重畳型駄洒落ユーモアにおける常識的知識グラフを用いた潜在表現抽出

井上蒼一朗　谷津元樹　森田武史
青山学院大学理工学部情報テクノロジー学科
a5818014@aoyama.jp {yatsu,morita}@it.aoyama.ac.jp

概要

機械による言語的なユーモアの認識能力を向上させるため，駄洒落ユーモアのうち潜在的な表現を含む重畳型駄洒落の検出手法を提案する．提案手法は背景知識の獲得のため常識的知識を豊富に含む大規模知識グラフを活用する．本稿では，定性的評価実験において確認された，検出の成功事例，および失敗事例における主な失敗要因について説明する．

1　はじめに

近年機械がユーモアを表出・理解することに対する関心が高まっている．また，機械がユーモアな表現を表出・理解することでそれに接するユーザの生活の質を向上させることが報告されている．人間のコミュニケーションは，画一的な形式によらず自由であり駄洒落やなぞかけに代表される言語的なユーモア含んでいる．ユーモアな表現を機械が理解することによって人間同士の会話に近い自然な対応をすることができる．

1.1　研究課題と目的

ユーモアはジェスチャーなどを用いる身体的なユーモアと，駄洒落やなぞかけに代表される言語的なユーモアに大きく分類される．中でも駄洒落は比較的に認識が容易であり年齢層を問わず親しみやすいものと考えられる．知識グラフや自然言語処理技術を用いて駄洒落ユーモアの検出や理解の有効性を高めるためには下記の課題が存在する．後述する滝澤の研究 [1] によれば，駄洒落には併置型駄洒落と重畳型駄洒落の 2 つの分類が存在する．以下にそれぞれの例文と構造を示す．

併置型駄洒落

(1) **ふとんが吹っ飛んだ**

この例は「ふとん」と「吹っ飛ん」が，音韻的に類似しているため駄洒落であると認識することができる．

重畳型駄洒落

(2) **怪我がなくても大分県**

しかしこの例は併置型のように音の類似するペアがないことから従来の手法では検出が困難である．例文 (2) では，「怪我」と「大分」の間に何らかの潜在的表現を媒介する関係があると予測できる．その場合，外部の知識を用いることでその潜在的な表現及び関係を見出すことができる．すなわち，「『怪我』は『痛い』という性質を持つ」という関係性である．このとき，潜在的な表現である「痛い（イタイ）」は文内の「大分（オオイタ）」に音韻的に部分一致するので，上記の文を駄洒落文として認識することができる．本研究は重畳型駄洒落に存在するこのような潜在的な表現及び文内の語との関係性を後述する ConceptNet などの大規模知識グラフを用いて発見することを目的とする．

2　関連する研究と技術

はじめに，大規模知識グラフについて述べる．主要な大規模知識グラフとして著名なのが DBpedia[1] や Wikidata[2] であり，セマンティック Web 技術によるフリーの知識ベースであり，コミュニティベースで作成され，他のオープンデータとのリンクを持つリソースから構成されている．

ConceptNet[3, 4] とは DBpedia，Wikidata と同様に集合知を知識源とした大規模な知識グラフである．ConceptNet は，DBpedia や Wikidata のような RDF による表現に類似した構造を持つ．Turtle や N-Triples といった具象構文ではデータが構成されていないものの，RDF トリプルと同一視できる三つ組による抽象的な構造を表現している．4.4 節に述べる API は JSON-LD 形式により具象化された Linked Data との親和性の高い知識表現を返す．知識収集源は，

表 1　駄洒落データベース [荒木 2018] における併置型・重畳型駄洒落の収録件数及び割合

	併置型駄洒落	重畳型駄洒落
件数	67835 (98.4%)	1103 (1.6%)

WordNet，DBpedia，OpenCyc などの Linked Open Data に加え gamification に基づく知識収集サイトなど多岐にわたる．ConceptNet は主要な大規模知識グラフに比べるとトリプル数，リソース数は劣る (DBpedia との比較では，トリプル数 8500 万以上に対し約 2100 万，リソース数約 2.28 億に対し約 800 万) が，十分な規模を持つと考える．

次に，本研究で認識の対象とする駄洒落文を収載するデータセットについて述べる．駄洒落データベース [5, 6] はユーモアの面白さの評価に利用できる標準的なデータセットの確立を目指して構築された，Web 上の駄洒落を収録するコーパスである．Web 上に存在する駄洒落のクローリングを行い 68938 件の駄洒落文を収集・分類することにより構築された．併置型駄洒落と重畳型駄洒落は表 2 のように分類されている．最新の版 [6] では，3 名の被験者による面白さのスコアの付与が行われている．

駄洒落の検出・理解を目的とする手法の基礎に関わる提案は以前から行われている．滝澤 [7] は駄洒落を理解するシステムの構築の一環として併置型駄洒落と重畳型駄洒落を音素列や長さの一致などを基に分類し明確な基準を初めて提示している．また，併置型の駄洒落検出器の構成図を示しプロトタイピングを行っている．しかし，この論文では実際のシステムをなにも実装しておらず，併置型駄洒落のみの検出が対象であり重畳型駄洒落の検出についての手法の提案がなされていない．本研究では滝澤の分類に基づいて重畳型駄洒落を対象とする検出手法の提案を行う．

谷津ら [8] は本研究の目的と同様に，ユーザによる発話をユーモアとして検出することを目的に韻文ユーモアである駄洒落の教師あり学習に基づく検出手法を提案している．具体的には入力文中に音韻的に類似する 2 つの区間の有無を子音の音韻類似度を用いて判定し，bag of words とともに特徴量としている．この論文では駄洒落の型を分類して音韻類似を用いて駄洒落を検出しているが併置型駄洒落を対象にした特徴量を同様に重畳型駄洒落にも適用しているため重畳型駄洒落を正確に検出できていない．そこで本論文では重畳型駄洒落の構造に着目した検出手法を新たに提案する．

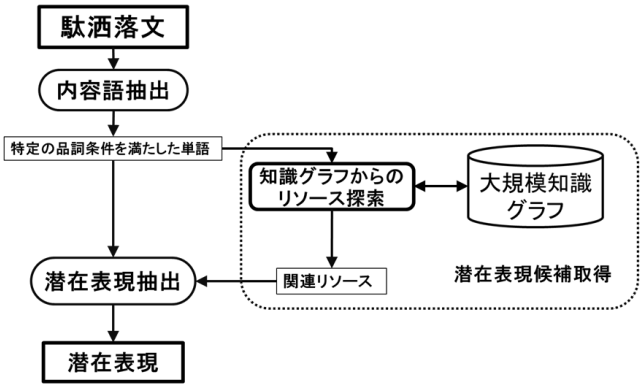

図 1　システム構成図

3　提案

提案システムの構成を図 3 に示す．システムは内容語抽出部，潜在表現抽出部及び潜在表現候補取得部から構成される．駄洒落文が入力されると内容語抽出部 (3.1 節) を用いて形態素解析が行われ，内容語形態素が得られる．次に潜在表現抽出部 (3.2 節) が，潜在表現候補取得部 (3.3 節) を用いて内容語形態素ごとに知識グラフから関連リソースを探索し，リソースの探索に成功した場合に編集距離ベースの音韻類似度関数を用いて潜在表現の抽出を試みる．潜在表現の抽出に成功した場合は重畳型駄洒落の検出に成功となり，失敗した場合は他の関連リソース・内容語形態素について再度探索をする．すべての内容語形態素及び関連リソースについて潜在表現が抽出できなかった場合，重畳型駄洒落の検出は失敗となる．

3.1　内容語抽出部

内容語とは，名詞，形容詞及び動詞などにおいて実質的かつ語彙的な意味を持つ語を指す．GiNZA[] を用いた形態素解析を行っている．内容語のみを抽出するために，品詞を次のカテゴリー，すなわち **名詞-普通名詞**, **名詞-固有名詞**, **動詞-一般**, **感動詞-一般**, **副詞-一般**, **形容詞-一般** のいずれかに属するものに制限した．また，計算量の削減及び単独で意味をなさない記号を排除するためにストップワードを用いている．次のストップワードを指定した：[ ] ( ) - / . ， = は の が に.

3.2 潜在表現抽出部

潜在表現抽出部では，内容語に対応する関連リソースより潜在表現として適切なものを抽出する．潜在表現として適切であるためには，内容語と関連リソースのラベルとの間に音韻類似性が成立する必要がある．音韻類似性を検出するためには 4.5 節に述べるレーベンシュタイン距離に基づく類似度関数を用いる．関連リソースを抽出できるまでリソースの取得を反復する必要があるが，繰り返しの実装方法として幅優先探索を基にした再帰関数を用いる．再帰関数においては，潜在表現候補の取得，音韻類似度関数による潜在表現の判定，検出を行えなかった場合の探索対象の拡大，及び探索深さの制限による終了条件のチェックを行う．本モジュールの疑似コードを付録 A に示す．

3.3 潜在表現候補取得部

潜在表現候補取得部では，潜在表現抽出部において指定された内容語に対応する関連リソースまたは指定の内容語の関連リソースにさらに関連するリソースの探索を大規模知識グラフにおいて行う．潜在表現抽出部の `find_latent_expressions` 関数において `depth` が 1 のときに `find_latent_expression_candidates` 関数に与えられた引数 `org_res_label` に相当する内容語のリソースは関連リソースに含まないものとする．例えば，「卵」という内容語のリソースに対し「鮭」という関連リソースが取得されて，潜在表現抽出部における再帰によって（depth=2 となる）さらに「鮭」より関連リソースを取得する場合「卵」が取得可能であっても，内容語リソースと一致する「卵」は取得しない．本研究では大規模知識グラフとして ConceptNet 5[?] を使用する．ConceptNet 5 はオープンソースソフトウェアとして Web API を備えたサーバーの実装を Github に公開している[1)]．提案手法では大規模知識グラフの知識源としてこの API を用いる．本モジュールの疑似コードを付録 B に示す．

4 評価と考察

提案手法を用い，駄洒落データベース [5] に収録された 1103 件の重畳型駄洒落のうち，3 名の面白さスコアの平均値が 2.66 を上回ることを要件として絞り込んだ 131 件のうち，さらに潜在表現が明らかに常識的知識として存在すると判断可能な 22 件を対象に検出実験を行った．その結果，ConceptNet 中に関連表現は存在するが，音韻類似や言語の表記体系の違いにより検出できない例が 9 件確認された．

4.1 事例別の考察

本節では，認識の成功及び失敗の事例を対象に個別的に定性的評価を行い，今後の手法改良に繋がると考えられる発見について述べる．

4.1.1 成功例文

・この鮭の卵，いくら？

この例の内容語抽出を行うと「鮭，卵，いくら」になる．それぞれの単語の関連リソースを探索し自身の親リソース以外の内容語形態素結果と音韻類似の比較を行う．すると，「いくら」の関連リソースに MadeOf（いくら – MadeOf - 卵）という関係性で「卵」が存在し音韻類似の比較を行った結果，入力文中の「卵」と完全一致する．したがって，「卵」という言葉が潜在的な語として抽出することが可能となり検出成功となる．

・食べ物が体の中でかくれんぼ．もう，胃ーかい?

内容語抽出を行うと「食べ物，体，中，かくれんぼ，胃」になる．探索した関連リソースと左記の各内容語形態素との音韻類似の比較を行うと，「胃」に対し PartOf（胃 – PartOf – 体）という関係性で存在する「体」が完全一致する．したがって，「体」を潜在表現とした検出が成功する．

・煙とともに灰さようなら

内容語抽出を行うと「煙，灰」になる．探索した関連リソースと左記の各内容語形態素との音韻類似の比較を行うと，「灰」に対し RelatedTo（灰 – RelatedTo – 煙）という関係性で存在する「煙」が完全一致する．したがって，「煙」を潜在表現とした検出が成功する．

4.1.2 音韻類似の検出漏れや表記体系の違いによる失敗

・一番頭を使う作業は農作業！

内容語抽出を行うと，「頭，使う，作業，農作業」となる．「頭」という単語から「脳」という単語は取れているがそれらを音韻類似の比較を行う際に潜在表現として検出できていない．原因として，音韻類似の比較の仕方が編集距離ベースであるため，文字数が異なる単語（この例では「脳」→「ノウ」と「農

1） https://github.com/commonsense/conceptnet5

作業」→「ノウサギョウ」）を比較すると部分的には一致していても類似度の値（式 (4．5．1) により，
sim(“脳”，“農作業”) = 0．25）が文字数が一致する場合よりも著しく低くなってしまう．

・これを運ぶの？うん，そう．」 <運送>

内容語抽出を行うと，「運ぶ，うん」となる．「運ぶ」の関連リソースに「貨車」，「貨車」の関連リソースに「運送」があるが，音韻類似度比較の対象が形態素単位であるため，「うん，そう」のように途中に読点がある場合違う単語として処理されてしまい，音韻類似の比較を行うことができない．今後の課題として文全体の読みあるいはその部分との音韻類似の比較を可能にする必要があると考えられる．

4.1.3 ConceptNet 中の関連リソース不足による失敗

・オリンピックなんて，銅でもいい．

「オリンピック」という言葉の ConceptNet 上の関連リソースにおいて「銅」という言葉が存在しなかった．Wikidata を探索した場合正解に相当する関連リソースとして発見されたため，異なる知識グラフについての実装は今後の課題といえる．

・天気予報によると，雨だす！

「天気予報」という言葉の ConceptNet 上の関連リソースにおいて「アメダス」という言葉が存在しなかった．DBpedia を探索した場合正解に相当する関連リソースとして発見されたため，異なる知識グラフについての実装は今後の課題といえる．

4.1.4 関連リソースが想定できるものの取得が困難な例

・虫退治は，アースとあさって

ConceptNet，DBpedia や Wikidata いずれの大規模知識グラフ中にも「虫退治」に用いられる商品という意味での「アース」が関連リソースに存在しないため，潜在表現候補の抽出ができない．

5 おわりに

本研究の目的は，大規模知識グラフである ConceptNet を用いて潜在的な音韻類似の対を含む駄洒落文を検出することである．目的を達成するための提案手法として，ConceptNet を用いて各単語の関連リソースを取得しそれぞれの単語との音韻類似度を求めることで，しきい値以上となった単語を潜在表現をとして抽出した．研究を進めていく中で，関連リソースの探索の仕方や音韻類似度の比較において，手法を見直さなければならない点がいくつか見受けられた．

5.1 今後の課題

4 節の考察を経て得られた課題は 4 つである．1 つ目は音韻類似度の比較を行う際に比較対象となる単語の文字数が異なる場合正確な比較ができないという点である．比較の方法が編集距離ベースであるため文字数が異なる場合，音韻的には近くともしきい値が低くなってしまい，関連がない単語として処理されてしまう．今後は，音韻類似度の比較を文字数に関わらず子音の類似度を測ることのできるシステムを実装することで 1 つ目の課題を解決することができると考える．2 つ目の課題は英語表記及びローマ字表記の単語の比較ができないという点である．関連リソースの探索を行うと，実際に WEB 上ではローマ字表記の単語を経由すれば理想の単語にたどり着くことができたが，ローマ字表記の単語はひらがなに変換することができないためしきい値を超えることができない．今後は，英語表記及びローマ字表記の単語をひらがなに変換する方法を考え構築する必要があると考える．3 つ目は関連候補の出現数上限の制限により求める関連リソースの取得が困難，という点である．潜在表現候補の上限がデフォルト値で 20 件として指定されており，API に渡すクエリパラメータを指定して調整することは可能であるが上限数を増やすと他の例文の潜在表現検出結果が正解と異なる結果となる．そのため，潜在表現候補の上限地値の適切な指定方法を構築することが今後の課題といえる．4 つ目は例文中に読点がある場合違う単語として処理され，音韻類似の比較を行うことができないという点である．本研究の対象とした例文に，読点の前後の文字列を結合すると単語として成立するケースが存在したが，単語として処理することができないことから潜在表現の抽出は失敗となった．今後は，読点を省く処理を行いさらに前後の文字列をつなぎ合わせることで 1 つの単語として見なすことができるような仕組みを考えることが必用であると考える．

謝辞

本研究は JSPS 科研費 (基盤研究 (C)21K12007) の助成を受けたものである．

参考文献

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. **Journal of Web Semantics**, Vol. 7, No. 3, pp. 154–165, 2009.

[2] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. **Communications of the ACM**, Vol. 57, No. 10, pp. 78–85, 2014.

[3] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In **Thirty-first AAAI conference on artificial intelligence**, 2017.

[4] Robyn Speer and Catherine et al. Havasi. Representing general relational knowledge in conceptnet 5. In **Proceedings of the 8th International Conference on Language Resources and Evaluation**, pp. 3679–3686, 2012.

[5] 荒木健治, 佐山公一, 内田ゆず, 谷津元樹. 駄洒落データベースの拡張及び分析. 人工知能学会第 2 種研究会 ことば工学研究会資料, SIG-LSE-B803-1, pp. 1–15, 2018.

[6] 荒木健治, 佐山公一, 内田ゆず, 谷津元樹. 駄洒落データベースを用いた面白さの評価及び分析. 人工知能学会第 2 種研究会 ことば工学研究会資料, SIG-LSE-B902-4, pp. 63–76, 2019.

[7] 滝澤修. 記述された「併置型駄洒落」の音素上の性質. 自然言語処理, Vol. 2, No. 2, pp. 3–22, 1995.

[8] 谷津元樹, 荒木健治. 子音の音韻類似性及び svm を用いた駄洒落検出手法. 知能と情報 (日本知能情報ファジィ学会誌), Vol. 28, No. 25, pp. 833–844, 2016.

A　find_latent_expression 関数の疑似コード

```
Function find_latent_expresion
Input (org_res_label, res_list, cwml, depth, latent_expression_cand_dict)
Output (latent_expressions, latent_expression_cand_dict)

latent_expressions と related_res_list を初期化
if depth > 2:
    return latent_expressions, latent_expression_cand_dict
for r in res_list:
    lecd, rr_list = find_latent_expression_candidates(r, org_res_label, depth)
    for key_word, val_set in lecd.items():
        if key_word が空でない and latent_expression_cand_dict のキーに key_words が存在しない:
            latent_expression_cand_dict[key_word] = val_set
        else:
            latent_expression_cand_dict[key_word] を val_set との和集合とする
    related_res_list に (rr_list) を加える
    for cwm in cwml:
        latent_expressions = detect_latent_expression(cwm, latent_expression_cand_dict)
    if latent_expressions が空でない:
        return latent_expressions, latent_expression_cand_dict
for i=0 to length of related_res_list-1 :
    if org_res_label が空文字列である:
        org_res_label = cwml[i]
    return find_latent_expression(org_res_label, related_res_list[i], cwml,
                                  depth+1, latent_expression_cand_dict)
return [], {}   /* 探索中断 */
```

B　find_latent_expression_candidates 関数の疑似コード

```
Function  find_latent_expression_candidates
Input  (res,org_res_label)
Output  (latent_expression_cand_dict,related_res_list )

辞書 latent_expression_cand_dict とリスト related_res_list を初期化
E = 知識グラフから res が含まれる全てのトリプルを取得
for e in E:
    subject, relation, object = e
    if searching_res_label が latent_expression_cand_dict のキーに存在しない:
        latent_expression_cand_dict[org_res_label] = {}
    if res == subject & object のラベルが日本語である:
object を latent_expression_cand_dict[org_res_label] と related_res_list に追加
    else if res == object & subject のラベルが日本語である:
subject を latent_expression_cand_dict[org_res_label] と related_res_list に追加
return latent_expression_cand_dict, related_res_list
```