辞書構築のための普遍的な言語規則に基づく単語抽出法

土田正明　　水口弘紀　　久寿居大
NEC サービスプラットフォーム研究所
{m-tsuchida@cq,hironori@ab,kusui@ct}.jp.nec.com

1　はじめに

自然言語処理は，情報検索，情報抽出，情報要約など，膨大な情報を有効に利用するための基盤技術として重要となってきている．自然言語処理では，日本語など分かち書きされていない言語を対象とする場合，形態素解析が必要である．形態素解析では，辞書に未登録な語である**未知語**が含まれる文を正しく処理することは難しい．文を正しく解析するためには未知語が存在しないように辞書を強化する必要がある．しかしながら，人手で未知語を発見して辞書をメンテナンスする作業は高コストである．

また，近年，インターネット上で誰もが簡単に記事を配信できるようになり，口語調の表現や多様な話題を含む文書が増加している．増加に伴い，評判分析など，それら文書を解析するニーズが高まっているが，未知語を多く含むため正しく解析することが難しく，未知語問題の解決はより重要になってきている．

これまでにも，未知語抽出を目的とした研究 [6, 1, 5, 4, 3] は存在するが，多くは未知語文字列の抽出が対象であり，辞書に登録するために必要な，原形，品詞，活用規則の獲得までは考慮されていない．

本論文では，低コストに辞書を強化するため，日本語テキストから原形，品詞，活用規則の同定も含めて単語を抽出することで未知語を発見する方法を提案する．

2　課題

これまで，解析対象の多くは，新聞記事など文体が整った文書であったため，未知語の多くは名詞であった．一方，近年増加しているブログなど崩れた文体の文書には,「カワいい」といった意図的に表記を変えた語や,「メモる」などの造語が多く用いられる．このように，未知語抽出では，活用語にも対応することが重要となってきている．

従来の代表的な単語抽出法では，活用のある未知語を抽出することが難しかった．代表的な方法には，i) 形態素解析，ii) 文字種分割，iii) Ｎグラム，がある．ただし，それぞれには問題がある．i) は，見かけ上解析が成功する語を検出できない問題がある．例として，Juman[2] では,「昨日のメモった？」は,「昨日 / の / メモ / っ(未知語) / た / ？」となり,「メモった」を活用語と認識できない．ii) は，複数の文字種からなる語が抽出できないという問題がある．iii) は，活用を考慮しないため,「メモった」と「メモる」などが別々の単語として認識されてしまうという問題がある．また，Ｎグラムの頻度に基づき単語を抽出する場合，活用語は活用毎に別単語と見なされるため抽出されにくい．つまり，代表的な方法では，活用のある未知語を抽出することは難しい．

本研究の課題は，名詞など非活用語に加え，従来難しかった活用のある未知語も含めて抽出することである．次節より，具体的な方法を説明する．

3　提案方式

提案法は大きく２つからなる．

1. **単語候補抽出**：文字種，活用，付属語といった文体に普遍的な言語規則に注目して単語候補を抽出する
2. **単語判定**:各単語候補の文字数，文字種，抽出頻度，収集された活用の種類や数，を用いて単語らしさを評価し，同時に品詞と活用の種類を推定する

１）は，文字種，活用，付属語といった普遍的な情報に基づいて処理するため，文書に依らず安定して動作する．活用規則を利用しているため，活用の違いを吸収することで単語候補の出現頻度を適切に数えることができる．また，活用規則が分かれば品詞と原形も決まるため，原形，品詞，活用規則の情報を含めて単語候補を抽出できる．

一方で，１）の結果には，単語分割の荒さや活用規則，品詞，原形の推定ミスにより，単語としてふさわしくない候補も数多く含まれる．

そこで，２）により，各候補が単語であるか否かを判定する．以上の構成で，非活用語と活用語の両方を対象に，原形，品詞，活用規則を含めた単語抽出が実現できる．以降，それぞれの詳細を述べる．

3.1　単語候補抽出

単語を抽出するため，予備調査として語の構成パタンを検討した．結果，多くの語は，１）同一文字種（日本，ジャパン，etc...）か，２）非ひらがな文字列からひらがな文字列（小さい，ヤバい，etc...），のどちらかで構成されていることが分かったため，提案法では，上記の２タイプの語をターゲットとする．

１）の単語候補は，文字種で分割することで抽出できる．２）の多くは用言であり，語幹と活用で構成されているため，非ひらがな文字列の直後が活用であるかを調べることで，単語候補を抽出できる．

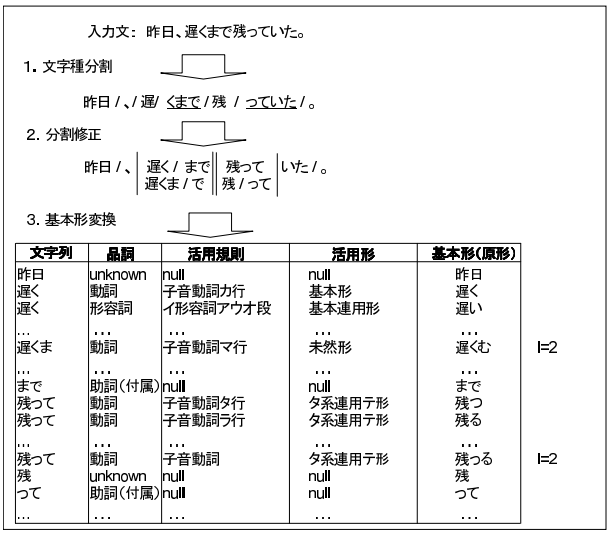
上記より，単語候補抽出は，以下の３ステップで構成されている．

1. **文字種分割**：文字種の変化に基づき分割

2. **分割修正**：ひらがな文字列の前と後に，付属語か活用が存在するかチェックし，文字種分割による分割結果を修正する

3. **基本形変換**：活用で修正された部分に対して，全ての可能な活用により基本形に変換し単語候補としてデータベースに蓄積する

以降，図 1 を用いて各処理を説明していく．活用規則は，Juman[2] のものを用いる．



文字列	品詞	活用規則	活用形	基本形(原形)	
昨日	unknown	null	null	昨日	
遅く	動詞	子音動詞カ行	基本形	遅く	
遅く	形容詞	イ形容詞アウオ段	基本連用形	遅い	
…	…	…	…	…	
遅くま	動詞	子音動詞マ行	未然形	遅くむ	l=2
…	…	…	…	…	
まで	助詞(付属)	null	null	まで	
残って	動詞	子音動詞タ行	タ系連用テ形	残つ	
残って	動詞	子音動詞ラ行	タ系連用テ形	残る	
…	…	…	…	…	
残って	動詞	子音動詞	タ系連用テ形	残つる	l=2
残	unknown	null	null	残	
って	助詞(付属)	null	null	って	
…	…	…	…	…	

図 1: 単語候補抽出処理の流れ

文字種分割　文字種を，ひらがな，カタカナ，漢字，数字，アルファベット，記号と定義する．基本的には，文字種が変化する部分で文分割する．ただし，例外として「ー，～（長音）」「っ，ッ（促音）」のみが同一文字種に挟まれている部分では分割しない．これは，長音や促音から語が始まることはないためである．この例外処理により,「肝っ玉」「スーツケース」などが分割されず，より適切な単語候補抽出のための文字種分割ができる．

分割修正　分割修正は，漢字，カタカナ，アルファベットの直後のひらがな文字列に対して，以下のステップで行う．

1. **前方修正**：パラメタ $l = 1, 2$ のそれぞれで以下の処理を行なう．
   (a) 前方から付属語，活用で一致する部分を探す．活用は l 文字目から探索する．
   (b) 最長一致した文字列を元の文字列から分割する．
   (c) a) の一致文字列が活用である場合は，一致文字列を前の非ひらがな文字列と結合する．
2. **後方修正**：$l = 1, 2$ のそれぞれの前方修正で残ったひらがな文字列を入力
   (a) ひらがな文字列の後方から付属語，活用で一致する部分を探す．
   (b) 最長一致した文字列が付属語であれば，残ったひらがな文字列から分割する．

基本的には，まず，最長一致を優先に付属語と活用の文字列を探索する．最長一致部が付属語であれば分割し，活用であれば前の文字列と結合することで，分割を修正する．仮に，付属語と活用の両方で最長一致した場合は，どちらの可能性も単語候補とする．前方修正では,「小さい」など，２文字目を活用する語が多数存在するため，２文字目からも探索して分割修正する．

図 1 の下線が，分割修正の対象となるひらがな文字列である．１番目の修正箇所である「くまで」の処理を説明する．前方修正では，$l = 1$ の時,「く」で活用する規則が存在するため「く / まで」，$l = 2$ では,「ま」で活用する規則が存在するため,「くま / で」となる．両方とも活用であるため，前方の非ひらがなの「遅」と結合し，ぞれぞれ「遅く / まで」「遅くま / で」に修正される．

後方修正は，前方修正 $l = 1$ の残りである「まで」と $l = 2$ の残りである「で」の両方を対象とする.「まで」は助詞に,「で」は接続詞として存在するため，後方の修正は行なわれない．

「っていた」も同様であり，前方修正では，$l = 1$ の時,「って」が付属語と活用に存在するため，付属語の場合「残 / って / いた」，活用の場合「残って / いた」となる．$l = 2$ では,「て」が活用に存在するため「残って / いた」となる．後方修正は，いずれも「いた」が対象となるが付属語に存在しないため，修正は行なわれない．

以上で，図 1 の分割修正の結果が得られる．

基本形変換　分割修正の結果には，活用により修正された語（以下，活用語と呼ぶ）とそうでない語（以下，非活用語と呼ぶ）が存在する．そのうち，活用語に関しては，活用部分を，適用できる全活用規則で基本形に変換し，それぞれを単語候補とする．これは，１つの活用の文字列には，複数の活用規則が適用できる場合があり，活用規則を一意に決めることができないためである．例えば,「遅く」のような「く」と活用する規則は６つある．それぞれ，１)「子音動詞カ行の基本形」から「遅く」，２)「子音動詞カ行促音便形の基本形」から「遅く」，３)「イ形容詞アウオ段の基本連用形」から「遅い」，４)「イ形容詞イ段の基本連用形」から「遅い」，５)「イ形容詞イ段特殊の基本連用形」から「遅い」，６)「ナ形容詞特殊のダ列特殊連用形」から「遅だ」,「助動詞そうだ型の基本連用形」から「遅し」，と基本形変換できる．

基本形変換の結果は，図 1 の２列目が品詞，３列目が活用規則で，４列目が活用名，５列目が基本形となっている．２列目は，活用語の場合，活用規則から品詞を決定し，非活用語は「unknown」とする．また非活用語は文字列をそのまま基本形とする．６列目が $l = 2$ であるものは，前方修正で２文字目が活用に一致したことを表す．そのため,「遅くま」では「ま」が活用となり,「ま」に適用できる「子音動詞マ行の未然形」により,「遅くむ」と基本形変換される．

最後に，付属語を除き，非活用語と全ての基本形に変換した活用語を単語候補としてデータベースに格納

する．単語候補データベースには，図１の２列から５列名までの情報を格納する．

次節より，単語候補のデータベースの中から単語を抽出するための単語判定処理について説明する．

3.2 単語判定

単語判定では，単語候補の単語らしさをいくつかの指標を組み合わせて評価することで，各候補が単語か否かを判定する．指標には，１）抽出頻度，２）文字種，３）文字数，４）抽出された活用の種類と数，を用いる．

１）からは，語としての一般性を評価できる．２），３）は，組み合わせることで単語らしい文字構成を表現できる．例えば，２から４文字の漢字のみで構成される，３から１０文字のカタカナのみで構成される，などである．４）は，活用語の尤もらしさを評価できる．理由は，単語候補が正しい活用語であれば，基本形，連用形，連体形など，様々な活用を伴い抽出されると考えられるためである．逆に１，２個の活用しか抽出されない候補は，偶然活用と一致したと考えられる．

単語判定法を具体的に説明する．まず単語候補データベースから単語候補の情報を取得する．単語は，原形，品詞，活用規則からなるため，単語候補データベースから，同じ原形，品詞，活用規則を持つレコードを対象に，頻度，活用の種類と数を取得する．具体例を表１を用いて説明する．表１から，「遅い，形容詞，イ形容詞アウオ段」の抽出頻度は合計の１７８，抽出された活用の種類が「タ系連用テ形，文語基本形，文語連体形，タ系連用タリ形，タ系条件形，タ形，基本形，基本連用形」で，活用の数が８となる．

次に，指標から作成した単語判定ルールで単語候補を判定する．例えば，「頻度が２０以上で，活用に基本形が含まれ，活用の種類が６種類以上ならば単語である」というルールがあるとする．上記の「遅い，形容詞，イ形容詞アウオ段」は，全ての条件を満たすため，単語と判定される．

実際には，ルールの条件の組み合わせが複雑にならないように，「頻度２０以上＝５」「活用が６種類以上で基本形を含む＝１０」など，細かいルールにスコアを付け，適合するルールのスコアの合計が閾値を超えた場合に単語と判定する．

表 1: 単語候補データベースの例

品詞	活用規則	活用	原形	頻度	文字列(参考)
…	…	…	…	…	…
形容詞	イ形形容詞アウオ段	タ系連用テ形	遅い	8	遅くて
形容詞	イ形形容詞アウオ段	文語基本形	遅い	4	遅し
形容詞	イ形形容詞アウオ段	文語連体形	遅い	2	遅き
形容詞	イ形形容詞アウオ段	タ系連用タリ形	遅い	2	遅かったり
形容詞	イ形形容詞アウオ段	タ系条件形	遅い	1	遅かったら
形容詞	イ形形容詞アウオ段	タ形	遅い	23	遅かった
形容詞	イ形形容詞アウオ段	基本形	遅い	62	遅い
形容詞	イ形形容詞アウオ段	基本連用形	遅い	76	遅くて
動詞	子音動詞カ行	タ形	遅く	2	遅いた
動詞	子音動詞カ行	基本形	遅く	76	遅くて
…	…	…	…	…	…

4 評価実験

本節では，単語抽出法の精度評価について説明する．具体的には，本手法で得られた単語と Juman[2] の辞書を比較し，辞書に含まれる語は正解とし，辞書に未登録の語 (**未知語**) はサンプリングして目視評価をする．全体の精度は，上記の割合と精度から推定する．

実験データには，ブログ記事１万件を用いた．単語判定ルールは，表２を用い，閾値は２とした．表２の ID1 から４は，活用語のためのルールであり，５から７は非活用語のためのルールである．非活用語の品詞は名詞と仮定した．目視評価による正解の基準と Juman 辞書との比較基準は，原形，品詞の大分類，活用規則の一致とした．目視評価では，複合語を不正解としたが，複数形態素からなると考えられる固有名詞（キングコング，姓名に分かれる人名など）は正解とした．これは，実際の応用上，固有名詞は，固有表現抽出などで複合語としてまとめた単位で処理することが多く，分割する必要はないと考えたためである．

結果を表 3 に示す．抽出された単語は非活用語が 17560，活用語が 1285 で，合計 18845 語であった．そのうち 8539 個が Juman 辞書に含まれた．未知語候補の 10252 から 300 語をランダムサンプリングして目視評価したところ 45 %が正解であった．したがって，単語抽出全体の精度は，$(10252\times0.45+8539)/18845 = 0.70$ と推定できる．

今回の実験で抽出された未知語の例を示す．活用語では「ケチる，バラす，ハマる」などが抽出された．非活用語は，「密会，新酒，海鮮」など一般的な語から，「心斎橋，小川流果」など固有名詞などがあった．実際に活用語と非活用語の両方の未知語を抽出できることが確認できた．

抽出された語は，Juman で形態素解析すると見かけ上解析が成功するものもある．例えば，「バラ（名詞）/す (サ変動詞：する)」「心 (名詞)/斎 (名詞)/橋 (名詞)」など，である．このように，形態素解析で発見できない未知語を検出できていることからも，本手法の有効性が確認できた．

表 2: 実験に用いた単語判定ルール

ID	単語判定ルール	スコア
1	カタカナで始りひらがなで終わる	1
2	漢字で始まりひらがなで終わる	1
3	アルファベットで始まりひらがなで終わる	1
4	活用が6種類以上で基本形を含む	1
5	カタカナのみで構成され3文字以上10文字以下	1
6	漢字のみで構成され2文字以上4文字以下	1
7	頻度が5以上	1

表 3: 単語抽出の精度と数

	精度	語数
Juman 辞書内の語	–	8539
未知語候補の推定精度	0.45(135/300)	10252
全体の推定精度	0.70	18845

5 考察

本手法の単語抽出ミスを分析したところ以下に分類できた．1 から 4 は提案法の設計上起こる典型的なミスであり全体の約 75 %を占める．特に 1 はその半数を占めるため，ミスの主要因と言える．

1. 複合名詞：宇宙人，未収録，関連性，etc...
2. 名詞と「的だ」を形容詞（ナ形容詞）(2)：対象的だ，etc...
3. 形容詞（ナ形容詞）の語幹を名詞：豪快，etc...
4. 名詞＋「だ」から形容詞（ナ形容詞）：受験生だ，etc...
5. カタカナの活用：ヤバイ，etc...
6. その他ミス：ゲゲゲ，デメル，本来，etc...

1,2 は文字種分割で分割できないことが原因である．このミスを解消するには，抽出された単語を用いての複合語チェックや，接尾辞，接頭辞の辞書を持つなどが考えられる．3 は，活用が十分に収集できなかったことが原因である．文書の規模を増やし，活用を収集すれば正しく抽出できると考えられる．4 は,「だ（判定詞）」がナ形容詞とほぼ同じ活用をするために起こる．ナ形容詞であるか，名詞＋「だ」であるのか判断するためには，名詞と考えられる部分が助詞と連接しているかチェックすることや，判定詞になくナ形容詞にある活用「に（ダ列基本連用形）」の有無に注目するなどが考えられる．

今回，1 から 3 は抽出ミスしたが，実際には 1 語と考えて問題ない場合もある．例えば，頻出単語を集計する場合,「宇宙人」を「宇宙」「人」に分けて集計するより，1 語としたそのままの方が適切であろう．また，文法的には,「一般的だ」を 1 つの形容詞として考えても問題ないと考えられる．

今後は，1）複数の文字種からなる単語抽出，2）品詞推定の強化，に取り組む．1）は，文字種で分割しているため「大リーグ」「大みそか」など，複数の文字種からなる単語が抽出できないためである．2）は，現状は品詞の再分類ができない，非活用語を全て名詞と推定しているが不適切な場合もある，などの問題があるためである．

6 関連研究

中渡瀬ら [6] は，文字列の出現頻度を利用して単語らしさを判定している．単純な出現頻度では，短い文字列が有利になるため，各文字列長別に，出現頻度分布の期待値と標準偏差が同じになるように正規化し，その正規化頻度を用いて,「語でない文字列より語である文字列のほうが正規化頻度は高い」という前提条件で，正規化頻度を比較しながら後の境界を探索することで，単語を抽出している．

永田ら [1] は，字種の組み合わせに基づく未知語タイプを設定し，各未知語タイプと品詞との連接確率および未知語の単語としてのもっともらしさを用いて未知語を抽出する手法を提案している．

池谷ら [5] は，漢字２，３文字の文字列を未知語候補とし，その分割パターンを設定し，その文字列が単語になる確率を推定することで未知語抽出を行っている．

浅原ら [4] は，複数の形態素解析結果の候補を用いて，文字単位に，1）文字，2）字種情報，3）各文字が属する形態素情報，4）形態素中における文字の位置情報を素性として，ＳＶＭによりチャンキングすることで，未知語文字列を認識している．

森ら [3] は，訓練コーパスからある品詞の前後の文字の分布を求めておき，一定以上出現する N グラムの前後の文字の分布を，各品詞の分布に重みを付けて，どの程度近似できるかという最適化問題として定式化することで，未知語抽出と品詞推定を同時に解決している．ある程度近似できるＮグラムは，重みの高い品詞の未知語と判断できる．

我々の提案法は，これまで考慮されていなかった，品詞，原形，活用規則も含めて抽出できる．また，形態素解析や訓練コーパスが必要なく，形態素解析システムや訓練コーパスへの依存がないため汎用的と言える．

7 おわりに

本稿では，文字種，活用形，付属語に基づいて原形，品詞，活用規則を含めて単語候補として，各候補の抽出頻度，文字数，文字種，活用の種類や数，に基づき単語判定することで，原形，品詞，活用規則を含めた単語抽出を実現する方法を提案した．評価実験では，単語抽出の精度は約 70 %であった．また，形態素解析では検出できない未知語を抽出できているとから本手法の有効性が確認できた．今後は，複数の文字種からなる単語の抽出に対応し，品詞推定を強化していきたい．

参考文献

[1] 永田昌明. 未知語の確率モデルと単語の出現頻度の期待値に基づくテキストからの語彙獲得. 情報処理学会論文誌, Vol. 40, No. 9, pp. 3373–3386, 1999.

[2] 黒橋禎夫, 河原大輔. 日本語形態素解析システム juman version 5.1 使用説明書. 京都大学大学院情報学研究科, 2005.

[3] 森信介, 長尾真. n グラム統計によるコーパスからの未知語抽出. 情報処理学会論文誌, Vol. 39, No. 7, pp. 2093–2100, 1998.

[4] 浅原正幸, 松本裕治. 形態素解析とチャンキングの組み合わせによる日本語テキスト中の未知語出現箇所同定. 情報処理学会研究報告 NL154-8, 2003.

[5] 池谷昌紀, 新納浩幸. 文字列が単語になる確率を用いた未知語抽出. 情報処理学会研究報告 NL135-7, pp. 49–54, 2000.

[6] 中渡瀬秀一, 木本春夫, 中川優. 統計的手法による辞書未登録語の獲得法. 電子情報通信学会論文誌, Vol. J81-D-2, No. 2, pp. 238–248, 1998.