Vol.2015-NL-223 No.1
2015/9/27

統計値だけに基づくことを特徴とするキーワード抽出システムの新規実装と評価

手島亮太[†1]　吉田光男[†2]　岡部正幸[†3]　梅村恭司[†2]

文書からキーワードを抽出する方法の一つに，形態素解析などを用いず反復度と呼ばれる統計量のみを使った方法がある．この反復度を用いた方法では，キーワードとともによく用いられる「の」や「は」などの付属語も併せて，一つのキーワードとして特定されることが誤抽出の主な原因の一つである．そこで本稿ではキーワードに残りやすい付属語を特定することで，付属語によるキーワードの誤抽出を減らす抽出手法を提案する．この手法では，キーワードの前後に位置する語の出現回数を数えた上で，キーワードの前後には付属語が多く出現するという特性を利用することで，キーワードに残りやすい付属語の特定を行う．これに併せてキーワード抽出におけるしきい値の決め方についても議論する．また，実験により提案手法の F 値が従来手法に比べ向上することを確認した．

1. はじめに

日本語では単語の分かち書きが行われないことから，日本語の処理を行う場合に辞書を用いた形態素解析を利用することが一般的である．しかし，形態素解析のような辞書を用いた方法には,分割の精度が辞書に依存することから，新語や未知語に対応するために辞書の更新が欠かせないという問題がある．

これに対し，辞書を利用せずにテキスト中の文字列の統計量を用いることで，キーワードの抽出を実現するものがある．その一つとして，文献[1]では反復度という統計量を用いることでキーワード抽出を実現している．しかし，文献[1]では，高頻度で出現する文字列は「自立語は高いdf_2/dfを持ち，付属語は低い値を持つ」という特性が失われ高い反復度となることが確認されており，高頻度文字列である「の」などの付属語がキーワードに残ることが報告されている．そこで本稿では，キーワードの前後の文字列に着目することで付属語を特定し，それを用いることで分離することの出来なった高頻度文字列を含むキーワードの除去を行う．このときのキーワード抽出における処理全体で辞書を使わない．そして実験により，反復度を用いたキーワード抽出における付属語の特定とそれを用いたキーワードの抽出方法の有用性を示す．

2. 準備

本研究では，辞書などの語彙データを用いない方法でキーワード抽出を実現する．そのために，語の候補として極大部分文字列を取り扱い，その中からキーワードを見つけるための統計量として反復度を用いる．本章では，それぞれの定義について述べる．

2.1 極大部分文字列

極大部分文字列[2]とは，文書中の出現位置が全て同じである部分文字列による集合を考えたとき，その集合において一番長い文字列のことを指す．ここで，極大部分文字列は部分文字列間の半順序関係における極大元として以下のように定義されている[3].

> 文字列Tの i 番目から j 番目までの部分文字列をT_{ij}としたとき，Tの相異なる部分文字列t_1, t_2の間に$t_1 < t_2$の関係が成り立つことは次式を満たすことである．ただし，l_nはt_nの長さとする．
> $0 \leq \exists k \leq l_2 - l_1, \forall i \ s.t. \ T_{i(i+l_1)} = t_1 \Rightarrow T_{(i-k)(i-k+l_2)} = t_2$

以上の定義から分かるように極大部分文字列は，出現位置が一致する部分文字列集合を代表する文字列である．ここで，出現位置が一致することは出現回数も一致するということであり，それらの集合に属する文字列は同じ統計量を有することになる．よって，代表である極大部分文字列のみをキーワードの候補として取り扱うことにする．

2.2 反復度

反復度（*adaptation*）は語のもつ特徴量の一つで，語の繰り返しの度合いは語の意味に依存するという性質を表すものである[4]．反復度は，ある文書に文字列xが 1 回以上含まれていることを条件とした時に，その文書に文字列xが 2 回以上含まれている場合を考えた条件付き確率の推定値である．ここで，ある文字列が文書に出現する確率及び 2 回以上出現する確率は，文書集合全体では文書頻度を用いて推定することができる．したがって，反復度は次のように計算することができる．

$$adaptation(x) = \frac{p(e_2(x))}{p(e_1(x))} \approx \frac{df_2(x)}{df(x)} \qquad (1)$$

ここで，$e_1(x)$は文書が文字列xを 1 回以上含む事象，$e_2(x)$は文書が文字列xを 2 回以上含む事象，dfはコーパス全体で文字列xを 1 回以上含む文書数，df_2はコーパス全体で文字列xを 2 回以上含む文書数とする．

また，反復度についての日本語および中国語における調査から以下の特徴が報告されている[1].

- キーワードは高い反復度を持つ
- キーワードの反復度はdf/N（Nは全文書数）に対して，

†1 豊橋技術科学大学大学院情報・知能工学専攻
†2 豊橋技術科学大学情報・知能工学系
†3 豊橋技術科学大学情報メディア基盤センター

ほぼ無相関である
- 自立語の終端と付属語の境界に敏感に反応し，自立語の境界特定に有用である

なお，ここでのキーワードとは調査対象の文書に対して著者が付与したものであり，キーワードは自立語に属するものとして扱っている．

3. 反復度を用いたキーワード抽出法

従来手法であるキーワードらしさを表すスコアとして反復度を用いたキーワード抽出手法[1]について述べる．3.1 節で実際の抽出手順について述べ，3.2 節で従来手法および従来手法を基にした手法の問題点について述べる．

3.1 従来手法のキーワード抽出の手順

従来手法では，文章分割とキーワードのふるいわけ作業の 2 つのステップで構成されている．文章分割では，キーワードらしさのスコアに基づき，自立語の境界を一つの単位として文章分割する．分割した結果には助詞なども含まれているため，キーワードらしい文字列のみを残す処理を行う．

3.1.1 従来手法の単位分割

単位分割のために用いるキーワードらしさは式(2)で定義されている．

$$Score_{adaptation}(x_i) = \begin{cases} -\infty & \wedge \quad df_2 < 3 \\ \log 0.5 & \wedge \quad df_2 \geq 3, \dfrac{df}{N} > 0.5 \\ \log \dfrac{df_2(x_i)}{df(x_i)} & \wedge \quad df_2 \geq 3, \dfrac{df}{N} \leq 0.5 \end{cases} \tag{2}$$

ここで，上段はサンプル数が少ない文字列（$df_2 < 3$）に対するスコア，中段は高頻度文字列（$df/N > 0.5$）に対するスコア，下段はそれ以外の文字列に対するスコアである．サンプル数が少ない文字列では反復度のふるまいが不安定になるため，スコアを$-\infty$としている．高頻度文字列は反復度による語の種類の分別性が失われるため，それらの文字列が単体で分割されてしまわないようにスコアを$\log 0.5$に抑えている．

上記のスコアを用いたビタビアルゴリズムにより，テキスト全体におけるスコアが最大となる分割を求める．これは長さ n の文字列に対し$O(n^2)$であり次式で定義される．

$$viterbi\_adaptation = \underset{X}{\operatorname{argmax}} \left(\sum_{X=\{x_1, x_2, \cdots, x_n\}} Score_{adaptation}(x_i) \right) \tag{3}$$

3.1.2 従来手法のキーワードの特定

3.1.1 節の方法で分割された文字列が，キーワードであるかどうかの判定は(4)式で行われる．ここで，$length$ は文字列xの長さである．

$$keyword = \left\{ x \middle| 0.00005 < \frac{df(x)}{N} < 0.1, length > 1 \right\} \tag{4}$$

3.2 従来手法の問題点

従来手法において抽出失敗例として以下の 4 つが挙げられている[1].

- 英数字による語が不完全になるもの
- 「・」によって文字列が切られるもの
- 「～の」や句読点などが末尾に残るもの
- 末尾の 1 文字が削れるもの

これらの問題は，自立語と付属語の境界推定に失敗したために起きた問題であるが，境界を前に（語として短くなる方に）推定したか，それとも後ろに（語として長くなる方に）推定したかによって性質は大きく異なる．

「「・」によって文字列が切られる」，「末尾の 1 文字が削られる」という 2 つの失敗例は，自立語の境界を前に推定した失敗例である．すなわち，本来はより長い自立語であるにも関わらず短く切られたものである．一つの原因は，反復度が境界に反応する性質上，自立語だけでなく，その部分文字列もまた高い反復度を持つことである．これに対し文献[5]では，自立語の部分文字列において，単語の境界でのみ高い値となるように式(2)のスコアを補正する方法（以後，従来手法 2）を提案した．別の原因として，文書中での反復出現数が十分でないために反復度をうまく推定できていないことが挙げられる．これに対し文献[6]では，類似する文書による文書拡張を行う方法を提案した．

上記の失敗例に対して，「「～の」や句読点などが末尾に残る」という失敗例は自立語の境界を後ろに推定した失敗例である．これは助詞や句読点など，文書において頻繁に使われる文字列が，自立語とともに一つのフレーズとして抽出されたものである．すなわち，付属語も含めたものが一つの自立語として認識されていることから，スコアの補正や文書拡張とは異なるアプローチが必要となる．特に文書拡張を使う方法では，拡張により同じ付属語が繰り返し使われることで，キーワードに助詞や読点が付属しやすくなることが報告されている．

従来手法において，自立語と付属語の境界を誤って前に推定してしまう問題に対してはいくつかの改善案が提案されている．しかし，問題の性質の異なる問題，つまり境界を後ろに推定してしまう問題に対してはその改善案がないままであり，キーワードに付属語が残るという問題に対する何らかのアプローチが必要だと言える．

また，しきい値の決め方の問題もある．従来手法や改善手法[5]において，多くのしきい値を利用する．そして，それぞれのしきい値は対象の特性（例えば文書長など）に依存する．このことについて，文献[1]で，式(2)の$\log 0.5$（式中段）の値は文書長に依存すると言及しており，式(4)のキ

ーワード条件も経験則と実験結果に基づいた「ベストではないがキーワードらしさを表す確かな値」として用いていることが述べられている．したがって，今一度しきい値の導入について議論することが望ましいと考えられる．

4. 提案手法

本稿では，極大部分文字列に対して以下の手順で処理することでキーワードを特定する．

(1) 反復度をしきい値としたキーワード抽出
(2) 付属語の特定と利用
(3) 出現文書数をしきい値としたキーワードの除去
(4) 他のキーワード候補の部分文字列であるキーワードの除去

(1)の処理では，反復度を用いることで付属語である極大部分文字列を除去し，キーワードの候補となる自立語を抽出する．次に(2)の処理では，直前の処理において正しく除去されずに自立語に「の」や「は」などの付属語が残ってしまった極大部分文字列の除去を行う．(3)の処理では，反復度の振る舞いが不安定であるサンプル数が少ない文字列の除去を行う．最後に(4)の処理では，自立語の部分文字列である極大部分文字列の除去を行う．これは文献[1]にあるように，自立語の部分文字列もまた高い反復度を有していることに起因する処理である．

4.1 反復度をしきい値としたキーワードの抽出

キーワードとなる自立語は高い反復度を持つことから，反復度によるしきい値を用いてキーワード候補を抽出する．ここで，単語の候補となる文字列の集合，すなわち極大部分文字列集合を X としたとき，キーワード候補集合 X_1 は次式となる．

$$X_1 = \left\{ x \in X \middle| \frac{df_2(x)}{df(x)} \geq 0.3 \right\} \qquad (5)$$

4.2 付属語の特定と利用

文献[1]では，反復度が自立語の境界を特定するための特徴量として有用であることが示されおり，前節の結果の多くは正しく自立語と付属語の境界で分割されていることが予想される．したがって提案手法では，反復度によって分割された自立語と付属語の境界にある文字列を特定することで付属語の特定を行う．付属語の特定は，反復度がしきい値以上の文字列としきい値以下の文字列の差分を利用した文字列の数え上げと，数え上げ結果を基に付属語を特定するという 2 つのステップから構成される．

4.2.1 付属語の数え上げ

自立語に付属語が残ってしまう問題は高頻度文字列に起因する問題であるため，自立語に残りやすい付属語は付属語として出現する回数も多いことが予想される．したがって，自立語とともによく現れる付属語特定のために，反復度がしきい値以上である文字列としきい値未満である文字列の差分から付属語の出現数の数え上げを行う．

数え上げの例を表 4-1 に示す．ここで，*cf* はすべての文書におけるその文字列の出現数の総和である．表 4-1 では反復度がしきい値以上である文字列「ロボット」に対し，しきい値未満である「ロボットに」から「に」が付属語として 50 回現れ，「ロボットには」から「には」が 25 回現れたとしてカウントしている．さらに，「ロボット工学に」に対して「ロボット工学には」から「は」が付属語として 5 回カウントされる．例には存在しないが同じ文字列が付属語として判定された場合は，その文字列の付属語としての出現回数は累積していくことになる．

表 4-1 の例では「ロボット工学に」がしきい値以上であると判定されているが，これは自立語として考えたとき正しくない．ここで先の例で「に」は 50 回も付属語として現れており，(表中の情報では) 付属語として多くの回数現れている．この情報を用いることで「ロボット工学に」−「に」=「ロボット工学」が正しい自立語であると思い直すのが提案手法の着眼点である．

図 4-1 に後方に残る付属語数え上げのアルゴリズムを示す．ここで，数え上げ結果は付属語をキー，出現回数をバリューとしてもつハッシュテーブル C とし，取り扱う文字列集合は X，しきい値以上の文字列集合は X_1 とする．また「重複カウントしてしまう要素を排除する」とは「のロボットに」と「ロボットに」のように，テキスト上で同じ出現位置である付属語を複数回カウントしないための動作で，後方一致する文字列を排除する動作である．なお，接頭辞文字列を接尾辞文字列に，後方一致を前方一致に，置き換えることで前方に残る付属語の特定アルゴリズムとなる．

表 4-1 付属語の数え上げ動作の例

キーワード候補	*cf*	しきい値判定	動作
⋮			
ロボット	100	○	
ロボットに	50	×	「に」の回数(50)をカウント
ロボットには	25	×	「には」の回数(25)をカウント
ロボット工学	30	○	
ロボット工学に	10	○	
ロボット工学には	5	×	「は」の回数(5)をカウント
⋮			

```
BEGIN
W ≔ 空のハッシュテーブル
Xr ≔ X - X1
FOR word in Xr DO
    IF X1に wordの接頭辞文字列が存在する THEN
     mlp ≔ X1で最も長い wordの接頭辞文字列
     suffix ≔ word と mlpの差分文字列
     IF Wに suffixをキーとするものが存在する THEN
         W[suffix]に wordを追加
     ELSE
         W[suffix] ≔ 空のリスト
         W[suffix]に wordを追加
     ENDIF
    ENDIF
ENDFOR
Wから重複カウントしてしまう要素を排除する
C ≔ 空のハッシュテーブル
FOR key→suffix, value→list in W DO
    C[suffix] ≔ 0
    FOR word in list DO
     C[suffix] ≔ C[suffix] + wordの出現回数
    ENDFOR
ENDFOR
END
```

図 4-1 付属語数え上げのアルゴリズム

4.2.2 付属語の特定

本稿では，付属語としての出現回数が多い順に並べ付属語となる回数の下限を決めた．なお，片仮名は付属語にならないと考えて，初めて現れた片仮名の出現回数を付属語の出現回数の下限とするヒューリスティックスを採用した．これは 4.2.1 で述べたように，自立語とともによく現れる付属語は出現回数が多いという仮定に基づいた特定方法であり，一般的に付属語にならない片仮名が現れるまでを付属語として特定している．図 4-2 に付属語特定のアルゴリズムを示す．ここで特定結果は後方付属語集合 S である．

```
BEGIN
S ≔ φ
sortedC ≔ Cを出現回数でソート
FOR key→suffix, value→cf in sortedC DO
    IF suffixが片仮名でない THEN
     S ≔ S ∪ {suffix}
    ELSE
     BREAK
    ENDIF
ENDFOR
END
```

図 4-2 付属語特定のアルゴリズム

4.2.3 付属語を先頭・末尾にもつキーワードの除去

自立語に付属語が残るという問題に対し，付属語が先頭あるいは末尾に残る自立語をキーワード候補から外すという対処を行う．これにより，付属語が残っている自立語として正しくないものがキーワードとして選ばれることを阻止する．特定した前方付属語集合を P，後方付属語集合を S とし，ある文字列xの接頭辞集合をP_x，接尾辞集合をS_xとしたとき（P_x, S_xにxは含まれない），4.1 節のX_1を用いて付属語によるフィルタリング結果X_2は次式で表される．なお，式中の∄は存在記号∃の否定を表している．

$$X_2 = \{x \in X_1 | \nexists p(p \in P_x \wedge p \in P) \wedge \nexists s(s \in S_x \wedge s \in S)\} \quad (6)$$

4.3 出現文書数をしきい値としたキーワードの除去

振る舞いが不安定であるサンプル数の少ない文字列を取り除くために，出現文書数をしきい値としたフィルタリングを行う．提案手法では，そのしきい値を出現文書数$df = 4$とした．フィルタリング結果X_3は次式で表される．

$$X_3 = \{x \in X_2 | df \geq 4\} \quad (7)$$

4.4 他のキーワード候補の部分文字列であるキーワードの除去

本稿では高頻度文字列と自立語の部分文字列が反復度のしきい値条件を満たしてしまうという問題を解決するために他のキーワード候補の部分文字列であるキーワードを除去する．ここで，2 つの文字列において，「ロボッ」に対する「ロボット」のように一方を完全に含む文字列を上位文字列とし，ある文字列xの上位文字列による集合をU_xとしたとき，抽出キーワードは次式で表される．

$$\text{KEYWORD} = \{x \in X_3 | \nexists u \ s.t. \ u \in X_3 \wedge u \in U_x\} \quad (8)$$

5. 提案手法の妥当性

提案手法では，片仮名による付属語の特定や他のキーワード候補の部分文字列を候補から除去するなど，一つ一つを見ると一見不思議に思える処理がある．実際には，それらの処理は新たなしきい値の導入を避けながら，従来手法の抱える問題を改善するように働きかけている．そして，それをシンプルに実現するための方法の一つに過ぎない．以下に，これらの処理の具体的な作動原理について述べる．

5.1 反復度をしきい値としたキーワードの除去について

文献[1]によると，日本語においても自立語の反復度は付属語の反復度に比べて高いという性質が成り立つこと，そして NTCIR1 などの論文抄録において著者が論文に付与したキーワードの反復度はおよそ 0.3~0.5 であることが報告されている．したがって，提案手法ではこの反復度をしきい値とすることでキーワードとなる自立語の抽出を行う．具体的には，キーワードの再現性を重視し，反復度が 0.3 をしきい値とする．

従来手法では，反復度をしきい値としては用いていないが，単語分割のためのスコア式(2)でキーワードらしさとして利用している．ただし文献[1]では，高頻度あるいはサンプルが少ない文字列は反復度の振る舞いが不安定になることが報告されており，スコア式ではそれらの文字列で分割されないようにスコア修正を施している．提案手法では，サンプルが少ない文字列に対しては 4.3 節の処理を，高頻度文字列に対しては 4.4 節の処理を行うことで対処している．

5.2 付属語の特定について

付属語の特定方法の一つとしては，ユーザがキーワードを基にシステムに正解を教え，機械学習を行う方法が考えられる．すなわち，付属語としての出現回数が多い順に語を並べて，ある出現回数になるか，あるいは特定の文字列が現れるまでを付属語として特定するといった方針において，そのしきい値や特定の文字列を機械学習によって用意するというものである．それに対し，本稿では付属語としての出現回数が多い順に並べたときに，初めて片仮名が現れるまでを付属語として特定するといった方法を用いている．これは一見すると辞書を用いたストップワードのように見えるが，実際にはユーザがキーワードの正解をシステムに教え込むことで「少なくとも片仮名はキーワードの一部である」として学習されるという仮定に基づいた方法である．

5.3 付属語によるキーワードの除去について

正しく付属語が特定出来ていれば，キーワードの先頭や末尾から付属語を削るという処理も考えられる．それに対し本稿では，削るのではなく，付属語を先頭・末尾に持つものを候補から外すという処理を行っている．これは，自立語に付属語が残った文字列が反復度のしきい値条件をクリアしている場合，一般的に自立語単体もまた反復度のしきい値条件をクリアしているからである．したがって，単純にキーワード候補から外すだけで自立語に付属語が残る問題は対処することが出来る．

日本語文章において助詞や句読点は高い頻度で使われるものであり，反復度を利用したキーワード抽出では，それらの付属語をどう取り扱うかは大きな問題である．提案手法では予め付属語を特定し付属語を取り去ったキーワードのみ残すことで，従来手法の抱える付属語の問題を改善している．

5.4 出現文書数をしきい値としたキーワードの除去について

文献[1]において，出現数の少ない文字列は反復度が高くなるなどその振る舞いが不安定であること，さらに，そのような文字列の数が膨大であることが述べられている．それらの少数サンプルの文字列をキーワードの候補から外すために，本稿では出現文書数によるしきい値を設け，出現文書数dfが 4 未満のものをキーワード候補から外すという処理を行う．

サンプル数が少ない文字列の問題に対し，従来手法では$df_2 < 3$を，提案手法では$df < 4$をしきい値としている．2 つのしきい値は厳密には同じ効果をもたらすものではないが，どちらもサンプル数が少ない文字列の除去を行っているという点で共通した処理であること，さらに本手法では文献[7]にあるものと同じ処理を 4.4 節で行うことから文献[7]のしきい値を採用している．また，従来手法では式(4)で，df/Nが 0.00005 未満のものはキーワードとしては無視しており，これは本手法によるしきい値よりも大きな値である（最も小さなコーパスでさえ$df < 4.58$に相当する）．したがって，文献 1 で実験した全てのコーパスにおいて，本手法が設けたしきい値よりも大きなしきい値によって無視されていることを考えると従来手法よりも緩やかな制限だといえる．

5.5 他のキーワード候補の部分文字列であるキーワードの除去について

この処理ではキーワードの長い方を残すことから，キーワードと同じように高い反復度を持つ自立語の部分文字列を候補から外すことが出来る．ここで，自立語の境界を誤って後ろに推定していた場合，誤ったキーワードのみが残ることになるが，本手法では自立語に残りやすい付属語を特定し付属語によって誤って長く推定されないようにしていることから，有効な方法だといえる．また，付属語が自立語に残っていたのと同じように，高頻度文字列もまた他のキーワードの一部となることが多いため，高頻度文字列についてもキーワード候補から除去できると考えられる．

一方で，この方法には正しいキーワードも候補から外してしまうという問題がある．先に挙げた表 4-1 の例で言えば，「ロボット」は「ロボット工学」によって候補から外されることになる．もちろん，「ロボット」のように一般的なキーワードが必要な場面も多いことから，そういう場面では本手法の利用は慎重にならなければならない．この問題を避けながら自立語の部分文字列の問題に対応する方法として，3.2 節で述べた文献[5]の方法があるが（文献[6]は本手法と同じく複合語が優先される），その場合，対象に依存したしきい値が新たに導入されることになる．したがって，今回は文献[6]と同じく長いキーワードの方が専門的で重要な情報を持つと考え，単純に長いキーワードのみを残すことにした．

この方法は文献[7]において，部分文字列によって統計量が合わさる（例えば，「京都」には「東京都」の統計量が合わさってしまう）という問題に対策するために導入されたものである．故に，本手法でも同じようにキーワード間における不正な統計処理を防ぐことが出来る．これは，従来手法にはない利点である．

6. キーワード抽出におけるしきい値

従来手法と提案手法のそれぞれで用いるしきい値についてまとめたものを表 6-1 に記す．ここで，*length* はキーワードの長さ，不等式は満たすべき条件であり，等式は当該値がその値になることを表している．この表からは，従来手法ではキーワード判定に用いるしきい値が多く制約が厳しいこと，さらに高頻度文字列によって誤って分割されないようにしきい値が設けられていることが分かる．すなわち，従来手法では，キーワードを抽出するにあたって多くのしきい値を利用している．それに対し，本手法では，従来手法のキーワードのしきい値と高頻度文字列のしきい値を廃し，低頻度文字列を除去するために 1 つ，キーワード候補となる自立語を特定するために 1 つ，計 2 つのしきい値しか設けていない．さらに注目すべきは，しきい値を減らすだけでなく,従来手法の抱える付属語の問題を改善し，自立語の部分文字列や高頻度文字列の問題に対しても対応している点である．ただし，現時点において，付属語の特定は機械学習を想定したストップワードによる実装である．

表 6-1 しきい値の比較

	低頻度	キーワード	高頻度
従来手法	$df_2 < 3$	$0.00005 < \frac{df}{N} < 0.1,$ $length > 1$	$\frac{df}{N} > 0.5,$ $Score = \log 0.5$
提案手法	$df < 4$	$\frac{df_2}{df} \geq 0.3$	

7. 実験

従来手法[1]と従来手法を改善した手法（従来手法 2）[5]，そして提案手法の 3 つの手法についてキーワード抽出結果の比較を行う．実験には NTCIR-1J コレクション[8]を用いた．これは日本語論文の抄録で構成された情報検索のためのテストコレクションであり，332,918 件の文書で構成される．また，それぞれの文書には著者が付けたキーワード（以後，著者キーワード）が付属している．

7.1 キーワード抽出の一例

はじめに，手法によって得られるキーワードの違いを確認することを目的として，手法ごとの抽出結果の一例を図 7-1 に示す．ここで，抽出元は NTCIR-1J コレクション全件について，各文書をタイトルとアブストラクトで構成したものであり，図 7-1 は結果の一つについて，抽出キーワードを角括弧で挟み，太字と下線で表現したものである．

[移動マニピュレータ]の軌道制御 (第 1 報:衝突回避を考慮した制御)
[移動マニピュレータ]に目標作業軌道が与えられたとき、作業に適した[移動ロボットの軌道]を生成する手法を提案する。マニピュレータの作業性と移動ロボットの動き易さに加え、更に変化する環境との衝突回避も考慮した評価関数を設定し、車輪の回転角速度空間において[最適軌道]を探索する。

移動マニ[ピュレータの][軌道制御] (第 1 報:[衝突回避]を考慮した制御)
移動マニピュレータに目標[作業][軌道]が与えられたとき、[作業]に適した移動ロボットの軌道を生成する手法を提案する。[マニピュレータの][作業]性と[移動ロボット]の動き易さに加え、更に変化する[環境]との衝突回避も考慮した評価関数を設定し、[車輪]の回転角速度[空間]において最適軌道を[探索]する。

移動マニピュレータの軌道制御 (第 1 報:[衝突回避]を考慮した制御)
移動マニピュレータに目標[作業][軌道]が与えられたとき、[作業]に適した移動ロボットの軌道を生成する手法を提案する。[マニピュレータの][作業]性と[移動ロボット]の動き易さに加え、更に変化する[環境]との衝突回避も考慮した評価関数を設定し、[車輪]の回転角速度[空間]において最適軌道を[探索]する。

図 7-1 キーワード抽出の一例
(上段：提案手法，中段：従来手法 1，下段：従来手法 2)

このサンプルでは，従来手法は「マニピュレータの」のようにキーワードに付属語が残っているが,提案手法では,そうした複合語の問題を避けられていることが分かる．また，従来手法では「軌道」や「作業」などの短いキーワードが多いが，提案手法では長いキーワードを優先していることから複合語や助詞を含む長いキーワードなど従来手法に比べて得られるキーワードが長いことが分かる．

7.2 著者キーワードによる性能評価

NTCIR-1J コレクションの各文書のタイトルとアブストラクトを一つの文書として構成した 332,918 件の文書集合からキーワードを抽出し，重複を取り除いた著者キーワード 376,969 件を正解として評価した結果を表 7-1 に示す．ここでの評価尺度として，次に示す式(9)の適合率，式(10)の再現率，式(11)の F 値を用いる．

$$\text{適合率} = \frac{\text{正解数}}{\text{抽出数}} \tag{9}$$

$$\text{再現率} = \frac{\text{正解数}}{\text{著者キーワード数}(=376969)} \tag{10}$$

$$\text{F 値} = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}} \tag{11}$$

表 7-1 の結果から再現率ならびに適合率と再現率の調和平均である F 値に関しては，再現率が 6%に対して 11%，F 値が 10%に対して 15%と，それぞれ提案手法が従来手法を上回る結果が得られた．一方で，適合率は従来手法 1 が 24.9%，従来手法 2 が 28.7%であるのに対し，提案手法は 23.2%と少し劣るという結果になった．

F 値と再現率が良くなった理由の一つとして，しきい値制約が緩くなったことで多くの語がキーワードとして抽出

されるようになったことが考えられる．また，それに加えて，提案手法は長いキーワードを選ばれやすくしているため「移動マニピュレータ」のような複合語が選ばれるようになり，従来手法では得られなかった専門的な用語が抽出できるようになったことで再現率が高まり，F 値でも勝る結果につながったと考えられる．

一方で，適合率が下がった理由として，一つに従来手法に比べて提案手法の制約が緩やかであること，もう一つに助詞を含む長いキーワードが正解として評価されないとことが考えられる．一つ目は，従来手法は特定の範囲のdf/Nを持つ文字列のみをキーワードとするなど提案手法に比べて厳しいしきい値制約があることに加え，それらを用いたスコア計算とビタビアルゴリズムによる文章の単位分割により尤もらしいもののみが選ばれるようにしていることから，従来手法の方が提案手法に比べて誤抽出が少ないのだと考えられる．なお，しきい値によってキーワードが絞られていることは,提案手法のキーワード抽出数が 18 万件であるのに対し，従来手法 1 と従来手法 2 はそれぞれ 9 万件と 8 万件でおよそ半分になっていることから分かる．二つ目は，文章の単位分割をしない提案手法では「移動ロボットの軌道」や「多眼カメラを用いた高精細画像」のような付属語を含むなどした長いキーワードが抽出されることがあるが，このようなキーワードは著者の定義するキーワードにないため誤抽出として判定されることで適合率が低くなっていると考えられる．ただし，これらの付属語を含むなどした長いキーワードには，人間が読む上では直感的に内容を理解できるものも含まれているおり，それらを評価することができれば実質的な適合率はより高いことが予想される．

7.3 少ない文書数での著者キーワードによる性能評価

前節では 332,918 件全てを使って評価を行ったが，常に十分なだけのデータを用意できるとは限らないことから，文書数が少ない場合についての振る舞いを調査する．本実験では,限られた数の文書集合を再現するために NTCIR-1J コレクションを 16 分割，すなわち 20,807 件の文書集合 16 組とし，それぞれからキーワード抽出を行う．評価尺度に前節と同じものを用い 16 組の文書集合の結果の平均と標準偏差を表 7-2 に示す．

表 7-2 より提案手法が従来手法 1 及び 2 に比べて，適合率が 1 よりも 6%，2 よりも 3%高く，再現率ではそれぞれより 0.5%，F 値でも 0.9%高い結果が得られた．これは前節の結果に加えて適合率の面でも上回ったということである．

適合率についても上回った理由の一つとして，文書数が減ったことで従来手法にあるしきい値の制約が緩まり，それぞれの手法で得られるものが近くなったことが考えられる．提案手法は，付属語による誤抽出を減らす取り組みをしているため，従来手法が提案手法と同一のしきい値であれば，提案手法が適合率の面で優位に働くと考えられる．

表 7-1 NTCIR-1J コレクション全件による性能評価

	抽出数	正解数	適合率	再現率	F 値
提案手法	184,047	42,781	23.2%	11.4%	15.3%
従来手法 1	92,159	22,945	24.9%	6.1%	9.8%
従来手法 2	79,795	22,932	28.7%	6.1%	10.0%

表 7-2 16 分割した NTCIR-1J コレクションによる性能評価

手法	指標	平均	標準偏差
提案手法	適合率	40.23%	0.32%
	再現率	2.36%	0.02%
	F 値	4.46%	0.04%
従来手法 1	適合率	34.09%	0.31%
	再現率	1.88%	0.02%
	F 値	3.56%	0.03%
従来手法 2	適合率	37.16%	0.35%
	再現率	1.87%	0.02%
	F 値	3.57%	0.03%

8. まとめ

本稿では，辞書を用いることなく反復度によってキーワードを抽出し，そのキーワードの前後の文字列に着目することでキーワードに残りやすい付属語が特定できることを示した．そして，キーワードに付属語が残ってしまうという問題に対し，特定した付属語を用いることでキーワードの誤抽出を減らす手法を提案した．また実験により提案手法が従来手法のしきい値を減らした上で，F 値が向上することを確認した．

参考文献

1) 武田善行, 梅村恭司, “キーワード抽出を実現する文書頻度分析,” 計量国語学会, vol. 23, No.2, pp. 65-90, 2001.
2) 岡野原大輔 , 辻井潤一,“全ての部分文字列を考慮した文書分類,” 自然言語処理研究会, NL187, pp. 59-64, 2008.
3) 中谷秀洋, “極大部分文字列を使った twitter 言語判定,” 言語処理学会 第 18 年次大会 発表論文集, pp. 547-550, 2012.
4) K. W. Church, “Empirical Estimates of Adaptation: The chance of Two Noriegas is closer to p/2 than p2,” COLING, 1, pp. 180-186, 2000.
5) 田中路子 , 梅村恭司,“キーワードの境界推定のためのポテンシャル関数,” 情報処理学会研究報告, vol. 2002, No. 20, pp. 97-104, 04 03 2002.
6) 長町健太, 武田善行 , 梅村恭司, “文書拡張によるキーワード抽出,” 自然言語処理, vol. 14, No. 1, pp. 67-86, 10 01 2007.
7) 手島亮太, 岡部正幸, 梅村恭司, “検索結果の絞り込みのために有用な語集合の特定,” 第 20 回言語処理学会発表論文集, pp. 137-140, 2014.
8) 神門典子, 栗山和子, 野末俊比古,大山敬三, “NTCIR-1：情報検索システム評価用テストコレクション構築の方針と実際,” 情報処理学会研究報告, pp. 33-40, 1999.