語彙獲得のための過分割未知語の検出

村脇 有吾　　黒橋 禎夫
京都大学大学院情報学研究科
murawaki@nlp.kuee.kyoto-u.ac.jp kuro@i.kyoto-u.ac.jp

1　はじめに

日本語の形態素解析では、形態素候補の列挙に辞書を用いる手法が主流となっている [3, 1, 2]。この手法は既に高い精度を達成しているが、辞書にない形態素 (**未知語**) の解析を誤りやすいという問題がある。この未知語問題の一つの解決策として、テキストからの未知語の自動獲得による辞書の拡張が考えられる。

では、どのようにしてテキスト中の未知語を見つければよいだろうか。この検出タスクは、日本語は分かち書きされないため、自明ではない。特に、「うざい」、「きもい」等の未知語は、解析器が既知語の組に過分割してしまい、検出が難しい。そこで、一つの形態素が様々な表記を取りえるという日本語の性質を利用して、過分割された未知語を検出する手法を提案する。

2　未知語検出タスク

我々が提案したオンライン未知語獲得 [4] における検出タスクの位置づけを述べる。未知語獲得とは、未知語について、用例から辞書項目を帰納的に生成するタスクである。ここで、辞書項目は辞書に記述される抽象的な形態素であり、用例はそのテキスト中での具体的な出現である。

オンライン未知語獲得では、未知語は逐次的に入力されるテキストから獲得され、形態素解析へ直接フィードバックされる。これを実現するために、以下のようにサブタスクを設定しており、検出タスクは未知語獲得における最初のステップとなっている。

検出 各文の形態素解析結果から未知語の用例を検出。例えば、未知語「うざい」を含む文「うざいと思った。」に対して、「う」付近に未知語を含むことを見つける (正確な形態素境界の同定は不要)。

列挙 各未知語用例に対して、辞書項目の候補を列挙。列挙される候補は語幹と品詞からなる (例えば、語幹「うざ」と品詞「イ形容詞」)。

選択 各未知語用例に対して、最適な辞書項目の候補を選択。選択は、過去に検出された用例を蓄積しておき、それら複数用例の比較により行う。比較される用例が増え、曖昧性が十分に解消できた時点で獲得し、形態素解析器の辞書を更新する。

検出タスクの性質としては、検出された未知語用例のみが獲得対象となるため、一般に高い再現率が望まれる。適合率については、選択が複数用例の比較により行われるため、誤検出がただちに誤獲得には結びつくわけではない。しかし、誤獲得が形態素解析に副作用を起こすのは避けたい。また、ウェブテキストを対象とする場合、誤字や非規範的表記等、形態素解析器にとって未知の現象が現れる。こうした現象への対策は、本来頑健な形態素解析の実現により行うべきだが、未知語獲得側でも誤獲得がないように頑健な処理が望まれる。

3　ベースライン手法とその問題

形態素解析器は、入力文に対して、辞書引きと未知語処理により、出力すべき形態素の候補を列挙する。未知語処理により列挙される形態素候補を**未定義語**と呼ぶ。日本語の未知語処理では、字種に基づくヒューリスティクスが広く用いられている。例えば、カタカナの連続が一つの形態素候補とされる。これにより、未知語「ググる」を含む入力文「ググってみた。」に対して、未定義語「ググ」が形態素候補となり、これを含むパスが出力に選ばれる。従って、形態素解析結果中の未定義語に着目するのが、未知語用例の一番簡単な検出手法である。これをベースライン手法とする。

ベースライン手法では検出されない未知語用例が存在する。日本語の単純な音韻体系がわざわいし、未知語が既知語の組として過分割される場合がある。このような未知語には、主にカタカナの過分割とひらがなの過分割がある。カタカナの過分割の例としては、

- カースト ⇒ カー + スト
- アブラハム ⇒ 油 + ハム

などが、ひらがなの過分割の例としては、

- うざい ⇒ 卯 + 剤
- うざくて ⇒ 卯 + 座 + 区 + 手

- めんどかった ⇒ 面 + 度 + 買う
- かぐや姫 ⇒ 家具 + や + 姫

などがある。

では、どうすれば過分割された未知語用例を検出できるだろうか。まず、「カースト」や「アブラハム」等の外来語については、元言語において一語という知識が有効かもしれない。次に考えられるのは、語彙的な不整合である。例えば、「卯」と「剤」の連接は、明らかに人間の直観に反している。では、なぜこのような連接を形態素解析器が選択するのか。その主な原因は、解析器が候補選択に使う連接の手がかりが (一部の語彙化された付属語を除いて) 品詞の連接であり、語彙の連接が考慮されないことにある。例えば、「卯」と「剤」の連接は、解析器にとって、よくある名詞と名詞の連接でしかなく、この語彙的な不整合に気付かない。逆に言えば、語彙知識があれば、こうした未知語用例を検出できるかもしれない。

4 N-gram 言語モデルとその問題

語彙知識として、まずは単純な単語 (形態素) N-gram を考える。N-gram の構築には、形態素に分割されたテキストが必要となる。しかしタグ付きコーパス (例えば京都テキストコーパス) は小規模であり、データ・スパースネスが問題となる。一方、音声認識等で用いられる N-gram は、形態素解析結果から構築される。

では、過分割の未知語用例、すなわち解析誤りの検出にも、形態素解析結果が利用できるだろうか。一つの仮説として、テキストを大規模化すれば、解析誤りが無視できるようになるかを考えてみる。いくつかの形態素の調査から、この仮説は成り立ちそうにないことがわかる。例えば、「カースト」は、前後の文字列によらず「カー + スト」に分割される。また、「卯」の漢字表記「卯」とひらがな表記「う」の出現頻度をカウントすると、直観に反して「う」が「卯」の 50 倍以上となる。これは、「うざい」を「う」と「ざい」に過分割するといった解析誤りを N-gram がシステミックに拾っていることを示唆する。つまり、形態素解析器結果から構築した N-gram は、そのままでは未知語検出に利用できそうにない。

5 提案手法

5.1 アイデア

単語 N-gram に含まれるシステミックな誤りに対応するため、我々は表記ゆれの利用を提案する。日本語では、選択選好があるものの、一つの形態素が様々な表記を取りえる。例えば、「卯」は「卯」や「う」、「剤」は「剤」や「ざい」と表記される。従って、もし「うざい」の構成が「う + ざい」ならば、「卯剤」、「卯ざい」、「う剤」等の異表記が少しは出現すると期待される。しかし、こうした異表記は N-gram に出現しないので、この分割は怪しいと考えられる。

形態素の表記ゆれは、形態素解析器 JUMAN の辞書では、**代表表記**により吸収されている。例えば、「卯」と「う」は代表表記「卯/う」に集約される。そこで、表記ゆれの検出に代表表記を利用する。

いま、形態素列 w_{-1}, w_0, w_1 について、w_0 に着目し、w_1 との連接を調べるとする。このとき、出現頻度の比

$$L_{w_0,w_1} = C(w_0, r_1) \,/\, C(w_0\prime, r_1) \qquad (1)$$

が閾値以上であれば、w_0 を検出箇所として検出する。ただし、$w_0\prime$ は w_0 の異表記、r_1 は w_1 の代表表記とする。例えば、「うざい」の「う」に対して、

$$\begin{aligned} L_{\text{う, ざい}} &= \frac{C(\text{“う”}, \text{“剤/ざい”})}{C(\text{“卯”}, \text{“剤/ざい”})} \\ &= \frac{C(\text{“う”}, \text{“ざい”}) + C(\text{“う”}, \text{“剤”})}{C(\text{“卯”}, \text{“ざい”}) + C(\text{“卯”}, \text{“剤”})} \end{aligned}$$

を検査する。同様にして、後ろ向き bigram、つまり w_{-1}, w_0 の連接も検査する。

5.2 N-gram の構築

N-gram の構築の前に、検査対象表記を選定する。まず代表表記単位で対象を絞り込む。例えば、複数の表記がない形態素や、「コンピュータ」と「コンピューター」のようなカタカナ同士の表記ゆれを除外する。次に、対象代表表記 (例えば「卯/う」) について、検査対象表記から異表記へのマッピング (例えば「う ↦ 卯」) を作成する。具体的には、ひらがなや混ぜ書きを検査対象表記に、漢字やカタカナ表記を異表記にする。ただし、「いる」、「ある」、「ごく」等のひらがな表記の方が一般的な形態素は、検査対象から除外する。

N-gram の構築では、まずテキストを形態素解析する。解析結果の形態素列 w_{-1}, w_0, w_1 について、w_0 の代表表記 r_0 が検査対象ならば (「卯」と「う」の両者が該当)、前向きのカウント $C(w_0, r_1)$ と後ろ向きのカウント $C(r_{-1}, w_0)$ を更新する。ただし、未定義語を含む解析結果は利用しない。

5.3 N-gram の適用

検出タスクでは、解析結果の形態素列を前から順に走査し、未知語を含む箇所を検出する。形態素列中の各位置で、まず未定義語に着目するベースライン手法

を適用し、次に N-gram の検査を適用する。同じ用例の重複検出を避けるため、一度検出すると句読点等が出てくるまで検出をスキップする。

N-gram の検査は、形態素列 w_{-1}, w_0, w_1 について、w_0 が検査対象のとき (「う」のみが該当)、まず前向き bigram w_0, w_1、次に後ろ向き bigram w_{-1}, w_0 を検査する。異表記が複数ある場合は、そのすべてが条件を満たす場合に検出する。検出の閾値は予備実験に基づき経験的に設定した。

検出タスクでは、一般の N-gram 言語モデルと異なり、ゼロ頻度問題は重要ではない。そもそも訓練データにないということは、未知語である可能性が高い。まず、unigram のゼロ頻度問題は無視できる。未定義語はベースライン手法により検出できるので、対象を既知語に絞り込めるからである。bigram $C(w_0, r_1)$ については検討の余地があるが、今回はゼロ頻度であれば検出する。また、異表記の bigram $C(w_0\prime, r_1)$ のゼロ頻度は、まさに検出したい事例である。

6 実験

提案手法について、まず未知語用例の検出を評価し、次にオンライン未知語獲得で獲得される未知語の精度を評価した。

6.1 用例検出の実験設定

未知語用例の検出の定量評価には正解データが必要となる。一般に未知語の出現頻度は低いので、正解データはある程度の規模が必要だが、大規模テキスト全体に対する正解付与はコスト面から現実的でない。そこで、過分割の可能性がある短い形態素の連続のみを抽出し、それらに人手でタグ付けする。

まず、過分割候補の抽出を以下の手順で行う。

1. テキストの各文を形態素解析する。
2. 形態素列を前から順に走査し、以下のルールのいずれかにマッチする箇所を検出する。一度検出すると句読点等が出てくるまでスキップする。
   - ひらがな 1 文字 + ひらがな 1 文字
   - ひらがな 2 文字 + ひらがな 1 文字
   - ひらがな 1 文字 + ひらがな 2 文字
3. このルールだけでは付属語等の正しい組も大量に抽出されるので、それらをフィルタリングする。具体的には、あらかじめ京都テキストコーパスから同じルールにマッチする箇所を抽出しておき、それらと一致する箇所を除く。

検出された箇所に対して、人手で以下のタグを付与する。いずれにも該当しなければ何も付与しない。

U 未知語の語幹。「うざい」の「うざ」や、「あきんど」等。

E 解析誤りのうち、現在の形態素解析器の文法・語彙で正しく解析しうるもの。例えば、「何回かしか着ない」の既知語の連続「か + しか」は「か + し + か」と誤分割される。

O その他の形態素解析の誤り。旧かなづかい、非規範的表記、誤字、文抽出の失敗等。例えば「やつてみませうよ」や「やーめたっ」。

このうち、未知語用例の検出タスクにおいて検出が要求されるのは **U** だけで、**E** と **O** は、タスクの性質の理解のために、参考として与えている。これらは検出しても未知語獲得に貢献せず、むしろ誤獲得を生むおそれがあるが、提案手法では特に対策していない。

正解の判定条件は、検出箇所とタグ付け箇所が交差し、検出箇所の先頭位置がタグ付け箇所の先頭位置以降の場合とする。ただし、検出箇所の先頭位置が **U** の終了位置の場合を含む。例えば、「うざい」が「う」で検出された場合、正解となる。検出箇所にタグがなければ無視し、再現率のみを調べる。

テキストとして、ウェブコーパスから無作為に選ばれた 8517 文を用いた。過分割候補は 535 箇所抽出され、287 個のタグが付与された。N-gram の構築にはウェブコーパス約 1 億ページを用い、頻度 10 で足切りした。

6.2 用例検出の実験結果

表 1: 用例検出の実験結果

	U	E	O
ベースライン	57	1	69
提案手法	**116**	7	88
正解	159	18	110

検出実験の結果を表 1 に示す。提案手法はベースラインと比べて再現率を改善している。ベースラインからの新たな検出例を示す (下線は検出箇所)。

- かもめ ⇒ 鴨 + 目
- あずまんが ⇒ あ (感動詞) + 図 + 漫画
- すじこ巻き ⇒ する + 事故 + 巻く

提案手法でも検出されない未知語用例には、「めも」の「目 + も」、「しらす干」の「知る + す (接尾辞)」、「でかい」の「出る + かい (終助詞)」等がある。これらはいずれも局所的には自然なので、bigram では検出が難しい。また、前後の文脈によって検出が左右される未知語もある。例えば、「どれみちゃん」は「どれ + 味 + ちゃん」となって検出されるが、「どれみ！」だと「どれ + 見る」となって検出されない。

提案手法では **E** や **O** の検出も増加している。付与された **O** の割合からわかるように、ウェブテキストには形態素解析器にとって未知の現象が頻出する。これらが悪影響を及ぼすかは次の獲得実験で調べる。

6.3 未知語獲得の実験設定

検出をサブタスクとするオンライン未知語獲得において、獲得された未知語の精度を人手で評価する。対象テキストとして、TSUBAKI [5] の検索結果上位1,000 ページを用いる。クエリとして、「捕鯨問題」、「赤ちゃんポスト」および「JASRAC」を用いた。

正解の判定条件は、語幹と品詞の両者が正しい場合とする。ただし、構成的な語で分割に迷う場合は正解とする。また、名詞については、普通名詞や固有名詞といった品詞の細分類を区別しない。

獲得実験では文頭周辺を検出対象から外す。ウェブテキストの文頭はノイズの可能性があるからである。HTML からの正確な文抽出が難しいだけでなく、スニペットのように原文自体の文区切りがおかしい場合もある。

6.4 未知語獲得の実験結果

表 2: 未知語獲得の実験結果

クエリ	ベースライン	提案手法
捕鯨問題	99.0% (204/206)	97.2% (210/216)
赤ちゃんポスト	98.9% (91/92)	95.9% (94/98)
JASRAC	96.7% (532/550)	96.1% (566/589)

検出実験の結果を表 2 に示す。提案手法では、ベースラインに対して獲得未知語数が増えているが、全体の精度はやや下がっている。

ベースラインからの新たな獲得例には、「ぶろぐ」、「はてな」、「わんこ」、「ドラえもん」等の名詞のほか、動詞「ぐぐる」やイ形容詞「めんどくさい」がある。

誤り例をいくつか示す。名詞「とん」は、「とんかつ」の「とん」だけでなく、「とんでもねえ」の「とん」等と組み合わせて獲得されている。子音動詞タ行の動詞「はいつ」の獲得は、「…政府にはいち早い…」の「配置 + 早い」という誤解析や、「はいっ！」等の組み合わせにより行われている。また、イ形容詞「めんどい」は、検出に問題はないが、候補選択を誤り名詞「めんどい」として誤獲得された。

テキスト中の誤字は、検出されるものの、用例が少ないため最終的な獲得にいたっていない。

7 関連研究

未知語の自動獲得の研究のうち、検出に関しては、福島ら [6] が自明なカタカナ用言に対象を限定し、桑江ら [7] も同様の設定で実験を行っている。森ら [8] は、コーパスから総当たり的に形態素を抽出し、それらを形態素解析にかけて未知語か否かを判定している。

テキストの誤り検出に文字 N-gram[10] や単語 N-gram[9] を利用する研究がいくつかある。このうち、文字 N-gram は OCR 等の誤りを想定している。単語 N-gram は、かな漢字変換に起因する同音異義語の選択誤りを想定しており、あらかじめ用意された同音異義語セットを対象とする。

8 おわりに

本稿では、表記ゆれを利用して過分割された未知語用例を検出する手法を提案した。提案手法はベースラインと比べて再現率を改善したが、bigram では検出が難しい用例も存在する。また、今回はカタカナ過分割を対象外となっていたが、外来語の原語の情報や、文脈的な手がかり等も組み合わせて、過分割問題を解決したい。

参考文献

[1] Masayuki Asahara and Yuji Matsumoto. Extended models and tools for high-performance part-of-speech tagger. In *Procs. of COLING 2000*, pp. 21–27, 2000.

[2] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Procs. of EMNLP 2004*, pp. 230–237, 2004.

[3] Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. Improvements of Japanese morphological analyzer JUMAN. In *Procs. of The International Workshop on Sharable Natural Language Resources*, pp. 22–38, 1994.

[4] Yugo Murawaki and Sadao Kurohashi. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Procs. of EMNLP 2008*, pp. 429–437, 2008.

[5] Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. TSUBAKI: An open search engine infrastructure for developing new information access methodology. In *Procs. of IJCNLP-08*, pp. 189–196, 2008.

[6] 福島健一, 鍛治伸裕, 喜連川優. 機械学習を用いたカタカナ用言の獲得. 言語処理学会第 13 回年次大会 発表論文集, pp. 815–818, 2007.

[7] 桑江常則, 佐藤理史, 藤田篤. 後続ひらがな列に基づく語の活用型推定. 情報処理学会研究報告, Vol. 2008-NL-186, No. 186, pp. 7–12, 2008.

[8] 森信介, 長尾眞. 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.

[9] 三品拓也, 貞光九月, 山本幹雄. 確率的 LSA を用いた日本語同音異義語誤りの検出・訂正. 情報処理学会論文誌, Vol. 45, No. 9, pp. 2168–2176, 2004.

[10] 荒木哲郎, 池原悟, 佐藤政伸, 榮代正男. マルコフ連鎖モデルを用いた日本語文の置換型, 挿入型及び脱落型誤りの検出・訂正法の改善. 電子情報通信学会論文誌, Vol. J85-D-II, No. 1, pp. 66–78, 2008.