# 対立語抽出に基づくWebニュースからの漫才ロボット台本自動生成手法の提案

真下　遼†　灘本　明代†

† 甲南大学 知能情報学部　〒 658-8501 兵庫県神戸市東灘区岡本 8-9-1
E-mail: †hr-x7type@yahoo.co.jp, ††nadamoto@konan-u.ac.jp

**あらまし**　ロボット工学の進歩によりコミュニケーションロボットの開発が進んでいる．しかし，人がロボットをコミュニケーションの対象と捉えるのは未だ困難である．これまで我々は，人とロボットとの円滑なコミュニケーションの実現を目的とした漫才ロボットを行うために，ロボットに実装し実演するための漫才台本自動生成システムを提案してきた．本論文ではさらに，Web ニュースの主題の対立語を抽出し，その対立語を用いた漫才台本自動生成手法を提案する．

**キーワード**　ロボット，Web，漫才台本自動生成，対立語

## 1. はじめに

近年，ロボット工学の技術は急速に進歩しており，様々なロボットの研究開発が行われている．特に人とコミュニケーションを図ることを目的としたコミュニケーションロボットの研究開発の発展は，近い将来にロボットがより身近な存在となって人々と共存する社会の実現を予期させる．しかしながら，人がロボットをコミュニケーションの対象と捉えるには未だ抵抗があると考えられる．

人とロボットに関する研究として，神田ら [1] はロボット同士の対話を人が観察することにより，人とロボットとの関係性が緩和され，人はロボットに対して人と接するのと同様の自然なコミュニケーションを実現できることを明らかにした．そこで我々は，ロボット同士の対話の内，娯楽性の高い対話である漫才に着目した．ロボットが演じる漫才をユーザに提供することでロボットとのコミュニケーションの円滑化が容易に行えると考えた．一方で，漫才が情報提供コンテンツとしての役割を持つことに着目すると，ユーザにはより理解しやすく親しみのある情報提供が望ましいと考えられる．その手がかりとして元お笑い芸人であり作家の松本哲也氏は「時事ネタで漫才を作るのは，一番お客さんの共感を得られやすい」[2] と述べている．そこで我々は，ユーザが興味のある内容の Web ニュース記事を漫才の題材に用いた漫才ロボット台本自動生成を提案する．

Web ニュース記事による漫才ロボット台本自動生成は以下の点で有用であると考えられる．1 つは，ユーザがロボットの漫才を観るという受動的な行為だけで Web ニュースの記事の情報を得ることが可能になり，若者を中心とした現代のニュース離れの克服を期待できるという点である．もう 1 つは，漫才ロボットでは笑いを交えて情報を提供することで，堅苦しく難しい内容の多いニュース記事であってもユーザは親しみを持って容易に情報を得ることが可能であるという点である．同じ受動メディアであるテレビニュースと比較しても，漫才ロボットはテレビニュースのように時間帯の制限を受けることがなく，台本生成素材に Web ニュースの速報記事を用いることで，漫才ロボットはテレビよりもさらに速報性の高い情報提供が可能となる．また，先に述べたように漫才ロボットによりロボットへの親しみの向上や癒しの提供も期待できる．

以下 2. 節で関連研究について述べ，3. 節では我々の提案する漫才台本自動生成のシステムと台本の構成について説明し，4. 節では，本論文で提案する対立ボケの自動生成手法について述べ，最後に 5. 節でまとめと今後の課題について述べる．

## 2. 関連研究

Web ニュースからの漫才台本自動生成にはニュースの理解度を高めることを目的の 1 つとしているが，同様にニュースの理解度を高めることを目的とした研究として，北山ら [3] は，比較ニュースに着目し，ユーザが閲覧しているニュースに関連するニュースをニュースアーカイブから的確に検索する検索方式を提案した．本研究では，ユーザが閲覧するニュース記事自体に娯楽性を付与して提供し，ニュースに対するユーザの親しみの向上からニュースの理解度を高めるといった点で異なる．

本論文での漫才台本自動生成では，ニュースに中に含まれる語に関連する語を発見し利用することで漫才を生成するが，このような情報発見の研究はいくつか行われている．佃ら [4] はある語に関する意外な情報の発見を目的に，語の意外度をグラフ構造と認知度によって用いて計算する手法を提案した．Oyama ら [5] はある語を詳細に説明する詳細語を Web ページの文章構造を利用して発見する手法を提案している．また，語の関係性に関する研究として Church ら [6] は，2 語の意味的関係性を相互情報量を用いて計算している．Turney ら [7] は，類義語の関係性を Web 検索の検索結果数を用いて計算している．Rudi ら [8] の Google Similarity Distance も Google の検索エンジンからの検索結果数を利用して，意味の関係性の度合いを計算している．計算式として正規化 Google 距離というものを提唱している．本研究では，Wikipedia を利用した語の階層構造と Web 検索の検索結果数の両方を用いることで，キーワードに関

して対照性のある語を一意に発見する．

漫才に関する研究も行われており，関 [9] は漫才の分析により漫才の対話中に含まれる対話のズレと笑いの関係性を明らかにしている．安倍 [10] は笑いが生まれる構造として「フリ」，「ボケ」,「ツッコミ」の要因に着目し，その関係性を視覚的に表現している．

## 3. 漫才台本自動生成

### 3.1 システムの概要

本論文で提案するロボットの漫才台本自動生成システムはユーザがキーワードを入力し，そのキーワードに関する記事の内容を漫才形式の台本に自動変換した後，その台本をロボットが演じてユーザに提示するシステムである．なお，本論文ではボケ役とツッコミ役の 2 体のロボットの対話形式でのしゃべくり漫才を前提としている．システムのおおまかな流れは以下のようになる．

（1） ユーザがキーワードを入力する．

（2） システムは入力されたキーワードを検索クエリとして Yahoo ニュースより記事を最新のものから，無作為に 1 記事を決定し．その記事のタイトル及び本文を抽出する．

（3） 抽出した記事のタイトル及び本文から様々なボケやツッコミを自動で生成し漫才台本を自動生成する．

（4） 自動生成した漫才台本を 2 体のロボットに演じさせる．

入力キーワードは 1 語でも複数でも構わず，またニュース記事の URL を直接指定して記事を選出することも可能である．次に選出した記事の本文の抽出を行うが，記事の本文の長さは記事によって大きく異なる．本研究の漫才はロボットの演じる時間を冗長の少ない 4 分以内と収めたため，抽出する記事の本文の文字数を 400 文字以下に制限する．ただし，記事の概要を損なう本文の抽出を避けるために本文の抽出を段落単位で行う．通常ニュース記事は内容理解に重要な項目順に構成される [11] ので，提案手法でも第 1 段落から順に 400 字の文字制限が超えない範囲まで段落単位で本文を抽出する．また，漫才の題材に人の死や不幸を用いるのは不謹慎であるため,「死亡」や「殺害」等をストップワードとし，記事タイトル及び本文中にこれらが含まれている場合は台本生成に不適切な記事として記事の選出をやり直す．本文抽出後は，提案手法により台本を自動生成する．台本は XML ファイルの形式で生成され，出来上がった台本をもとに 2 体のロボットが漫才を演じる．台本には台詞以外に細かい動きや表情の変化等も自動生成で記述する．

本論文で提案する漫才台本を演じるロボットは図 1 の 2 体のロボットである．全長約 100cm の背の高い方のあいちゃんがツッコミ役，全長約 50cm の背の低い方のゴンタがボケ役をそれぞれ担当する．2 体のロボットはいずれも手足等のインターフェースが存在しないが，代わりにモーターでの回転動作でツッコミの動作を表現する．また，目の部分は 50 種類の画像で切り替わり，喜怒哀楽の感情を表現することが可能である．

図 1 漫才ロボットあいちゃん（左）とゴンタ（右）

### 3.2 漫才の構成

漫才台本自動生成にあたり，台本をある程度形式化する必要がある．本論文では灘本ら [12] が提案していた漫才台本自動生成の枠組みを使用し，漫才台本をつかみ，本ネタ，オチに分割した三段構成の流れでの漫才台本の生成を行う．

#### 3.2.1 つ か み

つかみでは，挨拶を兼ねた最初の笑いと本ネタへの話題提供を行う．導入としては，自動生成当時の月の行事に関する身近な話題で開始する．次に，つかみでの最初の笑いとして表情ボケを行う．表情ボケとは，ロボットが話す内容に応じて感情を目の表情で表現するが，その表情が話す内容には明らかに不適切であるというボケである．また，本ネタへの話題提供としては台本生成の題材となった Web ニュース記事のタイトルを読み上げて記事の詳しい内容に触れる対話の流れを作る．以下に『2020 年「東京五輪」に決定』の Web ニュース記事で自動生成した漫才台本のつかみの例を示す．なお，漫才の演者である 2 体のロボットの内，主にボケ役を担当するロボットを「ボケ」，主にツッコミ役を担当するロボットを「ツッコミ」とそれぞれ表記する．

**ボケ**「いやー，ひさびさの地球だなぁ.」
**ツッコミ**「ホンマやなー.」
**ボケ**「もう地球は 3 月で 卒業のシーズンや.」
**ボケ**「ホンマ悲しいな〜.」（表情ボケ：楽しそうな表情をしながら）
**ツッコミ**「おいおい，楽しそうな顔になっとるやないか悲しいんならこっちやで.」（悲しそうな表情をしながら）
**ボケ**「おー，そうやった．この顔やったわ.」（悲しそうな表情をしながら）
**ツッコミ**「分かればええんや.」
**ツッコミ**「そんな事よりも．地球のことちょっとはわかってきたやん.」
**ボケ**「当たり前やないか，地元で地球のこと勉強しとんねんから!」

**ツッコミ**「ほな地球でこんなニュースあったん知ってるか?」
**ボケ**「ん?どれどれ … 2020 年「東京五輪」に決定? … 知らぬ.」
**ツッコミ**「アカンやん!せっかくやし今ちょっと読んでみて.」

#### 3.2.2 本　ネ　タ

本ネタでは，題材となったニュース記事の内容を読み上げてユーザに説明しながら，同時に様々なボケとツッコミを挿んでで笑いをとる．ここが漫才において最も主軸となる部分であり，漫才全体での割合時間も長く，ボケの回数も最も多く挿入される．ボケの挿入は，記事の構造をユーザが把握できるように，句読点による文単位で行い，1 文に付き最大で 1 ボケを挿入する．ただし，記事の最初の 1 文は記事の概要が顕著に表れるためボケの挿入は行わない．また，ボケが作成されない 1 文であっても，ツッコミは相槌を行うことで対話に参加する．本研究で自動生成されるボケとツッコミには，言葉遊びボケ，ノリツッコミ，過剰ボケ，そして対立ボケがある．以下は，自動生成された本ネタでの言葉遊びボケ及びノリツッコミの一例を示す.

**ボケ**「東京は決選投票でイスタンブールを破り，1964 年以来 2 度目となるオリンピック開催を決めた.」
**ツッコミ**「なるほど.」
**ボケ**「マドリードは 1 回目の投票でイスタンブールと同票となり,」
**ツッコミ**「うん.」
**ボケ**「最下位を決める ”凍傷 ”で落選した.」（言葉遊びボケ）
**ツッコミ**「そうそう，凍傷はホント怖いなー … って.」（ノリツッコミ）
**ツッコミ**「なんでやねん!凍傷って!それは低温が原因で生じる皮膚や皮下組織の傷害やろ!凍傷ちゃうくて投票や!」
**ボケ**「おっと，うっかりしてもうた.」
**ツッコミ**「しっかりせえよ.」

**言葉遊びボケ**

言葉遊びボケとは，ボケ役がニュース記事本文の単語を別の単語と読み間違うというボケである．例えば,「投票（とう-ひょう)」という単語を「凍傷（とう-しょう)」という単語と読み間違えるといったものである．この場合，ツッコミ役がその読み間違えた単語をツッコミ，正しい単語に訂正することでボケとツッコミの流れが成立する．言葉遊びボケの自動生成では，記事本文中に含まれる一般名詞をキーワードとして設定し，ローマ字に変換する．さらにローマ字に変換したキーワードの母音と子音を総当りで変化させて別の単語を生成する．この時，生成される単語は複数の場合があるが，ユーザへのわかりやすさを考慮して小学生の国語辞典をコーパスとして生成した単語と照らし合わせ，コーパス中に出現する単語のみをボケとして用いる．また，ツッコミとなる正しい単語への訂正には，その単語をタイトルとする Wikipedia の記事を利用することで，間違えた単語の説明を補足させる．補足情報として Wikipedia 記事の最初の 1 文がそのタイトルの概要を顕著に表していることから，その最初の 1 文を用いる．例えば,「凍傷」の説明には,「低温が原因で生じる皮膚や皮下組織の傷害」といった一文が抽出される．金水 [13] は漫才における笑いの発生は漫才の対話が Grice [14] の提案した言語表現の果たす機能である協調の原理からズレるために生じると述べている．言葉遊びボケでのツッコミの正しい単語への訂正の補足情報として Wikipedia 記事の文を用いることは，多くの場合訂正文としては冗長的印象を受ける．しかし，冗長的な文は協調の原理の内の量の公理にズレる内容であるため，ツッコミの訂正文を笑いが生じる 1 つの要因にできる利点があるので漫才においては有用であると考えられる．

**ノリツッコミ**

ノリツッコミとは，通常のツッコミと異なりツッコミ役が一度ボケの内容に同調し（のっかり）話題を展開した後に，改めて正しいツッコミを行うというものである．ノリツッコミでは，ツッコミ役が一時的にボケ役に転じることからツッコミでありながら同時にボケにもなるという特徴がある.「一つの漫才の中には，いろんな種類のボケが入っていた方がいい」[2] とのことから，本漫才にもできる限りボケの種類を増やすべきであり，その意味でノリツッコミを本漫才に取り入れる価値があると考えられる．本論文で行われるノリツッコミは，ボケ役の言葉遊びボケに対して適用し，読み間違えた別の単語の間違えにツッコミ役が同調し，その後ツッコミ役が自らその間違えを訂正するといった流れである．単語に同調する要素には様々あるが，本研究ではその単語の印象を用いる．ここで印象の抽出を簡略化するために，本研究では印象とはある単語を一つの形容詞によって表現したものとする．ある単語とは先の言葉遊びボケにおいて，読み間違えた単語のことである．先の例の場合「投票」という単語に対して読み間違えた「凍傷」という単語がこれに当たる．即ち，ノリツッコミの生成時の印象の抽出とは読み間違えた単語に関連した形容詞を抽出する．具体的には，印象を抽出する単語と共起する形容詞を検索結果のスニペットから抽出し，共起頻度の高い形容詞をその単語の印象とする．抽出された形容詞の例を表 1 に示す．例えば,「凍傷」というキーワードに対しての印象は表 1 より，最も共起頻度の多い「怖い」という形容詞が凍傷の印象となる．これにより「凍傷はホント怖いなー」という台詞になる．

表 1　キーワードと共起する形容詞

| query | 1st adjective | 2nd adjective | 3rd adjective |
|---|---|---|---|
| 凍傷 | 怖い | 寒い | 白い |
| 豆腐 | おいしい | 美味しい | やわらかい |
| 洋風 | かわいい | 明るい | おいしい |
| 賛辞 | すごい | 素晴らしい | うれしい |
| テレビ | 楽しい | めざまし | 幅広い |

#### 3.2.3 オ　　チ

オチでは，まとめとして台本生成の題材となったニュース記事の内容を 1 つのキーワードで簡潔に表現し，最後にそのキーワードをお題に自動生成した謎かけで笑いをとり締める．以下

に自動生成した漫才台本のオチの例を示す.

**ボケ**「以上!」
**ツッコミ**「もう終わりかいな!なんもわかってへんやん!」
**ボケ**「そんなことないしなー．要するに東京五輪の話やろ!」
**ツッコミ**「まぁせやけど，略しすぎや!」
**ボケ**「それなら，最後に東京五輪で一句よませてもらうわ.」
**ツッコミ**「ほぉ〜，やってみ.」
**ボケ**「整いました!」
**ツッコミ**「早いな!」
**ボケ**「東京五輪とかけて.」
**ツッコミ**「うん.」
**ボケ**「果物と解きます.」
**ツッコミ**「その心は … ?」
**ボケ**「どちらも夏季（柿）がつきものです.」
**ツッコミ**「 … もうええわ!」
**ボケ・ツッコミ**「ども，ありがとうございましたー.」

**謎かけ**

謎かけとは,「$A$ とかけて $B$ と解く．その心は，どちらも $C$（$C'$）がつきものです」といった形式で行われる一種の言葉遊びである．ここで，$C$ と $C'$ は互いに同音異義語の関係を持つ．謎かけの自動生成には $A$，$B$，$C$，$C'$ の 4 つの語を取得する必要がある．本論文での謎かけの自動生成の流れを以下に示す.

（1） $A$ を台本生成の題材となったニュース記事のタイトル中から 1 語設定する.

（2） $A$ と共起する単語を検索結果のスニペットから抽出し，共起頻度の高い単語を $C$ に設定する.

（3） $C$ の同音異義語を小学生の国語辞典コーパス中から抽出し $C'$ を設定する.

（4） $C'$ と共起する単語を検索結果のスニペットから抽出し，共起頻度の高い単語を $B$ に設定する.

$A$ は謎かけにおいて主題となる部分である．漫才台本の自動生成においてもオチの役割であるニュースのまとめの意味合いを担うことを考慮して，$A$ にはニュースの主題を設定することにより，謎かけを通してニュースの概要をユーザに印象付ける効果が期待される．そのためニュースの主題が顕著に現れるニュース記事のタイトル中から $A$ を設定する．先の例では『２０２０年「東京五輪」に決定』というニュース記事のタイトルから一般語をストップワードとして「東京五輪」を $A$ に設定する．次に，謎かけの定型句より $A$ と $C$ が互いに連想関係にあることに着目して $C$ を取得する．取得手法としては，ノリツッコミの印象抽出と同様の方法で $A$ と共起する単語を検索結果のスニペットから抽出し，共起頻度の高い単語を $A$ から連想される語として $C$ を抽出する.「東京五輪」の共起頻度の高い単語には「オリンピック」,「開催」,「夏季」,「プレゼンテーション」等の語が抽出される．$C$ に最も共起頻度の高い単語を設定した後，同音異義語を先の小学生の国語辞典コーパス中から抽出し $C'$ を設定する．同音異義語であるかどうかの判定は，$C$ をローマ字に変換して同じくローマ字の綴りが一致する語を同音異義語とする．$C$ の同音異義語がコーパス中から発見できない場合は，共起頻度が次いで高い単語から $C$ を再設定する．先の抽出の場合「オリンピック (orinpikku)」は同音異義語が見つからないので，$C$ を再設定して「開催 (kaisai)」の同音異義語を探す.「開催」の同音異義語には「快哉 (kaisai)」や「解砕 (kaisai)」等が考えられるが，今回用いたコーパスではわかりやすさを考慮しているためにこれらの難解な語は抽出されない.「夏季 (kaki)」に対しては「柿 (kaki)」が抽出されるため結果的に $C$ には「夏季」，$C'$ には「柿」がそれぞれ設定される．最後に，$B$ と $C'$ が $A$ と $C$ の関係と同様に互いに連想関係にあることに着目して $A$ から $C$ を抽出した方法と同様の方法で $C'$ から $B$ を取得し設定する.「柿」の共起頻度の高い単語には「果物」,「富有」,「産地」,「和歌山」等が抽出されるので $B$ には最も共起頻度の高い「果物」を設定する.

## 4. 対立ボケの生成

対立ボケとは，ある語に関して対照的な関係にある語を対立語と呼び，対立語を用いて行うボケである．以下に東京をキーワードとして本論文で提案する対立ボケの例を示す.

**ボケ**「2020 年夏季五輪の開催都市を決める国際オリンピック委員会総会が 7 日ブエノスアイレスで行われ，開催都市に東京を選んだ.」
**ツッコミ**「なるほど … ところで東京ってどんなかわかってるか?」
**ボケ**「あれやろ？近畿地方に属する日本の都道府県の一つやろ?」
**ツッコミ**「お前それ大阪と勘違いしてるやろ．東京は，日本の関東地方南部に位置し，1869 年 2 月 11 日以来，日本の首都が置かれている都市，地域や!」
**ボケ**「そうなんか，でも似たようなもんやろ」
**ツッコミ**「どこがやねん … 怒られるで！」

対立ボケでは対立語のような対照的な関係にある情報をユーザに提示することでユーザが記事を読むだけでは得られない知識の拡大が見込めるという利点がある．また，キーワードに関する知識をユーザが持っていなくとも，提示された対立語に関する知識をユーザが持っていればキーワードの情報をユーザがある程度推測可能になり内容把握の手助けになると考えられる．同時に対立語は語同士の関係性が直感的に把握しやすくボケとしても明確でわかりやすいと考えられる.

### 4.1 対立語の定義

本論文で対象とする対立語の定義について述べる．我々が対象とする対立語は，例えば「東京」に関しては「大阪」,「イチロー」に関しては「松井秀喜」,「サッカー」に関しては「野球」といったように 2 つの語同士が互いに対照的に，あるいはライバル的な関係性にある語のことである．それぞれの語の関

係に着目すると,「東京」と「大阪」は共に日本の都市,「イチロー」と「松井秀喜」は共に野球選手,「サッカー」と「野球」は共に球技であるといったようにそれぞれの語同士が共通の上位語を持っていることがわかる．そこで我々は共通の上位語を持つ同位語の中に対立語が含まれていると考えた．以下本論文では共通の上位語を持つ語のことを同位語と呼ぶ．また，野球選手を上位語に持つ「長島茂雄」も「イチロー」の同位語になり対立語である可能性があるが，野球選手であると同時に「イチロー」と同じメジャーリーガーでもある「松井秀喜」の方が「イチロー」との関係がより強く，対立語として適していると考えられる．そのため共通上位語の数も考慮する必要がある．さらに,「野球」の対立語候補のうち「サッカー」と類似競技であるために共通の上位語を複数持ち，有力な対立語候補になると考えられる語に「フットサル」が挙げられるが,「サッカー」と「フットサル」では競技人口を見ると「サッカー」の方が「フットサル」よりも圧倒的に普及しており「野球」の対立語には同程度に認知されている「サッカー」を対立語として選択する方が適切であると考える．

以上より，我々は対立語を，キーワードの同位語であり，より多くの共通上位語を持ち，且つ，同程度の認知度を持つ語と定義する．

### 4.2 対立語の取得手法

本節では，Web ニュース記事本文に含まれるある単語 1 つをキーワードとして，そのキーワードの対立語を発見し抽出する手法を提案する．例えば，ニュース記事のキーワードが「東京」である場合，その対立語の「大阪」を抽出する手法を提案する．提案手法では，大きく以下の 2 つの段階に分けて対立語を発見する．

（1） キーワードに関する同位語郡の取得と関連度による同位語郡のランキング.

（2） 同位語郡からのキーワードと認知度が近似する単語の取得.

以下，同位語の抽出及び重みを用いた関連度の計算手法について述べる．次に，語の認知度を Web 検索結果数を用いて取得し，キーワードの認知度との比率を計算する．これら 2 つの結果から対立語を発見する．

#### 4.2.1 同位語の取得と関連度の計算

同位語を取得する手法として Google Sets [15] がある．Google Sets は複数のキーワードの組み合わせを与えると，それぞれに関連性があると思われるキーワード郡を見つけて結果として返す．ただし，Google Sets は日本語に対応しておらずアルゴリズムも公開されていない．大島ら [16] は並列助詞「や」による Web 検索エンジンの組み合わせで Web 上から同位語を発見する手法を提案している．この手法では辞書等のデータベースを必要とせず瞬時に同位語を発見できるが，発見できる同位語の数が少ない場合がある．そこで我々は，キーワードに対する同位語郡を得るために，キーワードの上位語関係に着目し，ALAGIN フォーラム [17] から提供されている上位下位関係抽出ツールを用いる．これは Wikipedia から機械学習を使って上位下位関係となる用語ペアを数百万対のオーダーで抽出できるツールで，これを用いて Wikipedia 全記事の上位下位関係を抽出し，語の上位語階層コーパスとすることで同位語の抽出を行う．例えば,「東京」の上位語は表 2 に示すように日本の都市や就航地の他にも曲や作品いったように 11 語の上位語を取得できる．このようにキーワードの上位語は 1 つとは限らず複数個持つ場合があるが，これらのうち 1 つでも共通の上位語を持つ語はキーワードとの同位語として取得する．

得られた同位語とキーワードとの関連度に関して，共通の上位語を多く持つ同位語の方がよりキーワードとの関連度が強いと考えられる．また，どのような上位語でキーワードと同位語となっているかも考慮する必要がある．例えば，東京の上位語である「作品」は 120257 語の下位概念語を持ち，同様に東京の上位語である「日本の都市」は 24 語の下位概念語を持つが，同じ上位語でも少数の概念だけに含まれる上位語の方がより重要性が高いと考えられる．以上の考えに基づき，キーワードの上位語にそれぞれ重みの値を与え，取得した同位語の共通上位語の数だけ対応した重みを全て加算することで同位語の関連度を求める．

上位語の重みは，コーパス内にその語がどれほどの割合で含まれるかで表現する．即ち，上位語 $s_i$ の重み $Sta(s_i)$ は上位語の下位概念語総数 $n$ とコーパスの概念語総数 $N$ を用いて以下の式で求める．

$$Sta(s_i) = \log \frac{n}{N} \tag{1}$$

なお，コーパスの概念語総数 $N$ は今回用いたコーパスでは 2931465 語である．上記の式を用いて東京の上位語の重みを計算した値を表 2 に示す．下位概念語数の数が少ないほど高い重みが与えられるのがわかる．結果的にキーワードと同位語 $e_i$ の関連度 $Rel(e_i)$ は以下の式で与える．

$$Rel(e_i) = \sum_{i=0}^{n} Sta(s_i) \tag{2}$$

この時，キーワードと同位語の共通上位語の総数を $n$ とする．また，$Sta(s_i)$ は重みである．

表 2 東京の上位語とその下位語数による重み

| 上位語 $s_i$ | 下位概念語数 $n$ | 重み $Sta(s_i)$ |
|---|---|---|
| 日本の都市 | 24 | 11.7 |
| 就航都市 | 240 | 9.4 |
| 就航地 | 258 | 9.3 |
| グループ | 333 | 9.1 |
| 過去に存在した店舗 | 347 | 9.0 |
| オリジナルアルバム | 1100 | 7.8 |
| 曲 | 3717 | 6.7 |
| シングル | 19519 | 5.0 |
| 収録曲 | 35981 | 4.4 |
| 登場人物 | 95662 | 3.4 |
| 作品 | 120257 | 3.2 |

東京の同位語郡を式 2 を用いてランキングした結果の上位 10 語を表 3 に示す．ランキング順位が高い語は対立語になる可能性も高いと考えられる．

表 3　東京の同位語の関連度によるランキング

| 同位語 $e_i$ | 関連度 $Rel(e_i)$ |
|---|---|
| ニューヨーク | 35.1 |
| シドニー | 34.2 |
| ロンドン | 33.9 |
| パリ | 33.9 |
| 居酒屋 | 31.8 |
| バンクーバー | 30.7 |
| 北京 | 30.7 |
| ひまわり | 30.6 |
| 上海 | 26.3 |
| 大阪 | 24.5 |

**4.2.2**　認知度の計算と対比率

我々の提案する語の認知度は，検索結果数が多い語ほど，認知度が高い語と考え，Google 検索を用いてキーワードと同位語それぞれの Web 検索の検索結果数を取得することにより求める．

4.1 節でも述べたように，我々はキーワードと近い認知度を持つ同位語が対立語となると考え，語の認知度を語の検索結果数と見なし，キーワード $key$ の検索結果数 $Cog(key)$ と同位語 $e_i$ の検索結果数 $Cog(e_i)$ を用いて語の認知度の対比率 $Con(key, e_i)$ を以下の式で求める．

$$Con(key, e_i) = 1 - \log \frac{|Cog(key) - Cog(e_i)|}{max\{Cog(key), Cog(e_i)\}} \qquad (3)$$

ここで，$Con(key, e_i)$ の値が 1 に近いほどキーワードと同位語の認知度は近いと言える，反対に，$Con(key, e_i)$ の値が 0 に近ければ認知度に格差があると言える．

東京の関連度の高い上位 10 語の同位語の検索結果数 $Cog(e_i)$ と認知度の対比率 $Con(key, e_i)$ を計算した結果を表 4 に示す．なお，今回取得した東京の検索結果数 $Cog(key_i)$ は 324000000 であった．認知度の対比率 $Con(key, e_i)$ が高い語は対立語になる可能性も高いと考えられる．

**4.3　対立ボケ生成の流れ**

提案する対立語の取得手法を用いた対立ボケの自動生成の流れを以下に示す．

（1） ニュース記事本文を形態素解析して名詞を取得する．

（2） 取得した名詞の中から任意のキーワード $key$ を 1 つ設定する．

（3） 設定した $key$ の持つ上位語集合及び同位語郡を抽出する（4.2.1 節）．

（4） キーワードと同位語 $e_i$ の関連度 $Rel(e_i)$ を求める（4.2.1 節）．

表 4　同位語の検索結果数及び認知度の対比率

| 同位語 $e_i$ | 検索結果数 $Cog(e_i)$ | 対比率 $Con(key, e_i)$ |
|---|---|---|
| ニューヨーク | 16100000 | 0.05 |
| シドニー | 2750000 | 0.01 |
| ロンドン | 16200000 | 0.05 |
| パリ | 17600000 | 0.05 |
| 居酒屋 | 40700000 | 0.17 |
| バンクーバー | 2430000 | 0.01 |
| 北京 | 625000000 | 0.52 |
| ひまわり | 9050000 | 0.03 |
| 上海 | 1300000000 | 0.41 |
| 大阪 | 470000000 | 0.69 |

（5） $e_i$ の認知度 $Cog(e_i)$ 及び，$key$ の認知度 $Cog(e_i)$ との対比率 $Con(key, e_i)$ を求める（4.2.2 節）．

（6） $Rel(e_i)$ と認知度と $Con(key, e_i)$ の相乗平均から対立語候補をランキングする．

（7） ランキング 1 位の語を対立語 $antonym$ に設定し，$key$ と $antonym$ の Wikipedia 記事本文から $key$ と $antonym$ の概要 $key\_info$ 及び $antonym\_info$ をそれぞれ取得する．

$key$ に設定する語は階層構造の得られる Wikipedia 中に記事が含まれる語を設定する．$key$ の上位語によって同位語集合が得られるが，$key$ の同位語は $key$ の上位語の数によっては数万個と膨大な量に及ぶ．この同位語集合の中からより対立語らしい語を選出するために語の関連度と認知度の対比率の 2 つの値を用いて両方の値が共に高い語がより対立語らしい語という考えに基づき相乗平均によってランキングを行う．なお，相乗平均の計算の際に，同位語の関連度 $Rel(e_i)$ が最も高い値を 1 として正規化を行う．得られたランキングの 1 位の語を $antonym$ に設定し，$key\_info$ 及び $antonym\_info$ と合わせて以下のテンプレートに当てはめて対立ボケの台本を生成する．

**ツッコミ**「なるほど … ところで＜ $key$ ＞ってどんなかわかってるか?」
**ボケ**「あれやろ？＜ $antonym\_info$ ＞やろ?」
ツッコミ「お前それ＜ $antonym$ ＞と勘違いしてるやろ．
＜ $key$ ＞は，＜ $key\_info$ ＞や!」
**ボケ**「そうなんか，でも似たようなもんやろ」
**ツッコミ**「どこがやねん … 怒られるで！」

提案手法によって得られた対立語の結果の例を表 5 に示す．得られた同位語の中で $Rel(e_i)$ が最も高い語と $Con(key, e_i)$ が最も高い語と 2 つの値の相乗平均が最も高い語をそれぞれ表記する．キーワードから得られる上位語の数は様々であるが，基本的に上位語の数が多いほどより多くの同位語を得られる．まず，$Rel(e_i)$ が最も高い語を見ると,「バラク・オバマ」に対して同じアメリカ合衆国大統領であった「フランクリン・ルーズ

表 5　キーワードに関して取得された対立語

| キーワード | 上位語数 | 同位語数 | $Rel(e_i)$ 最大語 | $Con(key, e_i)$ 最大語 | 相乗平均 |
|---|---|---|---|---|---|
| 東京 | 13 | 264132 | ニューヨーク | 明星 | 大阪 |
| アメリカ | 10 | 371 | フランス | 韓国 | フランス |
| イチロー | 30 | 114920 | 新庄剛志 | 福山雅治 | 松井秀喜 |
| バラク・オバマ | 11 | 101950 | フランクリン・ルーズベルト | ジョージ・ワシントン | ジョージ・ワシントン |
| 与党 | 1 | 301 | 共産党/労働党/野党/etc | 野党 | 野党 |
| 野球 | 18 | 226471 | サッカー/バレーボール | サッカー | サッカー |
| チェス | 6 | 90481 | 将棋 | フィギュアスケート | オセロ |
| インターネット | 8 | 4134 | 新聞 | 雑誌 | 新聞 |
| マイクロソフト | 16 | 2454 | アップル | 東芝 | 東芝 |
| マクドナルド | 88 | 311653 | モスバーガー | キャノン | 吉野家 |

ベルト」が取得される等どの結果もキーワードに対して関連性の高い語が取得できていることがわかる．一方で $Con(key, e_i)$ が最も高い語は,「東京」に対して「明星」や「チェス」に対して「フィギュアスケート」等のように認知度が近くともそこに関連性を見つけることが困難な場合があり，$Con(key, e_i)$ が最も高い語にだけ注目しても意味があまりないと考えられる．相乗平均では，$Rel(e_i)$ が最も高い語や $Con(key, e_i)$ が最も高い語とは異なった語が取得されていることも多い．特に「与党」の場合，上位語が 1 語しか得られないために，$Rel(e_i)$ が最も高い語となる同位語が 301 語取得される．相乗平均により認知度を考慮したことで「与党」の対立語は「野党」であるということが一意に求めることが可能となる．同様に「野球」の対立語は「バレーボール」ではなく「サッカー」であるとして一意に取得される．また，語の認知度として用いた Web 検索の検索結果数は取得日によって結果数が異なることがある.「東京」の相乗平均による対立語が大阪となっているが，検索結果数によって $Con(key, e_i)$ が変化したことで，相乗平均による対立語に「北京」や「上海」が取得されることも見られ，認知度の計算には検索結果数以外の指標も必要になると考えられる．相乗平均により「バラク・オバマ」の対立語が「フランクリン・ルーズベルト」から「ジョージ・ワシントン」に変わっているが,「ジョージ・ワシントン」の方がより対立語として適しているかはさらに検証の必要がある．

## 5. まとめと今後の課題

本論文では，人とロボットとの円滑なコミュニケーションの実現を目的として，ロボットの漫才台本を Web のニュース記事から自動生成することを提案した．漫才台本をつかみ，本ネタ，オチに分割し，本ネタの部分にキーワードと対照的な語を提示することで情報視野の拡大等が期待できる対立ボケというボケの搭載を提案した．対立ボケの自動生成では，同位語の中に対立語が含まれることに着目し，キーワードの同位語郡を Wikipedia の階層構造を利用して取得した．さらにより対立語らしい語を同位語郡中から取得するために同位語の関連度と認知度の対比率にも着目してランキングを行い対立ボケの生成を行った．

今後の課題として，対立語の取得に制約があり，Wikipedia 記事中に含まれない語や上位語を多く持たない語の場合は対立語を取得できないという点である．これは設定したキーワードに応じた適切な対立語の取得手法を選択して対処していく必要がある．また，提案手法によって取得された対立語がキーワードの対立語として適切なものであるか検証の余地があり，同時に対立語の定義についてもさらに厳密に定める必要がある．

## 謝　　辞

本研究の一部は JSPS 科研費 24500134 の助成によるものです．生成した漫才台本の実演にあたり，中山弘隆氏らによって製作された 2 体のロボット，アイちゃんとゴン太にご協力頂きました．この場を借りて，深く御礼申し上げます．

## 文　　献

[1] 神田 崇行，石黒 浩，小野 哲雄，今井 倫太，中津 良平：人-ロボットの対話におけるロボット同士の対話観察の効果，電子情報通信学会論文誌 2002/7Vol.J85-D-I No.7,pp.691-700, 2002.
[2] 元祖爆笑王,-「漫才入門　ウケる笑いの作り方全部教えます」，リットーミュージック,2011 年.
[3] 北山 大輔, 角谷 和俊：ニュースアーカイブのためのコンテンツ構成順序を用いた比較ニュース検索, 電子情報通信学会第 18 回データ工学ワークショップ (2007), DEWS2007A9-4
[4] 佃 洸摂, 大島 裕明, 山本 光穂, 岩崎 弘利, 田中 克己：語の認知度と同位語間の関係に基づく意外な情報の発見,DBSJ Journal,Vol.11,No.3,2013
[5] S. Oyama and K. Tanaka: “ Query modification by discovering top- ics from web page structures ”, Proceedings of the Sixth Asia Pacific Web Conference (APWEB ’04) (2004).
[6] K. W. Church and P. Hanks: ”Word association norms,mutual information,and lexlcography”,Proc. of the 27th Annual Meeting of the Association for Computational Linguistics,pp. 76-83 (1998).
[7] P. D. Turney:”Mining the Web for synonymsl PMI-IR versus LSA on TOEFL”,Proc of the 12th European Conference on Machine Lerning(ECML 2001),pp.1725-1728(2004)
[8] Cilibrasi, R. L. and Vitanyi, P. M. B.(2007). The Google Similarity Distance. IEEE Transactions on Knowledge and Data Engineering, 19(3), 370-383.
[9] 関 綾子敏,-「おかしみの生成における言語操作の構造-漫才の資料として-」(『早稲田日本語研究』, 早稲田大学国語学会,2002 年.
[10] 安倍 達夫,-『漫才における「フリ」「ボケ」「ツッコミ」のダイナミズム-おかしみの構造図とその展開-』(『ことば研究会』, 人工知能学会第 2 種研究会ことば工学研究会資料,2005 年.

[11] ウィキニュース:スタイルマニュアル:http://ja.wikinews.org/wiki/
[12] Akiyo Nadamoto, Masaki Hayashi, Katsumi Tanaka”Web2Talkshow: Transforming Web Content into TV-Program-Like Content based on Automatic Transformation of Dialog”,Proc. of the 5th International Conference on Research, Innovation & Vision for the Future (RIVF ’07), pp. 1-6.,2007.
[13] 金水 敏,-「ボケとツッコミ-語用論による漫才の会話の分析-」(『上方の文化，上方ことばの今昔』大阪女子大学国文学研究室編）, 和泉書院,1992 年.
[14] Grice,H.P. ”Logic and Conversation” ,Syntax and semantics Vol, 3 in Cole and Morgan (eds.)
[15] Google Sets:http://labs.google.com/sets
[16] 大島 裕明, 田中 克己：両方向構文パターンを用いた Web 検索エンジンからの高速関連語発見手法 ,DBSJ Journal, Vol.7, No.3,2013.
[17] 高度言語情報融合フォーラム（ALAGIN）上位語階層データ:http://nlpwww.nict.go.jp/corpus/
[18] Kotaro Nakayama, ”Wikipedia mining for triple extraction enhanced by co-reference relation”, Proc. of the 1st International Workshop on Social Data no the Web(SDoW’08), 2008.