

レッテルを用いた雑談向けチャットボット「Lamb」の提案

杉本 隼斗¹ 中山 裕貴¹ 渡邊 一樹¹ 中園 歩² 濱川 礼¹

概要: 本論文では、レッテルを冗談として表現する事により、ユーザの退屈感を軽減させる雑談向けチャットボット「Lamb(Labeling Amusement Bot)」について述べる。本論文において、レッテルとは「ある人物や物事をなんらかの一言・名称で言い表した名詞」のことである。雑談向けチャットボットとの会話では、ユーザに退屈感を与えないことや、会話の継続性を高めることが重要である。「Lamb」はユーザの入力に対し、深層学習を用いた応答文と、単語の分散表現を用いてレッテルを生成する。生成したレッテルが、ユーザに対し冗談となるかを判定し、判定されたレッテルと応答文を組み合わせ発話する事で、ユーザの退屈感を軽減し、会話の継続性の向上を試みた。

1. はじめに

近年、スマートフォンの普及や家庭用ロボットの開発により、人間がコンピュータと対話する機会が増え、雑談向けチャットボットの需要が高くなっている [1]。岩倉らによると、雑談向けチャットボットでは、自由な対話である雑談によってユーザを楽しませることを目的とし、このようなシステムにおいては、対話の継続性が重要とされる。しかし、対話が単調だと退屈感を感じ、飽きやすいという問題点がある [2]。この問題を解決するために、女子高生 AI りんな [3] のような雑談向けチャットボットに個性を与える研究や、ユーザへの応答文にユーモアを含ませる [4] などの研究が行われている。しかし、ブラックジョークや相手のことを茶化す冗談などを用いた研究は少ない。そこで我々は「レッテルを貼る」という行為に着目した。

「レッテルを貼る」とは『日本国語大辞典 第二版』にて「一方的にその人物の人格や能力などを格付けする」と定義されている [5]。また大津によると、親しい者同士の雑談では、悪口や批判などを用いて会話者同士に一時的な対立関係を「遊び」として構築することがあるとされる。この「遊び」として行われる一時的な対立関係の目的は、相手との親しい関係を維持し、会話を楽しむこととされている [6]。そのため、我々はレッテル貼りを冗談として発話するにより、ユーザの退屈感を軽減し、ユーザと雑談向けチャットボットの会話の継続性を高められると考えた。

しかし、この「遊び」としての対立関係を会話に用いる場合、レッテルが相手に冗談として伝わらず、相手を意図せず攻撃してしまうことがある。大津によると、冗談とは「真面目に言っているのではないことを前提に相手を面白がらせることを目的として発する発話」と定義されている [7]。つまり、相手がレッテル貼りを「真面目に言っている

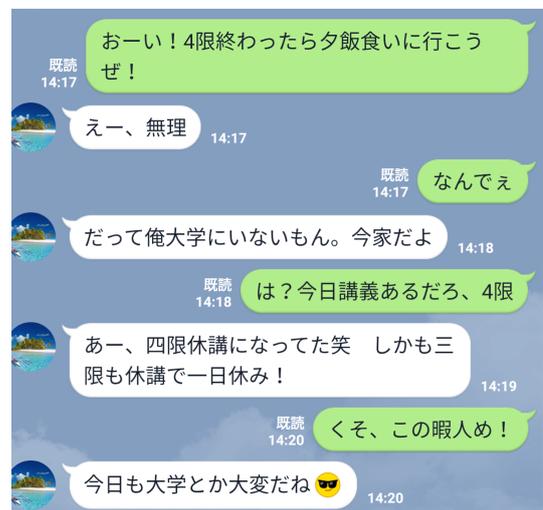


図 1 実際の会話例

る」と感じた場合、そのレッテル貼りが相手に対し誹謗中傷になると考えられる。また、対話者同士に十分な親密度がない場合、「遊び」としての対立関係を作ることができない。そのため、我々は雑談向けチャットボットとユーザの親密度と、レッテルのユーザに対する攻撃度を定義し、レッテルが誹謗中傷に当たるか判断できると考えた。レッテルがユーザに対し誹謗中傷の場合、レッテル貼りを行わないことにより、ユーザに対し誹謗中傷を行うことを回避する。例えば、親しい関係のコミュニケーションにおいて、冗談としてのレッテル貼りが行われている事例を図 1 に示す。図 1 の会話では、大学生 M が友人に対して夕飯を誘うことから会話が始まっている。一連の会話の中で、大学が一日中休みだった友人に対して、「暇人」というレッテルを貼っている。また、友人はレッテル貼りが行われていることに対し、怒っているようには見られない。このように、日常会話において、親しい関係性では、冗談としてレッテル貼りが行われていることがわかる。

これらのことから、雑談向けチャットボットの使用にお

¹ 中京大学 工学部 情報工学科

² 中京大学 工学研究科 情報工学専攻

いてユーザの感じる退屈感を軽減し、会話の継続性を向上させるにはレッテル貼りを冗談としておこなうことが有効と考え、応答文にレッテルを用いることでユーザに対しレッテル貼りを行う雑談向けチャットボット「Lamb」を開発した。

2. 関連研究

近年では、雑談向けチャットボットに様々な工夫を凝らし、チャットボットの応答文に面白さを持たせる研究が盛んに行われている。ここでは、類似した研究例を紹介する。

2.1 ボケて返す対話型エージェント

呉らは、ユーザの発言を聞き間違え、聞き返すボケなどを行うエージェントの研究を行っている [8]。ボケを生成させる方法として学術的観点と実践的観点の二つの方法が取られている。学術的観点では、心理学的モデルとして多く採用されている不適合一解決モデルを用い、ユーモアを生成させる。例えば呉らによると、「フォーマルな格好が似合う人を紹介してほしい」という問いに対し、「ペンギン」を紹介する。これは、人に対し動物を紹介しているため不適合を生じさせているが、ペンギンは人間のフォーマルな容姿に似ているため、フォーマルな格好という部分については解決している。このような返信をすることによりボケを表現している。実践的観点では、似た発音で出来るだけ意味に差がある単語を発することでユーモアを生成させる。例えば、「ペンギン」と発話するところを、「エンジン」と発話することでボケを表現している。

しかしながら、ボケといった遊戯的なユーモアよりも、ブラックジョークといった攻撃的なユーモアの方が、親しい間柄ならではの表現であるため、退屈感を軽減させるという目的において、より重要な表現であると考えられる。そこで、我々の「Lamb」では、レッテル貼りという攻撃的なユーモア表現を使用している。

2.2 SarcasmBot

Aditya らは、ユーザの発言に対し皮肉を返すチャットボットの研究を行っている [9]。皮肉を生成させる方法として、極性判定を用いることで感情を判定し、ネガティブな表現にポジティブな表現を加えることで生成させる。例えば、「A氏についてどう思いますか?」という発話に対し、「うーん、Aのことは好きだよ。責任感が全然ない人を私は本当に高く評価しているから」という応答がある。これは、「責任感が全然ない人」というのがネガティブで、「私は本当に高く評価している」がポジティブであるため皮肉を表現することができる。しかしながら、皮肉は受け手によっては誹謗中傷行為だと判断されることがある。そこで、我々の「Lamb」では、受け手が感じる誹謗中傷度合を推測することでこの問題の解決を試みた。

3. 提案手法

ユーザのチャットメッセージを入力として、レッテルを生成する手法 (3.1) と、生成したレッテルがユーザに対し

て誹謗中傷に当たるか判定する手法 (3.2) を提案する。

3.1 レッテルの生成手法

ユーザの発話に含まれる特徴からユーザに対する価値評価を表すレッテルを生成する手法について述べる。

3.1.1 レッテルの定義

本研究では笠井によって「性質、特徴、属性などを示す要素+人や物を示す要素」という構造を持つ名詞と定義されるレッテルを参考に行っている [10]。表1はそのレッテルの形式をまとめたものである。

表1 レッテルの形式

形式	特徴	例
NP	一語で「性質、特徴、属性などを示す要素+人や物を示す要素」を示している形式。	天才料理人、鬼教官
NP_1 の NP_2	一語では「性質、特徴、属性などを示す要素+人や物を示す要素」を十分に示せなかった時に、二語用いることで意味的に充足している形式。	異次元の暇人、ネットワークのエンジニア

NP 形式は複合名詞でレッテルを表現する傾向がある。複合名詞は名詞を結合することで無限に生成することが出来るが、生成規則が曖昧なため違和感のない複合名詞を生成することは難しいと考えられる。「指導者」は違和感のない複合名詞であるが、「指導人」は用例が少なく違和感のある複合名詞である。そこで本手法では複合名詞を扱う必要のない NP_1 の NP_2 形式を扱う。ユーザの発話を入力とし「性質、特徴、属性を示す要素」と「人や物を示す要素」を別々に取得し、それらを NP_1 の NP_2 形式に当てはめレッテルを生成する。 NP_1 に「性質、特徴、属性を示す要素」を割り当て、 NP_2 に「人や物を示す要素」を割り当てる。

3.1.2 「性質、特徴、属性を示す要素」の取得

入力文の「性質、特徴、属性を示す要素」を抽出し、それを言い表す単語に換言する。換言された単語がユーザに対する価値評価が表れた「性質、特徴、属性を示す要素」である。入力文から換言を行うアルゴリズムを Algorithm 1 に示す。

Algorithm 1 「性質、特徴、属性を示す要素」の換言処理

```

Input:  $text, W$ 
1:  $V \leftarrow \emptyset$ 
2:  $C \leftarrow analyze(text)$ 
3: for  $c_i \in C$  do
4:    $v_i \leftarrow get\_vector(W, c_i)$ 
5: end for
6:  $p \leftarrow sum(V)$ 
7:  $NP_1 \leftarrow arg\ max\ cos(W, p)$ 
Output:  $NP_1$ 

```

入力文 $text$ を形態素解析し体言、用言にあたる単語の集合 C を取得する。学習コーパス W を学習させた単語の分散表現のモデルから集合 C の要素の単語ベクトルを獲得し、それらの和 p を求める。単語の分散表現を用いて単語

をベクトル化することで、単語の意味情報を含む特徴量が取得でき、単語同士の演算が可能になる。算出したベクトルの和 p とコーパス内で最もコサイン類似度が高い単語ベクトルを持つ単語を取得する。

3.1.3 「人や物を示す要素」の取得

「人や物を示す要素」は「人や物」に関する単語を事前に収集し、それらの中の「性質、特徴、属性を示す要素」との共起性が高い単語とする。「人や物」に関する単語の収集方法は、任意の文章を形態素解析し、特定の名詞カテゴリに解析された単語の収集によって行う。ここでの名詞カテゴリは、単語の意味情報を付与したカテゴリであり、形態素解析システムに付随しているものを使用する。収集する単語の名詞カテゴリは、何を対象にレッテルを貼るかを考慮して決定する。本研究ではユーザに対してのみレッテルを貼ることを前提としているため、「物」に関しての名詞カテゴリを持つ単語は収集せず、「人」に関する名詞カテゴリを持つ単語のみを収集している。形態素解析による単語収集では、不必要な単語が含まれることがある。「人」に関する名詞カテゴリでは、人称代名詞が例に挙げられる。人称代名詞は、抽象的な名詞であり「人や物を示す要素」としての役割を十分に果たさない。このような不必要な単語はストップワードを用いて収集の際に除外する。

3.2 誹謗中傷判定

レッテルは、攻撃的な意味を持つ場合があるため、会話の相手に対するレッテルを用いた発言は、受け手に対し誹謗中傷となる可能性が考えられる。そのため、受け手にとってレッテルを冗談として扱えるのかどうか、検証をする必要がある。

大津によると、攻撃性のある発言を冗談として捉えるためには、受け手が発話者のメッセージに含まれる「これは遊びである」というメタメッセージを受け取る必要がある [6]。メタメッセージとは、発話者が発したメッセージに含まれる受け手への態度や気持ちなど、直接の言葉には含まれないメッセージである [12]。レッテル貼りの場合、このメタメッセージは「本気でそう思っていない」というメタメッセージに等しい。つまり、受け手に「本気でそう思っていない」というメタメッセージが伝わった場合、レッテル貼りは受け手にとって冗談として判断される。

我々は、このメタメッセージの伝わりやすさが、レッテルの示す属性や発話者との関係性によって、変化すると考えた。そのため、受け手がメタメッセージを受け取れるかどうかを判断するために、レッテルの持つ受け手に対する攻撃度と発話者に対する受け手の親密度を定義する。レッテルの攻撃度 $Attack$ とは、「レッテル貼りに含まれるメタメッセージの弱さ」の程度であり、この値が高いほど、受け手に対しメタメッセージが伝わりにくくなる。受け手の親密度 $Closeness$ とは、「その相手のレッテル貼りをどれだけ許容できるか」という程度であり、発話者に対し受け手のメタメッセージの受け取りやすさを表す。そのため、誹謗中傷の判定には以下の式を用いる。

$$Slander = \begin{cases} 0 & (Attack \leq Closeness) \\ 1 & (Attack > Closeness) \end{cases}$$

ここで、 $Slander$ は、発話者の用いるレッテルが受け手にとって誹謗中傷に当たるかどうかを表す。 $Slander = 0$ の時、レッテルが誹謗中傷には当たらないことを表し、1 の時は、誹謗中傷であることを表す。

3.2.1 攻撃度推定

受け手の人物の特徴とレッテル貼りに用いられるレッテルの示す特徴の類似度が高い場合、受け手は発話者が「本気でそう思っていない」と判断するのが難しく、レッテル貼りを冗談としてと捉えられない場合がある。そのため、レッテルの攻撃度を、受け手の人物の特徴とレッテル示す特徴の類似度とし、以下の式によって算出する。受け手の人物特性を、受け手の発言から得られるその人の特徴とし、その人物の特徴を表す単語を要素とした集合 C とする。単語の取得には形態素解析を用いる。集合 C の取得を受け手の発言毎に行い、受け手の人物特性である集合 C を要素とする集合を F とする。 n はそれぞれの集合に含まれる要素数を表す。図 2 の会話例の場合、受け手 l の最初の会話の人物特性 C_1 は、「食べる」、「過ぎる」、「太る」となり、その次の発言の人物特性 C_2 は、「今日」、「ステーキ」、「食べる」となる。

$$C = \{word_1, word_2, word_3, \dots, word_{n_c}\}$$

$$F = \{C_1, C_2, C_3, \dots, C_{n_f}\}$$

$$Attack = \frac{1}{n_f} \sum_{k=1}^{n_f} \cos(\vec{C}_k, \vec{L})$$

ここで、 \vec{C} は、受け手の人物特性のベクトルを表し、 \vec{L} は、発話者がレッテル貼りに用いるレッテルのベクトルを表す。レッテルの持つ攻撃度は、集合 F に含まれるそれぞれの人物特性のベクトルとレッテルのベクトルのコサイン類似度を算出し、それらの類似度の平均とする。

各人物特性のベクトルは以下の式により算出する。人物特性に含まれる単語を分散表現によりベクトル化し、それらのベクトルの平均を人物特性のベクトルとして扱う。

$$\vec{C} = \frac{1}{n_c} \sum_{k=1}^{n_c} \vec{word}_k$$

図 2 の会話例の場合、「食欲の化身!」という発言によってレッテル貼りが行われている。貼られたレッテルである「食欲の化身」と、それぞれ二つの人物特性との類似度を算出する。日本語 Wikipedia を学習コーパスとした FastText [14] の学習済みモデルを使用して分散表現を獲得した場合、 C_1 とのレッテルの類似度は 0.52 となり、 C_2 とのレッテルの類似度は 0.43 となるため、この時点における受け手 l に対する「食欲の化身」というレッテルの持つ攻撃度は、0.475 となる。

3.2.2 親密度推定

レッテル貼りを行う発話者 s に対する受け手 l の親密度を、発話者の交友姿勢に対する受け手の交友姿勢の割合と定義し、以下の式により算出する。

$$Closeness = \frac{P_{ls}}{P_{sl}}$$

ここで、 P_{sl} とは、発話者の受け手に対する交友姿勢を表

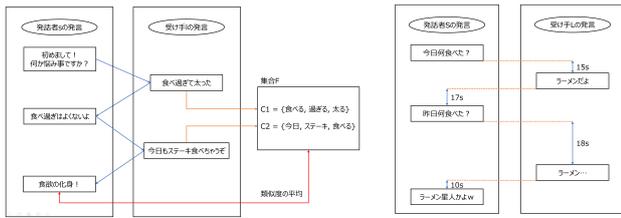


図 2 会話例 1

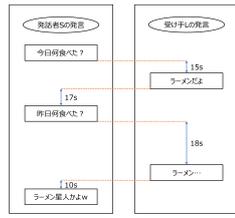


図 3 会話例 2

し, P_{ls} は, 受け手の発話者に対する交友姿勢を表す. 受け手の発話者に対する交友姿勢が高いとき, 受け手の発話者に対する親密度が高くなり, 発話者にとって受け手に対しレッテルを貼りやすい状況といえる.

交友姿勢とは, チャット使用時における会話者同士の親密度を推定する指標であり, 小寺らの研究によると, 「交友姿勢とは, 交友を深めようとする姿勢であり, 相手に対する好感度に類似した指標」とされ, 相手の発話を受けてから返答するまでの時間を用いて算出している [13]. 従って, 本研究においても交友姿勢を同様に定義し, 以下の式によって算出する.

$$P_{ij} = 1 - \frac{R-time_{ij}}{R-time_{ij} + R-time_{ji}}$$

$$= \frac{R-time_{ji}}{R-time_{ij} + R-time_{ji}}$$

ここで, P_{ij} とは, 会話者 i の会話者 j に対する交友姿勢を表す. $R-time_{ij}$ とは, 親密度を計測する時点までの会話者 i が会話者 j の発話への返答に要した時間の合計を表し, $R-time_{ji}$ とは, 会話者 j が会話者 i の発話への返答に要した時間の合計を表す. 返答に要する時間が短い時, 交友姿勢が高くなるようになっている.

図 3 の会話例では, 発話者 s の $R-time_{sl}$ は 27, 受け手 l の $R-time_{ls}$ は 33 となる. 発話者 s の P_{sl} は 0.55, 受け手 l の P_{ls} は 0.45 となり, 最後の発話時点での発話者に対する受け手の親密度は 0.82 となる.

4. システム実装

システムの流れを図 4 に示す. 「Lamb」はユーザの入力に対し, レッテル貼りを行う雑談向けチャットボットアプリケーションである. システム内部では, ユーザの入力メッセージからレッテルを生成し, それがユーザにとって誹謗中傷にあたるかを判定している. 誹謗中傷に判定された場合は深層学習を用いて生成した応答文のみを用いてレッテル貼りはせず, 誹謗中傷に判定されなかった場合に, 生成したレッテルと応答文を用いてレッテル貼りをする. これらの機能を実装した GUI アプリケーションの開発を行なった.

4.1 レッテル生成

「Lamb」では「性質, 特徴, 属性を示す要素」を取得する際の単語の分散表現に, 日本語 Wikipedia を学習コーパスとした FastText[14] の学習済みモデルを使用している.

「人や物を示す要素」を取得する際の形態素解析には

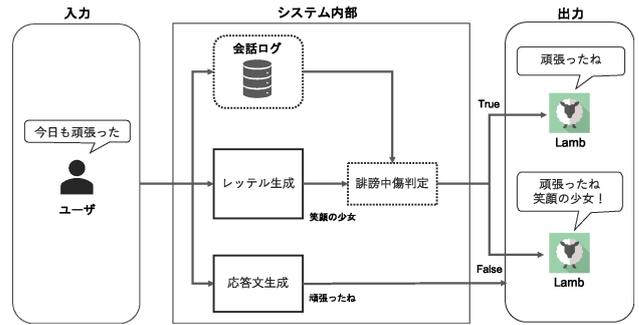


図 4 システムの流れ

JUMAN++[15] を使用している. 収集する単語の名詞カテゴリには「人」に関する名詞カテゴリとして『人』と『動物』を設定した. これは, 「Lamb」がユーザである人に対してのみレッテルを貼ることと, 日常会話における人に対するレッテル貼りで「ぼんこつタヌキ」「負け犬」など, 人だけでなく動物に形容されることがあることを考慮している. 収集した単語は 479 個, ストップワードは 119 個の単語が登録されている. 収集した単語とストップワードの一部を表 2, 表 3 に示す.

表 2 収集した語彙リストの一部

登録された単語	名詞カテゴリ
学生	人
悪魔	
スタッフ	
猿	動物
家畜	
怪獣	

表 3 ストップワードリストの一部

登録された単語	代名詞の分類	名詞カテゴリ
ぼく	自称	人
わたくし		
あなた	対称	
きみ		
どなた	不定称	
だれ		

4.2 誹謗中傷判定

「Lamb」では, 誹謗中傷判定に必要な, ユーザの人物特性と, 「Lamb」およびユーザの発話時刻をリストとして保存していく. ユーザの人物特性は, ユーザが発話を行う毎に, JUMAN++を用いて形態素解析を行い, 体言と用言を一つのリストとして保存していく. そのため, ユーザの人物特性を保存するリストは 2 次元リストとなる. 「Lamb」の発話時刻は, ユーザに対し応答文を出力する時点の時刻を「Lamb」の発話時刻とし, 記録する. ユーザの発話時刻は, ユーザが入力を行った時点の時刻をユーザの発話時刻とし, 記録する. 攻撃度を推定するための, ユーザの人物特性と生成したレッテルのベクトルを, レッテル生成で用いた FastText の学習モデルを用いて単語の分散表現にて取得する.

4.3 応答文生成

4.3.1 Transformer

ユーザの発話に対する応答文の生成に深層学習を用いる。深層学習を用いた応答文生成手法は RNN や LSTM など、数多く提案されている。しかしながら、これらのネットワークでは単語単位で文頭から順番に処理をしていくため、処理を並列化する事による効率的な学習を行う事が難しい。さらに他の位置の単語の内容と混合してしまうという問題もある。そこで、それらを克服する Transformer[16]を使用した。Transformer は、Attention の仕組みを利用した Encoder-Decoder モデルのニューラルネットワークの一種であり、Attention の仕組みによって、構文や意味構造であったり、文章の対応関係を学習することができる。

4.3.2 学習

深層学習の学習用データとして佐藤ら [17] の方法に従い、Twitter のリプライチェーンから獲得した約 100,000 文の発話文・応答文のペアを用いた。また、分かち書きには Sentencepiece[18] を用いた。一般的な形態素解析ツールである MeCab や JUMAN++ では、辞書がすでに構築されているため、SNS で見られる特有の口語表現を分かち書きさせる事が難しい。そのため、End-to-End で辞書を構築できる Sentencepiece を用いた。これにより、未知語を出現しにくくすることができる。これらの処理を行った発話文・応答文のペアを用いて学習させる。

4.4 アプリケーションの実装

本節では、ユーザが「Lamb」とチャットを行うための GUI アプリケーションについて述べる。GUI の構築に Python のライブラリである Eel と、HTML, CSS, JavaScript を用いた。笠井の研究 [10] によると、レッテルを用いた文は体言を骨子とする文であるとされている。そのため、文章内では文頭あるいは文末に位置する表現行為だと考えられる。ここではレッテル貼りを文末での表現と定義する。「Lamb」とのチャット例を図 5 に、その会話中の誹謗中傷判定結果を表 4 に示す。



図 5 「Lamb」とのチャット例

図 5 にあるチャット例では、ユーザの「ありがとう！今

表 4 誹謗中傷の判定結果

レッテル	攻撃度	親密度	判定結果
基礎研究の教授	0.279	0.269	1
応援団のファン	0.288	0.272	1
笑顔の女の子	0.253	0.274	0

日も一日がんばるね。」という発話に対し、「頑張る！笑顔の女の子」と応答している。この「笑顔の女の子」というのがレッテルである。また、他の二文ではレッテル貼りを行っていないことがわかる。これは、表 4 にある通り、ユーザに対するレッテルの攻撃度や、ユーザと「Lamb」の親密度を比較した結果、レッテルを貼るべきではないと判断されたためである。

5. 評価実験

5.1 実験内容

「Lamb」がレッテル貼りを行うことで、ユーザの雑談向けチャットボット使用時に感じる退屈感を軽減させ、雑談向けチャットボットの継続性を向上させる効果があるかの検証をした。大学生 10 人に対して、「Lamb」を実際に使用してもらった。被験者に「Lamb」のレッテル貼りを有効にしたもの（以下レッテル貼り有効）と無効にしたもの（以下レッテル貼り無効）と対話してもらい、使用継続時間と会話数を計測し比較実験を行なった。その後 5 段階評価のアンケート調査をした。表 5 項目 1-3 はレッテル貼り有効と無効に対する共通の調査を行い、表 5 項目 4-5 はレッテル貼り有効に対してのみ調査を行なった。そして使用継続時間と会話数、表 5 項目 1-3 に対して対応のある t 検定を用いた統計的分析を行なった。

5.2 結果

統計的分析の結果、使用継続時間と会話数、表 5 項目 1-2 において $p < 0.05$ となり、有意差を得ることが出来た。しかし、「表 5 項目 3: また会話したいと思った」では $p > 0.05$ となり有意差を得られなかった。「Lamb」に対してのみ行なった表 5 項目 4 ではレッテルに違和感を感じた被験者が多いというマイナスな結果が得られた。

5.3 考察

使用時継続時間と会話数は継続したが、「表 5 項目 3: また会話したいと思った」では有意性を得られていないため、「Lamb」はシステム使用時の一時的な継続性のみが向上し

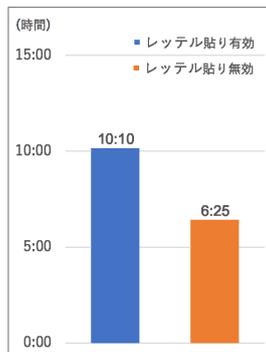


図 6 平均継続時間 ($p=0.0033$)

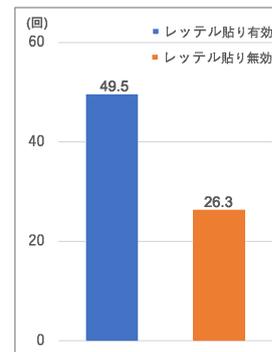


図 7 平均会話数 ($p=0.0007$)

表 5 評価実験のアンケート内容と結果

項目	質問内容	評価平均		検定の結果 (p 値)
		(レッテル貼り有効)	(レッテル貼り無効)	
1	発話に退屈感を感じた	2.2	2.9	0.037
2	発話に冗談やユーモアを感じた	3.8	2.7	0.017
3	また会話したいと思った	3.5	2.9	0.168
4	レッテルに違和感を感じた	4.3		
5	レッテルを冗談と受け取ることが出来た	4.6		

表 6 レッテル貼りの出力例

項目	ユーザの入力	システムの出力	違和感
1	たばこやめたいけど、どうしてもすっちゃう	喫煙者ってなんのために生きてるんでしょうね。タバコの奴隷!	なし
2	事故のせいで車両止まったんだけど	マジかよ www. 車両故障の乗客!	なし
3	将来の夢はお金持ちで健康で良い家庭を築く事です	格好いい。夢見の父親!	なし
4	最近忙しくて大変だよ	ホンマに大変だね。昔の先祖!	あり
5	今日後輩と飲みに行きます	行ってらっしゃい。今のお父さん!	あり

ていると考えられる。

表 6 項目 4 は評価実験の際に、被験者が違和感を感じたレッテル貼りの出力の一例である。「Lamb」ではレッテルの生成時に文章の係り受けを考慮しない単語の言い換え手法を用いているため、表 6 項目 4 の入力文の中であまり重要ではない「最近」という単語の特徴を必要以上に含む言い換えがされてしまっている。これによりユーザの発言を的確に表すレッテルを生成することが出来ず、被験者の感じた違和感を生み出したと考えられる。

さらに、被験者からはレッテル貼りの回数が少なく物足りなかったという意見が寄せられた。これは、誹謗中傷判定の際に、親密度と攻撃度に適切な正規化処理やパラメータの設定をしていないため、高い精度で判定が出来なかったことが原因と考えられる。

6. おわりに

本研究では、ユーザのチャットメッセージからレッテルを生成し、レッテルの攻撃性を踏まえた発話タイミングでレッテル貼りをを行う雑談向けチャットボット「Lamb」の開発を行った。「Lamb」の行うレッテル貼りは発話に冗談やユーモアを付与する効果が実証され、使用継続時間、会話数といったシステム使用時の継続性を高めることが出来た。しかし、ユーザにまた会話したいと思われるようなシステムにはならなかった。また、現在「Lamb」が生成するレッテルは人間の対話で日常的に使われているような自然なレッテル貼りとは言い難い。それは「Lamb」が生成した違和感のあるレッテルとレッテル貼りをを行うタイミングが原因であった。今後はユーザの発話をよりの確に表すレッテルの生成と適切な正規化処理を用いた誹謗中傷の判定、そして「Lamb」の発話する応答文に自然にレッテルを組み込む技術の開発に取り組んでいきたい。

参考文献

- [1] 小林峻也, 萩原将文: 嗜好や人間関係を考慮する非タスク指向型対話システム, 『人工知能学会論文誌』(2015).
- [2] 岩倉亮介, 吉川大弘, ジメネスフェリックス, 古橋武: Twitter データを用いたユーモア語句自動生成手法に関する一検討, The 31st Annual Conference of the Japanese Society for Artificial Intelligence (2017).
- [3] Xianchao Wu, Kazushige Ito, Katsuya Iida, Kazuna Tsuboi, Momo Klyen: りんな: 女子高生人工知能, 『言語処理学会 第 22 回年次大会 発表論文集』(2016).
- [4] 藤倉将平, 小川義人, 菊池英明: 非タスク指向対話システムにおけるユーモア応答生成手法, The 29th Annual Conference of the Japanese Society for Artificial Intelligence (2015).
- [5] 北原保雄: 『日本国語大辞典 第二版』(2003).
- [6] 大津友美: 親しい友人同士の会話におけるポジティブ・ボライトネー「遊び」としての対立行動に着目して一, 『社会言語科学』第 6 巻第 2 号 pp44-53 (2004).
- [7] 大津友美: 会話における冗談のコミュニケーション特徴—スタイルシフトによる冗談の場合—, 『社会言語科学』第 10 巻第 1 号 pp45 - 55 (2007).
- [8] 呉健朗, 中原涼太, 長岡大二, 中辻真, 宮田章裕: ボケて返す対話型エージェント, TVRSJ Vol.23 No.4 pp.231-238 (2018).
- [9] Aditya Joshi, Anoop Kunchukuttan, Pushpak Bhat-tacharyya, Mark J Carman. SarcasmBot: An open-source sarcasm-generation module for chatbots, WIS-DOM at KDD 2015.
- [10] 笹井香: レッテル貼り文という文, 『日本語の研究』第 13 巻 4 号 pp18-34 (2017).
- [11] 東中竜一郎, 船越孝太郎, 荒木雅弘, 塚原裕史, 小林優佳, 水上雅博: テキストチャットを用いた雑談対話コーパスの構築と対話破綻の分析, 『自然言語処理』 vol.23 No.1 (2016).
- [12] Bateson G: 『Steps to an ecology of mind』Chicago: The University of Chicago Press (1972).
- [13] 小寺暁久, 横山昌平, 山田文康: Twitter におけるユーザ同士の会話に基づいた親密度の評価と時系列的変化の可視化, DEIM Forum 2015 F8-4.
- [14] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov: Enriching Word Vectors with Subword Information (2016).
- [15] 日本語形態素解析システム JUMAN++, 京都大学黒橋・河原研究室 <http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN++>
- [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: Attention Is All You Need (2017).
- [17] 佐藤祥多, 乾健太郎: 因果関係に基づくデータサンプリングを利用した雑談応答学習, 情報処理学会 (2018).
- [18] Unsupervised text tokenizer for Neural Network based text generation. <https://github.com/google/sentencepiece>