5P-3

格要素を用いたテンプレートによる若者言葉の自動抽出

松尾 朋子† 安藤 一秋‡
香川大学大学院工学研究科† 香川大学工学部‡

1. はじめに

近年，一般家庭にもインターネットが普及し，誰でも簡単に Web 上で情報発信が可能となった．特に若者（10～20 代）は，日常会話で利用している若者特有の言葉（若者言葉：10～20 代までの若者が Web 上でよく使用する言葉）をブログや Twitter などでも使う傾向がある．若者言葉は，使用期間が短く，新しい言葉が造られやすい特徴がある．また，若者言葉は日本語の文法や規則から逸脱したものもあり，若者言葉に親しみのない世代には，若者言葉の意味を理解できない場合がある[1-5]．例えば，若者言葉に親しみのない世代が若者言葉で書かれたブログを読む場合，若者言葉の意味を理解できず，ブログの内容を理解できない場合がある．そこで，ブログの内容を理解するためには，若者言葉の意味が記載されている書籍や Web サイト，検索エンジンを利用して若者言葉の意味を調べる必要がある．しかし，若者言葉の意味が記載されている書籍や Web サイトでは若者言葉の収集や意味の調査を人手で行っているため網羅性に欠ける．

そこで本研究では，Web から若者言葉を自動収集し，それらの意味を推定する手法の実現を目的とする．著者らは，ベースライン手法[1]として，カタカナで記述される既存の若者言葉（種言葉）の直前・直後の形態素に注目したテンプレートを用いて，種言葉と同種のカタカナ若者言葉が自動抽出できる可能性について検討した．その結果，種言葉の直前・直後の形態素のみを利用したテンプレートでは，文脈的な制約が弱いため，種言葉と同種の若者言葉を抽出することが難しいことを確認した．

本稿では，種言葉と同種のカタカナ若者言葉を抽出するために格要素に注目し，文脈を考慮したテンプレートベクトルを利用して若者言葉を自動抽出する手法について提案する．

2. 格要素を用いたテンプレートベクトル

我々の先行研究[1]では，カタカナで記述された既存の若者言葉を種言葉とし，その直前・直後の形態素のみを利用したテンプレートを用いて，若者言葉の抽出実験を行った．しかし，種言葉の直前・直後の形態素のみを利用したテンプレートでは，文脈的な制約が弱いため，種言葉と同種の若者言葉を抽出することは難しいことが分かった．そこで本稿では，格要素に注目することで，文脈情報を加味したテンプレートベクトルによる若者言葉の抽出を目指す．

一般的に，意味的に類似した若者言葉は類似した文脈で利用されやすい傾向があるといえる．例えば，以下の 2 つの若者言葉について考える．

① みんなのテンションもジョジョに**アゲアゲ**で楽しかった！
② みんなのテンションもジョジョに**アゲポヨ**で楽しかった！

①の「アゲアゲ」と②の「アゲポヨ」は共に「テンションが上がっている状態のこと」を意味する若者言葉であり，それぞれ「楽しい」に係っている．また，「楽しい」が取りうる格も類似している．

したがって，同じ文脈で利用される若者言葉が動詞の場合，類似した格要素を取る可能性が高いと考えられる[2]．また，名詞の場合，同じ格要素として類似した動詞に係りやすく，また，その動詞が取りうる他の格要素も類似している傾向があると考えられる．

そこで本稿では，まずは動詞に係るタイプのカタカナ若者言葉を種言葉に設定し，若者言葉が係る動詞とその動詞が取りうる格要素の情報を用いてテンプレートベクトルを構築する手法について提案する．本稿で対象とする格要素は，鈴木ら[2]と同様，文脈の特徴が表れやすいと考えられる「ヲ格」，「デ格」，「ニ格」とする．

3. 格要素を用いたテンプレートベクトルによる若者言葉の自動抽出

3.1 テンプレートベクトルの構築

種言葉と同種のカタカナ若者言葉を抽出するため，格要素を用いたテンプレートベクトルを構築する手法について述べる．なお，ここで述べるテンプレートベクトルは，動詞に係りうるカタカナ若者言葉を抽出するものである．

種言葉 $s_i \in S$ に対するテンプレートベクトル $t_{ij} \in T$ の構築手順を以下に示す．

Automatic Extraction of Young People's Words based on Templates using Case Components
†Tomoko Matsuo，Graduate school of Engineering, Kagawa University
‡Kazuaki Ando，Faculty of Engineering, Kagawa University

手順 A-1：「種言葉 s_i +格助詞（ヲ，ニ，デ）」のヒット数の取得
「種言葉 s_i +格助詞（ヲ，ニ，デ）」でブログ検索し，ヒット数を取得する．
手順 A-2：ブログ記事の収集
手順 A-1 の結果を基に，ヒット数が最大の「種言葉 s_i +格助詞 cp_{ij_max}」を含むブログ記事を収集する．
手順 A-3：動詞の頻度取得
収集したブログ集合から「種言葉 s_i +格助詞 cp_{ij_max}」（=格要素 cc_{ij_max}）を含む一文 s_k を取り出して係り受け解析し，cc_{ij_max} が係る「動詞」の頻度をカウントする．
手順 A-4：テンプレートベクトル $t_{ij} \in T$ の構築
頻度が n 以上の「動詞 v_{ij}」を対象に，v_{ij} が取りうる残りの格要素を文 s_k から取得し，格要素 cc_{ij_max} と合わせてテンプレートベクトル t_{ij} の要素とする．重みは，tf-idf 値とする．

3.2 若者言葉候補の抽出

構築したテンプレートベクトルを用いて種言葉と同種の若者言葉候補を抽出する手法について説明する．
以下に，若者言葉候補を抽出する手順を示す．
手順 B-1：テンプレートベクトル $t_{ij} \in T$ を用いてブログ記事の収集
t_{ij} の「動詞 v_{ij}」をクエリーとしてブログ検索し，ブログ記事を収集する．
手順 B-2：カタカナ言葉の取得
手順 B-1 で収集したブログ記事から「動詞 v_{ij}」を含む文 s_k を係り受け解析し，もし t_{ij} の「格助詞 cp_{ij_max}」を含む格要素 cc_{ij_max} がカタカナ言葉であれば，その言葉を取得する．
手順 B-3：一般語のフィルタリング
手順 B-2 で得られたカタカナ言葉が一般辞書に含まれていれば，一般言葉としてフィルタリングする．一般言葉でない場合，若者言葉候補として収集する．フィルタリングには，三省堂 Web Dictionary，Weblio 辞書，Yahoo! 辞書，kotobank の 4 つの辞書を用いる．
手順 B-4：残りの格要素の取得とベクトル生成
手順 B-3 で収集した若者言葉候補を対象に，手順 B-2 と同じ文 s_k から「動詞 v_{ij}」が取りうる cc_{ij_max} 以外の格要素を取得する．そして，cc_{ij_max} と共にベクトルを生成する．
手順 B-5：若者言葉候補の取得
手順 B-4 で生成したベクトルが t_{ij} と類似していれば，手順 B-2 で抽出したカタカナ言葉を若者言葉候補と判定する．なお，類似度計算は，Jaccard 係数とコサイン類似度を利用する．

3.3 年代別検索を用いた選別

goo の年代別検索を用いて，若者言葉候補の中から若者言葉を選別する手法について説明する．
年代別検索（10 代～50 代）を用いた若者言葉の判定手順を以下に示す．
手順 C-1：各年代における登録記事数の推定
「の」を用いて年代別検索し，各年代のヒット数を得る．得られたヒット数を各年代の登録記事数とする．
手順 C-2：若者言葉候補を用いた年代別検索
若者言葉候補を用いて年代別検索し，各年代でのヒット数を得る．
手順 C-3：使用割合の計算
手順 C-1 で得られた「の」のヒット数と手順 C-2 で得られた若者言葉候補のヒット数を用いて年代別の使用割合を求める．
手順 C-4：若者言葉の判定
若者（10 代～20 代）と他年代（30 代～50 代）の若者言葉候補の使用割合を比較し，若者の使用割合が大きい若者言葉候補を若者言葉として出力する．

4. おわりに

本稿では，種言葉と同種のカタカナ若者言葉を抽出するために，格要素を用いたテンプレートベクトルの構築法とそれを用いた若者言葉候補の抽出手法，年代別検索を用いた若者言葉の選別手法について提案した．
提案手法は，まだ提案段階であるため，今後は構築したテンプレートベクトルを用いてカタカナ若者言葉を自動抽出し，提案手法の有効性を評価する．

参考文献

[1] 松尾他，“若者言葉の自動抽出に用いたテンプレート改良に関する検討”，第 12 回情報科学技術フォーラム，第 2 分冊，pp.89-90，2013.
[2] 鈴木他，“リムる・ドヤる・ポジる・パフェる―Web を用いたカタカナ動詞の言い換え・語源の獲得―”，情処学研報，2012-NL-209(8)，pp.1-7，2012.
[3] 秋田他，“文字種と画数を用いた新若者語の抽出”，言語処理学会第 19 回年次大会（NLP2013）予稿集，2013.
[4] 原田他，“若者語処理システムの評価”，電子情報通信学会技術研究報告，AS-4-2，"S-37"-"S-38"，2006.
[5] 松本他，“感情推定における若者言葉の影響”，言語処理学会第 17 回年次大会，pp. 846-849，2011.