

ロボットの日常会話におけるユーモア生成の試み

A humor generator applied to everyday conversation with human

中谷 仁^{*1}

Hitoshi Nakatani

岡 夏樹^{*1}

Natsuki Oka

^{*1} 京都工芸繊維大学 大学院 工芸科学研究科 情報工学専攻

Department of Information Science, Graduate School of Science and Technology, Kyoto institute of technology

Although there have been a lot of studies of humor generation with a computer, most of them have focused on plays on word like riddle, and hence they cannot be applied to everyday conversation between human and robot. As opposed to previous studies, this paper focuses on generating puns in everyday conversation. We propose an algorithm that generates puns by replacing a word in a daily utterance of a robot with a new word. The replacer is selected based on the meaning of the replacer and the replace. An experimental evaluation demonstrated the possibility of our method to generate humor.

1. はじめに

人型ロボットやペットロボット等人と接するロボットが開発され、親しみやすいデザインのロボットが増えているが、人とロボットの間にはまだ隔りがあるように感じる。それを解消するには、ロボットが人間のように冗談を交えながら話すのが良いと考える。ロボットにユーモアを生成する能力を備え付けることにより、人とロボットとのインタラクションが円滑になるという利点が生まれるのではないかと。

コンピュータユーモアについての従来の研究は、落語の大喜利の題材として用いられるなぞかけを生成する研究[前田 05]や、「サイはサイでも不足しがちなサイはなんだ? :答 野菜」のようななぞなぞを生成する研究[Sjobergh 07]などがあるが、これらの言葉遊びを対象にした研究は、ロボットと人との日常会話の中でユーモアを生成させることに応用することは難しい。

そこで本研究は、日常会話に使えるユーモアについて考察し、それを実際に実装するために「単語の意味」を利用することで、ロボットが発話するために用意した文章を変換し、駄洒落を生成することで日常会話にユーモアを生み出せると考えた。

2. 駄洒落についての考察

2.1 駄洒落の型についての考察

本研究では、日常会話に用いられるユーモアとして、駄洒落を考察した。駄洒落は定式化しやすくコンピュータでの処理に向いており、駄洒落の生成は今までも多くの研究がされている。ここで、駄洒落の生成方法の観点から見ると、駄洒落には大きく分けて併置型と置換型の二種類があると言える。^{*1} [ビンステッド 98]

まず、併置型は「委員会に行かなくて、いいんかい」のような、ある語の類似音語を近接して配置する駄洒落であり、置換型は図1のような「見よ、事務(ジム)で鍛えたこの体」のように、ある語を類似音語と置き換える駄洒落である。

ここで、二つの駄洒落のタイプについて、ロボットに自然な会話の流れから各々を生成させる場合について考察する。

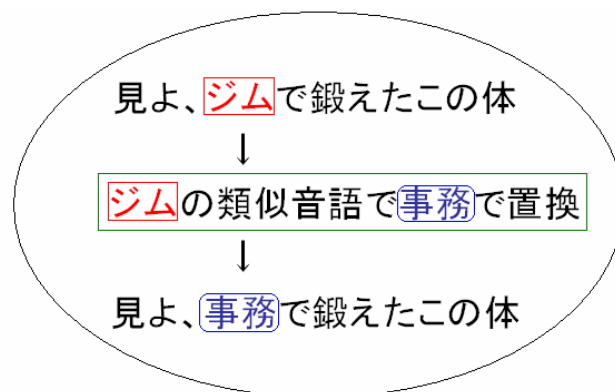


図1: 置換型駄洒落の例

併置型は、まず「蒸気機関車の気管支炎」のような意味の通らない駄洒落が生成されてしまう事があり、さらに、その文章が単独で意味が通っていても、会話の中で使うには不自然なものが生成される。これらを会話の中で用いると、会話の流れが途切れてしまう。例えば、自然な会話の中では「誰かが何かの委員会に行かない、かつその誰かは委員会に出席しなければならない」という事実があって初めて「委員会に行かなくて、いいんかい」という文章を発話できる。

よって併置型は、生成のタイミングが限定され、かつそのタイミングの推測が難しいという短所がある。

一方置換型は、本来の発話文章の一部の変更により生成できるので任意のタイミングで生成でき、かつ類似音語と駄洒落全体から置換前の語を推測できるので本来の発話内容を伝達できるという利点を持つ。例えば、「今アンカーにタスキが渡されました」という文章から、「今アンカーにたぬきが渡されました」という駄洒落を生成し発話しても、発話した駄洒落から「タスキが渡された」という事実が推測できるため会話の流れが途切れない。

以上の点から、本研究では置換型の駄洒落を生成対象とする。

2.2 田辺の研究

田辺の研究[田辺 03, 田辺 05]は置換型の駄洒落を生成する手法についての研究である。田辺は意味論的分析により、「駄洒落生成の際に置き換える語(以下置換語とする)は、置き

連絡先: 中谷 仁, 京都工芸繊維大学 大学院 工芸科学研究科 情報工学専攻, 京都市左京区松ヶ崎御所海道町, m9622020@edu.kit.ac.jp

^{*1} 実際には対比型も含めた3種類に分けられるとされているが、一般的でない判断したため、ここでは取り上げない

換える前の語(以下被置換語とする)に対して, 特定の観点から見て最も差が大きいとみなされ, かつ, より一般的な語を選ぶ」という生成規則を提案した. その実現方法として, [田辺 03]では, 各々の単語をいくつかの観点について主観評価した値を用いて, [田辺 05]では概念ベースという, 各概念をその特徴を表す属性と, 属性がその概念でどの程度重要かを表わす0から1の数値(重要度)の対の集合で表わされた表を用いて, それぞれ語の観点の差を計算する手法を提案している.

田辺の研究は, まだコンピュータでの実装には至っていない. そこで本研究では, 生成規則はそのままに, PLSI(Probabilistic Latent Semantic Indexing)を用いて語の意味をコンピュータ上で算出し, SVM(Support Vector Machine)を用い, できるだけ生成規則に沿った意味を持つ置換語の抽出を行い, 駄洒落を生成することを試みる.

3. 駄洒落生成システムの構築

3.1 システム全体の構成

今回コンピュータにおいて実装した駄洒落生成システムは, 図2のように, 入力をロボットが発話するはずである文章(以下発話文章とする), 出力をその文章から作成した駄洒落とし, 形態素解析部, 置換語候補検索部, 共起データ作成部, 置換語選択部, の4つのモジュールから構成されるシステムである.

ここで各モジュールの動作について説明する. まず, 形態素解析部は発話文章を形態素解析し, 形態素を被置換語として出力するモジュールである. 実装には形態素解析エンジン MeCab[工藤 08]を用いた. MeCab により形態素の読み仮名情報と, 品詞情報を得ることができる. 置換語候補検索部は, 3.2節で示す条件を満たす被置換語の類似音語を出力するモジュールである. 共起データ作成部は, 置換語選択部において用いられる名詞と動詞の共起回数を数え上げ, 出力するモジュールである. 今回は EDR 電子化辞書[情報通信研究機構 02]の日本語コーパス中の, 一つの文中にある名詞とある動詞が生じた場合「共起した」とした. 置換語選択部では, 置換語候補の中から, PLSI で算出した語の潜在的な意味を入力として, ユーモアを生成させる置換語の選択を学習させた SVM を用いて, 置換語を決定し出力するモジュールである. そして, 発話文章と置換語を組み合わせてできた駄洒落を出力するという構成になっている.

3.2 置換語候補検索部の構成

今回, 置換語候補検索部において, 置換語候補として出力する被置換語の類似音語の条件は, 「読み仮名が被置換語の読み仮名と前方部分一致, または後方部分一致する語でかつ語長が被置換語の 1.5 倍までの語」とした. ここで語長は, 仮名一文字を1として数えるものとする. 例として被置換語を「担任」の場合, 置換語候補の一つとして「探検」が含まれる. この条件を適用すると, よく用いられる以下のパターンの駄洒落を生成できる.

- (1) 置換
一文字を他の文字に置き換えた語. 例: たすき たぬき
- (2) 付加
一文字を元の語に付加した語. 例: はり はかり
- (3) 削除
一文字を元の語から削除した語. 例: たまご たま

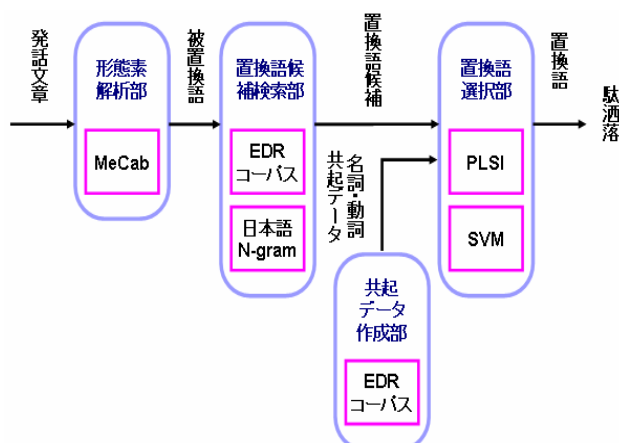


図2: 本研究で構築した駄洒落生成システム

今回は置換語候補の検索対象を, EDR 日本語コーパスに出現し, かつ Web 日本語 N グラム第一版[工藤 07]における Web 上での生起確率上位5万位までに含まれる語とした. これは, 普段聞きなれない難解な単語は意味が理解できないため, それを用いた駄洒落を出力してもユーモアを生成できない可能性が高いため, それらを置換語候補として選択され難くするための処置である.

3.3 置換語選択部の構築

3.3.1 PLSI

PLSI(Probabilistic Latent Semantic Indexing)[Hofmann 99]は文書ベクトルに対して次元圧縮を行う手法の一つであり, 潜在的な意味 z を介して文書 d と単語 w が共起するとしたモデルである. 文書 d と単語 w の共起確率 $P(d, w)$ は

$$P(d, w) = \sum_z P(d | z)P(w | z)P(z)$$

で表される.

文書 d と単語 w の共起頻度を $n(d, w)$ とした時, 対数尤度 L は次のようになる.

$$L = \sum_{d,w} n(d, w) \log P(d, w)$$

この L を最大にするように $P(z), P(z | d), P(w | z)$ は決定する. これらを TemperedEM アルゴリズムを用い最尤推定する. TemperedEM アルゴリズムは,

E-Step

$$P_\beta(z | d, w) = \frac{P(d)[P(d | z)P(w | z)]^\beta}{\sum_{z'} P(d)[P(d | z')P(w | z')]^\beta}$$

M-Step

$$P(w | z) = \frac{\sum_d n(d, w)P(z | d, w)}{\sum_{d,w'} n(d, w')P(z | d, w')}$$

$$P(d | z) = \frac{\sum_d n(d, w)P(z | d, w)}{\sum_{d',w} n(d', w)P(z | d', w)}$$

$$P(z) = \frac{\sum_{d,w} n(d, w)P(z | d, w)}{\sum_{d,w} n(d, w)}$$

逆温度パラメータ β は, $\beta = 1.0$ とすると EM アルゴリズムになるため確率分布が鋭くなり, $\beta < 1.0$ とすると, 確率分布がなだらかになり, 過学習を防げる. 今回は $\beta = 0.75$ とした.

また, [萩原 05]のように, 文書 d の代わりに動詞 v を用いることもあり, 本研究でも, 文書 d と単語 w の代わりに, 動詞 v と名詞 n を用いた.

本研究では名詞 n の潜在的な意味の確率分布 $P(z|n)$ を意味として用い, 語と語の意味の潜在的な意味の差を算出する. また, 今回の z の次元数は 30 とした.

3.3.2 SVM

SVM(Support Vector Machine)はパターン認識において最も優秀な機械学習手法の一つであるとされている. これは図3のようにある事例の特徴量とクラスが与えられた時, 特徴空間上のクラスの境界面を学習することによりクラスタリングを行う手法である. 汎化能力が高く, 「カーネルトリック」を用いることで非線形分類も可能となる.

本研究では, 図4のように被置換語 n' と置換語 n'' の潜在的な意味のベクトル $P(z|n')$, $P(z|n'')$ を合わせたものを特徴ベクトルとし, クラスは生成された駄洒落が面白い, 面白くないという二クラスの評価を与えることとした.

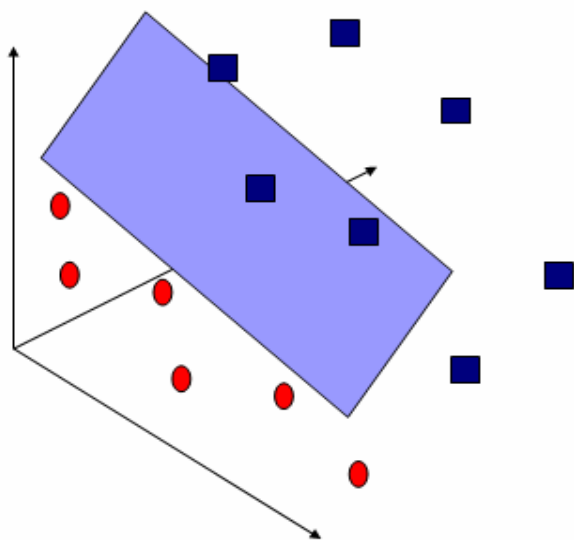


図3:SVM での特徴空間上での境界面

4. 実験および評価

本章では, 構築したシステムの評価を行うために, システムから置換語選択部を除き, 置換語候補からランダムに抽出した置換語により生成された駄洒落(以下ランダム駄洒落)を人間が評価し, 同様にシステムがどう駄洒落の評価を行うかを比較し, 語の意味を用いてユーモアが生成されているかどうかを評価する.

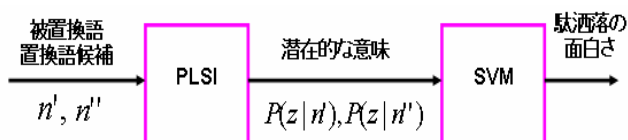


図4:置換語選択部の構造

$$\text{多項式カーネル: } k(x, y) = (x^T y + c)^d$$

$$\text{RBFカーネル: } k(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma^2}\right)$$

$$\text{シグモイドカーネル: } k(x, y) = \tanh(ax^T y - b)$$

$$\text{ANOVERBFカーネル: } k(x, y) = \left[\sum_i \exp(-s\|x_i - y_i\|^2)\right]^d$$

図5:SVM 使用を検討したカーネル

4.1 実験設定

今回は, 駄洒落の面白さの評価は, 主観評価で面白い, 面白くない, の二クラスとした. また, 駄洒落の評価は著者が全て行っている. これは人によって面白さの基準が違い, 評価が一定にならず機械学習をさせることが難しくなることを避けるためである. 発話文章には WindowsXP 上の Help ファイルの文章を用いて実験を行った. 理由として Help ファイルはコンピュータと人との一つの対話の形であると考えられるからである. 被置換語, 置換語は今回名詞のみとする. これは[田辺 05]において, 被置換語となるのは 68% が名詞であるという事実が述べられており, システムの複雑化を避けるために名詞のみとした.

まず, SVM に駄洒落の面白さを学習させる. SVM に学習させるデータは, ランダム駄洒落より抽出した 2300 文の駄洒落を評価したものである. 使用するカーネルは, 図5に示した4つのカーネルにおいて, 各々パラメータを変化させて 10-fold クロスバリデーションを行った結果, 平均再現率が最も高かった ANOVRBFカーネルにおいてパラメータ $s = 1.6, d = 3$ としたものとした. 指標として再現率を選んだのは, 面白くない駄洒落を面白いと判断して出力するのは問題があるが, 面白い駄洒落を面白くないと判断して出力しないこと自体には問題はないと言えるからである.

そして, 学習データに用いた以外の発話文章を入力し, 出力された駄洒落を評価し, SVM の出力と比較し, システムの正答率を100文の駄洒落について評価する. 表1に実際に使用した学習データの一部を示す.

4.2 実験結果

システムが出力した面白さの評価と人間が出力した面白さの評価については表2の通りとなった. ここで
帰無仮説: システムの駄洒落の面白さの判断と人間が実際に感じる面白さは相関がない.

対立仮説: システムの駄洒落の面白さの判断と人間が実際に感じる面白さは相関がある.

という仮説を立てる. カイ二乗検定を行ったところ $\chi^2(1) = 2.87$, $P < .1$ となり, 帰無仮説を棄却し対立仮説を採用する. よってシステムは駄洒落の面白さを学習できたといえる.

5. 考察

システムが面白いと判断した駄洒落の半数以上が人間は面白くないと判断している. よって実験は成功したがシステムの精度を改善すべきである. 表2に成功した駄洒落の例を挙げてい

表1:学習データ例

駄洒落	被置換語	置換語	面白い
しかし、自分で変更しない限り軟弱度は変わりません	難易	軟弱	Yes
検索結果の円筒をダブルクリックし、対応するトピックを表示します	エントリ	円筒	No
カードを1枚組札に移すと、10天加算されます	点	天	No
ほかに方法がないばあさんは、これらの時計は手動で設定する必要があります	場合	ばあさん	Yes
予防作文として、サードパーティ製のプラグインはすべて無効になっています	策	作文	No
ブロックレイアウトでは、方向キーボードが右下隅にまとめて表示されます	キー	キーボード	No
私と同じ難易度のプレイヤーが見つからないときめきはようになりますか?	時	ときめき	Yes
特売のタスクを実行する方法については、次の項目をクリックしてください	特定	特売	No
時間制のゲームの場合は、10秒経過するごとに2点減点されます	ごと	ごとき	Yes
マイクの差し押し口が複数付いているコンピュータもあります	差込	差し押し	No

るが、「Windows 怠慢サービス」は「タイム」という少し厳格なイメージのする語から「怠慢」というだらしないイメージの単語に置き換わることによって成功した例である。また、「これによりメッセージ配信は口臭になります」の「口臭」のように、被置換語に関係なく、置換語がユーモアを生成しやすい単語であったため面白くなった駄洒落も多くあった。「この冗談では、MicrosoftSMTPService はメッセージを返しません。」のように、意味が通り面白い文章になることもあったが、生成できたのは偶然であり、今回のシステムには意図的にこのような文章を生成する仕組みは存在しないので、システム性能評価からは除外する。また、このような成功例は稀であり、文脈を考慮した設計ではないので当然だが、ほとんどが意味の分からない文章ができる。よってシステムの精度が悪かったことについて、学習に用いる情報が少なかったのではないかと考える。発話文章全体の情報も含めて学習すべきであったと考える。発話文章として WindowsXPHelp 文章を用いたため、コンピュータに関する特殊な文章を対象に駄洒落を作っていたため、一つの単語を変えただけでは文全体に面白さが出なかったようだ。また、置換語から被置換語を想起できないものも数多く存在した。今回は WindowsXPHelp 文章の特殊性や、会話のように文脈に沿って次の文章が出てくるわけではなく、脈絡のない一つ一つの文章が出力される形式のために、被置換語の想起が通常の会話で使われるよりも困難であるが、それでも一音が変わっているものなら連想できるので、置換型駄洒落の利点を失わないため、類似音語の条件は一音までの変化を許すものにするべきである。

表2:人間とシステムの駄洒落の評価

	システム: 面白い	システム: 面白くない	合計
人間面白い	10	12	22
人間面白くない	51	27	78
合計	61	39	100

表3:成功した駄洒落の例

Windows 怠慢サービス
これによりメッセージ配信は口臭になります
この冗談では、MicrosoftSMTPService はメッセージを返しません。

6. おわりに

本研究では人とロボットとの日常会話に応用できる置換型駄洒落を出力するシステムを構築した。語の意味を用いて面白い駄洒落の生成を行い、成功した。だが、面白い駄洒落を生成する精度については改善の余地がある。

今後の課題は、置換語候補検索部における類似音語の導出方法の改良と、置換語選択部における面白い駄洒落を作るのに適切な置換語を選択するアルゴリズムの構築である。

参考文献

- [前田 05] 前田 美香:言葉の関連性を用いたなぞかけ生成とその評価,筑波大学大学院博士課程 システム情報工学研究科修士論文,(2005).
- [Sjobergh 07] Sjobergh Jonas, 荒木 健治:日本語を対象とした謎掛けの自動生成,情報処理学会研究報告 自然言語処理研究会報告,IPSI SIG Notes 2007(35), pp.91-95(2007).
- [ピンステッド 98] キム ピンステッド,滝澤 修:日本語駄洒落なぞ生成システム”BOKE”,人工知能学会誌,Vol.13,No.6, pp.920-927(1998).
- [田辺 03] 田辺 公一朗:駄洒落の形式化,Sanno University Bulletin,Vol.24, No.1, pp.213-222 (2003).
- [田辺 05] 田辺 公一朗:駄洒落のコンピュータによる処理 駄洒落生成システムの基本設計, Sanno College bulletin, School of Management and Informatics, Vol.26, No.1, pp.65-74(2005).
- [工藤 08] 工藤 拓:MeCab Yet Another Part-of-Speech and Morphological Analyzer, <http://mecab.sourceforge.net/>.
- [情報通信研究機構 02] 独立行政法人 情報通信研究機構: EDR 電子化辞書, (2002).
- [工藤 07] 工藤拓,賀沢秀人:Web 日本語 N グラム第1版,言語資源協会発行, <http://www.gsk.or.jp/>(2007).
- [Hofmann 99] Thomas Hofmann:Probabilistic Latent Semantic Indexing, Proc. of SIGIR'99, ACM Press, pp.50-57, (1999).
- [萩原 05] 萩原 正人,小川 泰弘,外山 勝彦:シソーラス自動構築における PLSI の利用,情報処理学会研究報告 自然言語処理研究会報告 Vol.2005, No.22 (2005).