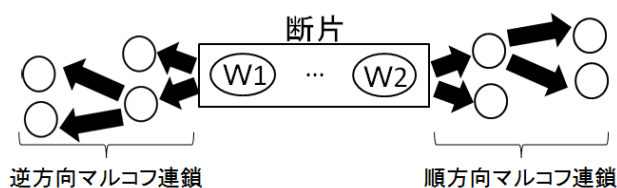


図4 Multiple Words Generating Module
の処理図

なお、Tweet 群の中から、最新の Tweet の断片に対して前述する文頭・文末へのマルコフ連鎖を行う。その理由は、最新の Tweet は、断片中の内容とユーザが意図した内容と一致している可能性が高いことが考えられるためである。また、逆方向マルコフ連鎖辞書および順方向マルコフ連鎖辞書を用いて、断片以前・以後の部分の会話を生成する理由としては、マルコフ連鎖の概念を導入することにより、より柔軟な発話文生成を行うためである。また、生成した Twitter マルコフ辞書内に、断片以前、以後の内容も 2-gram データとして保管されているため、断片の内容から発散しすぎることなく、会話を生成することも可能となるということの 2 点が挙げられる。

4. 2. 例外処理

但し、最高得点を保有するキーワード対を AND 検索した結果、Tweet 群が抽出できなかった場合には、2 番目、そして 3 番目の順に得点の高いキーワード対を用いて Tweet 群の抽出を行う。また、万が一 3 つのキーワード対全ての抽出結果が 0 件であった場合は、昨年の研究で作成した Generating Block 内のモジュールを起動し、会話文生成を行う。その時は、最高得点を保持する単語対の名詞を用いて、会話文生成を行うこととする。

また、会話をしているという感覚をユーザに提供するために、「おはよう」という入力文に対して「おはよう」という応答、「ありがとう」という入力文に対して「どういたしまして」といった応答に代表される、日常生活において頻繁に散見される定型表現に対する定型的な応答文の生成を行った。定型文自身には、名詞・動詞・形容詞のいずれも含んでいないことがほとんどであり、また、会話文の内容も応答文生成に必要な場合がほとんど見られないという観点から、入力文が定型文と一致した際には、その入力文を先述した「窓」の範囲から除外することとした。

5. おわりに

本研究では、直近の会話との繋がりを考慮し、そのうえ焦点の定まった会話文を生成することに重点を置いた会話システムの研究開発を試みた。

しかし、会話文生成に時間がかかることや、生成された会話文にある断片とそれ以外の部分との間の相関関係が安定しないなどの問題が挙げられている。現在のシステムでは一つの会話文を生成する度に Twitter マルコフ辞書を逐次生成する方式を取っているが、今後は Twitter へ常時接続し、システムダウンのときもマルコフ辞書を継続的に更新できるような状態で会話文生成を行うことにより、実行時間のかかる Web 検索を行う回数を最大限に減少させる。また、会話文にある断片とそれ以外の部分との間に意味的な相関関係が安定しないことについては、現在はマルコフ連鎖による確率のみを指標として会話文生成を行っているが、マルコフ連鎖以外の概念や規則を新たに導入することにより、さらに人間らしい会話文の生成を目指したい。

システムの実装時間が予定より大幅に伸びてしまったため、現在のところはまだ評価実験を行えていない。今後は、「窓の幅」を変えながら本研究で定義された窓内における単語間関連度の有効性を評価実験を通して検証していく。また、本研究の目標とされている会話の繋がりや向上と焦点の定まる会話文の生成についても本研究に関与していない第三者を対象にアンケートを実施する予定である。

参考文献

1. <http://twitter.com/>
2. 木下 翔央, 福地 隆, 幸崎 剣, 韓 東力 “Twitter のリプライ文と役割語変換を利用した会話文の生成” 情報処理学会第 73 回全国大会, デ-4, pp. 2-553-2-554, 2011.
3. 黒橋 禎夫, 白木 伸征, 長尾 眞 “出現密度分布を用いた語の重要説明箇所の特定” 情報処理学会研究報告. 自然言語処理研究会報告 96(87), 43-50, 1996.
4. 堀内 理沙, 上田 祐大, 原田 孝太, 韓 東力 “雑談システムにおける話題転換” 言語処理学会第 15 回年次大会, B2-8, pp. 312-315, 2009.
5. 北 研二『言語と計算 4 確率的言語モデル』東京大学出版会, 1999.
6. <http://usy.jp/twitter/index.php?Twitter%20API>