形態論的制約を用いた未知語の自動獲得

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

日本語は分かち書きされないため、複数の用例の調べなければ、適切に未知語を処理できない。そこで、我々は、テキストから逐次的に語彙を獲得し、その場で形態素辞書を自動更新する枠組みと、その具体的な実装手法を提案する。

語彙獲得の手がかりとしては、自立語に後接する付属語列に関する形態論的制約を用いる。複数の用例における付属語列の振る舞いを調べることにより、未知語が確実に獲得できることを示す。

2　背景

人手で整備された解析用辞書には、高頻度の形態素のみが登録され、低頻度の形態素は、解析上問題でありながら、これまで未知語として放置されてきた。未知語の多くは、低頻度だが数の多い、いわゆるロングテールに属し、人手での対応は現実的でない。そのため、計算機による自動処理が必要となる。

未知語処理は、形態素境界の同定と、品詞の割り当ての二つのタスクからなる。英語と比較すると、日本語は分かち書きされないため、未知語の品詞だけでなく、形態素境界すら明らかでない。また、英語の POS タグがテキスト中での用法を表すのに対し、日本語の品詞は、原則的に一つの形態素に一つ割り当てられる。そのため、例えば以下のような困難がある。

- 「倚りかかって」という用例からは、「倚-る」という動詞か、「倚りかか」という名詞か、あるいは「倚りかか-る」という動詞か判断できない。
- 「ググ-る」のように基本形が「る」でおわる動詞は、母音動詞か子音動詞ラ行か分からない。識別には、別の活用形、例えば、未然形「ググ-ら」(ググらない、ググらず) を調べる必要がある。
- 名詞についても、サ変名詞「完熟」は、「完熟の」、「完熟を」といった用例だけでは、普通名詞から区別できない。「する」、「できる」といった語が後続するか調べる必要がある。

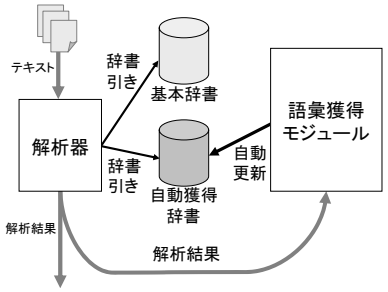

図 1: 語彙獲得システムの構成

- 反対に、普通名詞「熟度」をサ変名詞から識別するには、サ変名詞らしくないことに気付く必要がある。

未知語の処理方法としては、解析器内での未知語処理と、解析器外での自動獲得の 2 通りが考えられる。前者の枠組みでは、複数の用例を調べることは難しく、未知語を適切に処理できない。一方、自動獲得の従来手法は、学習と獲得の両段階で、コーパス全体から計算された統計量を用いている。この手法には、低頻度語の獲得精度に問題がある。また、ウェブコーパスのように、巨大で、かつ部分的に更新・拡張され得る場合にも適用が難しい。

3　基本的なアイデア

我々は、比較的少量の用例から逐次獲得する手法を採用する。語彙獲得モジュールは、図 1 のように、解析器の出力を継続的を受け取り、怪しい解析結果から新たな形態素を探す。ある時点で新たな形態素を獲得すると、人手を介さず直接辞書を更新し、以降の解析に反映させる。

語彙獲得には、手がかりとして形態論的制約を利用する。自立語にはいくつかの付属語が後続し、その連接には強い制約がある。従って、未知語について複数の用例を調べると、特定の文字列に様々な文字列が後続する現象が観測される。また、そうした後続文字列が、各品詞が持つ制約を満たすかを調べることで、未知語に適切な品詞が割り当てられる。

表 1: 品詞とサフィックス

形態素	品詞	語幹	語尾	活用形	サフィックスの例
走る	子音動詞ラ行	走	ら	未然形	らず, らないで
見る	母音動詞	見	ϕ	未然形	ず, ないで
希望	サ変名詞	希望	ϕ	を	を, をも
			ϕ	する	する, したら

もっとも、獲得開始時には、そもそも文字列中のどの範囲が未知語か明らかでない。そこで、語幹の文字列の一部をピボットとし、ピボットを手がかりに同じ未知語の複数の用例を集める。集まった複数の用例を調べて形態素の前方境界と後方境界を確定させる。形態素境界が決定すると、再び複数の用例を調べ、複数の品詞候補から最適な品詞を選択する。

従来手法では、入力ごとに最適な候補の推定を行ってきた。一方、逐次獲得では、ある時点で得られた入力が推定に十分な量とは限らないし、後から新たな入力が得られる可能性もあるため、確実と判断した時点で初めて形態素を獲得する。

4 用語

品詞体系は形態素解析器 JUMAN に従う。分類には品詞、品詞細分類および活用型があるが、その適当な組み合わせを便宜的に品詞と呼ぶ。今回は、動詞については個々の活用型、名詞は普通名詞とサ変名詞、形容詞はイ形容詞とナ形容詞を獲得対象の品詞とする。

形態素を構成する文字列のうち、不変部分を語幹と呼ぶ。活用する形態素は、語幹に語尾が付き、基本形、基本連用形などの複数の活用形のうちいずれかをとる。名詞のような非活用語についても、直後に接続する付属語の原形を擬似的に活用形と呼ぶ。また、語尾と 0 個以上の付属語を連結した文字列をサフィックスと呼ぶ。表 1 に例を示す。

5 サフィックスの獲得

語彙の自動獲得のために、あらかじめサフィックスに関する知識をコーパスから獲得する。

ウェブコーパス約 1 億ページの解析結果から、簡単なルールにより怪しい解析結果を除き、品詞ごとに、サフィックスと活用形の対応を獲得する。ただし、サフィックスを構成する付属語には、機能表現化した動詞なども含まれるため、大規模コーパスでもサフィックスの異なり数が収束しない。そこで、サフィックスを最長 5 文字でマージする。実験では、異なり数約 50 万、総出現回数約 33 億のサフィックスを得た。

あるサフィックスが、品詞 a に現れ、品詞 b に現れない時、a から b に弁別的と言う。また、品詞 a における、a から b に弁別的なサフィックスの出現確率の和を $d_{a,b}$ とする。ただし、獲得漏れや解析ミスの可能性を考慮して、弁別的なサフィックスには、高頻度 (実験では 10^{-5} 以上) のもののみを用いる。

弁別的なサフィックスは品詞の識別に利用できる。また、$d_{a,b}$ は品詞の識別しやすさを示す。例えば、$d_{\text{イ形容詞, 子音動詞ラ行}}$ はほぼ 1 なので、イ形容詞と子音動詞ラ行は明らかに異なる。一方、$d_{\text{普通名詞, サ変動詞}}$ は 0.001 程度であり、出現したサフィックスでは、普通名詞をサ変名詞から区別できない。反対に、$d_{\text{サ変動詞, 普通名詞}}$ は約 0.2 なので、ある程度の数の用例を調べれば、サ変名詞を普通名詞から識別できると期待される。

6 自動獲得のアルゴリズム

6.1 怪しい解析結果の検出

解析器が返す解析結果から怪しい部分を検出し、新たな語彙の獲得対象とする。

今回は、怪しい解析結果として、JUMAN が未定義語とする「形態素」を用いる。もちろん、未知語でありながら、既知語の組み合わせとして誤認識される場合がある。例えば、「うざい」は「う (卯/雨/鵜)」と「ざい (剤/在/罪/財)」に分解される。こうした未知語の検出は今後の課題とする。

解析結果から未定義語が検出されると、構文解析器 KNP により作った文節から、未定義語を含む文節とその前後の文節を抽出する。以降これを用例と呼ぶ。

6.2 ピボットの選択

同じ未知語の複数の用例を簡便に集約するために、各用例からピボットとなる文字列を選ぶ。ピボットは語幹の一部であり、同じ未知語の用例からは同じピボットが選ばれるようにする。ただし、異なる未知語から同じピボットが選ばれる可能性もある。

今回は、未定義語とされた文字列をピボットとする。ただし、連続する漢字の一つが未定義語となった場合、

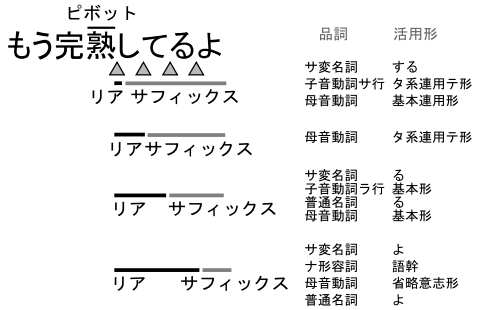

図 2: サフィックスのマッチングによる候補の列挙

サフィックスとならない一番右の漢字をピボットとする。例えば、用例「もう完熟してるよ」の場合、「完」が未定義語となるが、ピボットとしては「熟」を選ぶ。

6.3 後方境界候補の列挙

各用例について、図 2 のように、サフィックスのマッチングにより、後方境界の候補を列挙する。各後方境界候補について、品詞と活用形の組の候補も列挙する。

マッチングは、開始位置をピボットの直後から 1 文字ずつずらしながら行う。マッチの条件は、原則的に、サフィックスの終了位置が KNP の返す文節境界と一致する場合とする。KNP が未知語の周辺で文節境界を誤る可能性もあるが、付属語が後続している場合には、その可能性は低いと期待される。

6.4 前方境界の決定

用例を管理するために、図 3 のように、ピボットごとに、ピボットよりも前の文字列により、前向きのトライを作る。用例を格納する終端ノードは、文頭または文節頭 (BOS)、あるいは句読点など境界を表す文字 (SEP) のいずれかである。用例格納時に、BOS と SEP に十分な用例が入っていたら (実験では 3 以上)、その位置を前方境界として次のステップに進む。より少数の用例からの獲得が求められる場合のために、何らかの手法で形態素境界を判定し、BOS と SEP だけでなく、より深いノードに格納された用例も利用するという拡張を検討している。

6.5 後方境界の決定

前方境界を共有する複数の用例を調べることによって、不適切な後方境界の候補を取り除く。

前方境界を共有する用例の全体集合を R、そのうちリア p が候補として挙げられた用例の集合を R_p とする。リア p,q について、q が p を先頭から部分文字列として含む時、用例の包含関係を調べる。R_q が R_p に包含されるなら、q を候補から除外する。例えば、「完熟」のリア「」とリア「させ」について、用例「完熟の」、「完熟させた」、「完熟する」を調べると、リア「」はすべてマッチするが、リア「させ」は一つにしかマッチしない。ただし、サフィックスのマッチングのミスの可能性を考慮し、包含の条件は以下とする。

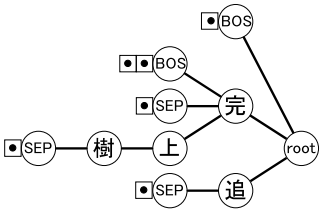

図 3: ピボット「熟」のトライ

$$\frac{|R_q - R_p|}{|R_p \cap R_q|} < \alpha$$

実験では $\alpha = 0.05$ とする。

6.6 品詞の識別

各語幹候補について、複数の品詞候補から最適なものを選び、それが適切であれば採択する。

語幹を共有する用例の全体集合を T とし、そのうち品詞 a が候補として挙げられた用例の集合を T_a とする。まず、

$$|T_a|/|T| < \beta$$

となる品詞 a を品詞候補から除外する。実験では $\beta = 0.9$ とする。特に動詞について、この時点で候補が一つに絞られる場合がある。このとき、用例中に出現する活用形の異なり数が 3 以上の場合に採択する。

残った品詞候補について、品詞のペアごとに確信度を計算する。品詞 a の品詞 b に対する確信度 $confidence_{a,b}$ は次の通り。

$$confidence_{a,b} = \sum_{suffix \in T_a} ss_{a,b}(suffix)$$

$$ss_{a,b}(suffix) = \begin{cases} p(suffix|a) & \text{弁別的,} \\ 0 & \text{oterwise.} \end{cases}$$

これは、a から b に弁別的なサフィックスがあれば、品詞 a の可能性が高いという考えに基づく。

ここで、品詞 b について、$confidence_{a,b} > \gamma$ を満たす品詞 a が存在する時、b を候補から除外する。実験では $\gamma = 0.01$ とする。$confidence_{a,b} \leq \gamma$ の時、弁別的なサフィックスの総出現頻度を用いて、確信度を以下のように変更する。

$$confidence_{a,b} = 1 - (1 - d_{b,a})^{|T_a|}$$

これにより、例えば、普通名詞からサ変名詞に弁別的なサフィックスがない時、サ変名詞から普通名詞に弁

別的なサフィックスが十分出現しなければ、普通名詞と推測する。

確信度を元に、以下を満たす品詞 a を最適な候補とする。

$$\operatorname*{argmax}_{a} \min_{b} confidence_{a,b}$$

最適な品詞候補が以下の条件を満たす時、採択する。

- 用例中に出現する活用形の異なり数が 3 以上
- すべての b について $confidence_{a,b} > \delta$

δ を大きくすると、採択に必要な用例数が増える。実験では $\delta = 0.9$ とする。

6.7 辞書の更新

採択時には、新たな形態素を辞書に追加して更新し、獲得に用いた用例をトライから削除する。また、トライ中のより深いノードも検査し、獲得した形態素で解釈できる用例を削除する。

特定の複合表現が連続して用例となった場合、本来よりも長い語幹で獲得される可能性がある。例えば、「もう完熟した」、「もう完熟で」といった用例が続くと、「もう完熟」が獲得される。そこで、獲得済みの形態素が、新たに獲得した形態素によって分割できる場合、辞書から削除する。

7 実験

人手での登録から漏れた頻出語の整備を目的として、ウェブコーパス約 1 億ページを対象に実験を行った。同時に、提案手法の小規模コーパスへの適用可能性を示すため、獲得に要した用例数を調べた。

大規模計算を行うために、ピボットごとに処理を並列化した。あらかじめ怪しい解析結果からピボットを抽出した上で、コーパス中に 100 回以上出現したピボット約 19 万個について逐次獲得を行い、最後に結果をマージした。

結果を表 2 に示す。獲得例にあるように、新聞には出現しない訓読みや新語、俗語等が獲得できた。また、ウェブコーパスの性質上、低頻度の獲得例には「手こづる」や「諦あきらめる」のような誤字が含まれる。

無作為に選んだ 100 個の形態素を調べたろころ、精度 77% (77/100) であった。ただし、誤りのうち、分割可能な複合語 9 個、名詞とナ形容詞の識別ミス 5 個を除くと、残りは 9 個となる。

誤りには解析器や語彙獲得モジュールにとって未知の文法現象が多い。特に、獲得対象外の品詞である感動詞と副詞を誤認識した。ウェブというコーパスの性格に由来する誤りには、「ウザイ」などの語尾までカタカナ表記の用言や、カタカナ表記の指示詞「コチラ」などがある。また、極端に短い語幹「ぷ-る」、「ぱ-る」なども獲得されてしまった。

表 2: 実験結果

品詞	獲得数	用例数	獲得例
母音動詞	1,854	18	焼る
子音動詞カ行	355	9	ムカつく
子音動詞ラ行	893	6	ググる
ザ変動詞	13	4	殉ずる
イ形容詞	814	18	甘酸っぱい
ナ形容詞	936	12	イマイチだ
普通名詞	37,935	52	防具
サ変名詞	6,854	9	クリック
合計	50,652		

品詞としては、他に子音動詞サ行、子音動詞タ行、子音動詞バ行、子音動詞マ行、および子音動詞ワ行も獲得している。
用例数は、獲得時点での用例数の中央値。

獲得に要した用例数は、普通名詞を除いて少ない。普通名詞、サ変名詞およびナ形容詞の識別は難しいが、実際には品詞を誤ってもそれ程問題ではない。今回は獲得に要する用例を多めに設定したが、少数の用例から一旦獲得し、獲得後も用例を監視して、適宜品詞を修正する方が良いかもしれない。

8 関連研究

獲得型の未知語処理では、学習と獲得の両段階で、コーパス全体から計算された統計量が用いられている。森ら [2] は、前後に隣接する文字列の n-gram を利用している。福島ら [1] は、対象をカタカナ用言に限定し、後続する文字列の n-gram を利用している。

9 おわりに

本研究では、テキストから未知語を逐次的に自動獲得する仕組みを提案した。今後は、ノイズに対する頑健性を改善するとともに、より詳細な実験により提案手法の有効性を検証したい。

今回利用した形態論レベルの制約では区別が付かないため、固有名詞を普通名詞扱いとしたが、今後、適切に品詞の再割り当てを行いたい。

参考文献

[1] 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.

[2] 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.