KyTea を利用した日本語 all-words WSD

新納 浩幸 [1)]　森 信介 [2)]　古宮 嘉那子 [3)]　佐々木 稔 [4)]

茨城大学 工学部 情報工学科 [1,3,4)]
京都大学 学術情報メディアセンター [2)]
hiroyuki.shinnou.0828@vc.ibaraki.ac.jp, mori@ar.media.kyoto-u.ac.jp,
kanako.komiya.nlp@vc.ibaraki.ac.jp, minoru.sasaki.01@vc.ibaraki.ac.jp

1 はじめに

本稿では我々が作成した日本語 all-words WSD のシステムを紹介する．

語義曖昧性解消は意味解析の根幹の処理でありながら，そのシステムが現実のアプリケーションで広く利用されているとは言いがたい．その大きな原因は，通常の語義曖昧性解消システムが対象単語を限定しているからである．そのような背景から，我々は対象単語を限定しない語義曖昧性解消である all-words WSD の研究に取り組んでいる．

all-words WSD は品詞 tagger の問題と見なせる．そのため訓練データを用意すれば京都テキスト解析ツールキット KyTea[1] を利用して，all-words WSD が簡易に実現できる．しかも KyTea による all-words WSD の入力は平文でよく，出力は通常の形態素解析の結果の上に語義が付与されたものとなる．このため開発したシステムは様々なタスクに利用されることが期待できる．

2 品詞 tagger の問題としての all-words WSD

all-words WSD は品詞 tagger の問題として扱える [1]．例えば以下の例文を考えてみる．

/ 国民 / の / 声 / を / 聞く /

文中の多義語は「国民」「声」「聞く」であり，岩波辞書の中分類では，それぞれ「17228-0-0-0, 17228-0-0-1」「27346-0-0-0 ～ 27346-0-0-5」「10487-0-0-0 ～ 10487-0-0-3」の語義がある．これらの語義の正しい組み合わせを求めるのが all-words WSD であるが，語義を単語の品詞とみた場合，これは品詞 tagger と同じ形の問題であることがわかる．つまり all-words WSD は図 1 のような語義をノードとする有向グラフを作成し，開始ノード S から終了ノード E への最適パスを求める問題と見なせる．

3 KyTea を利用した all-words WSD

KyTea はテキスト解析のツールキットである．単語分割されたテキストを訓練データとして，単語分割のモデルを学習する．そして学習されたモデルを利用して，プレーンなテキストに対して単語分割を行える．訓練データ中の単語に品詞を付与しておけば，単語分割と同時に品詞の付与も行える．つまり品詞の部分を語義に置き換えることで，all-words WSD のシステムを簡易に構築できる．構築できたシステムは単語分割も同時に行え，しかも訓練データの単語に品詞を付与しておけば，品詞の付与も行える．つまり KyTea によって構築できる all-words WSD のシステムはプレーンなテキストを入力として，出力としては通常の形態素解析結果の上に語義を付与したものとなる．

問題は訓練データであるが，ここでは，東京工業大学の奥村研で公開されている「語義タグ付きコーパス」を利用する．このコーパスは国立国語研究所の「現代日本語書き言葉均衡コーパス」(BCCWJ) のコアデータである 6 領域の計 1,980 文書中の全ての多義語に岩波辞書の語義を付与したものである．語義が付与された多義語の種類は 4,916 語であり，その総数は 114,696 語である．

上記の訓練データを用いて KyTea によって構築されたモデルを `wsd0.mdl` と名付ける．

`wsd0.mdl` を用いたシステムの実行例を図 2 に示す．上記の訓練データでは，語義の曖昧性のない単語には語義が付与されていない．このため `wsd0.mdl` を用

[1] `http://www.phontron.com/kytea/index-ja.html`

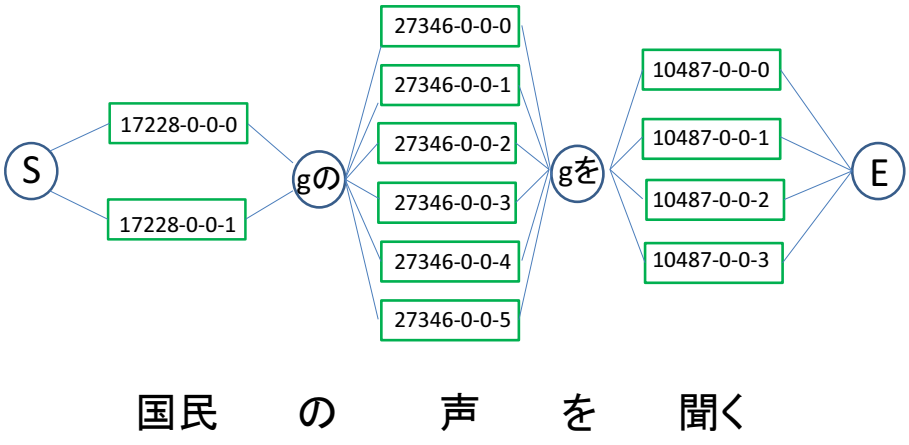

図 1: 語義をノードとした有向グラフ

```
> cat sample.txt
学問に基づいた本物の技術を創出できるのはやはり大学だと実感し始めていたからである.

> kytea -model wsd0.mdl < sample.txt
学問/名詞-普通名詞-サ変可能/7705-0-0-2 に/助詞-格助詞/NO 基づい/動詞-一般/NO た/助動詞/NO
本物/名詞-普通 名詞-一般/NO の/助詞-格助詞/NO 技術/名詞-普通名詞-一般/10703-0-0-2 を/助詞
-格助詞/NO 創出/名詞-普通名詞- サ変可能/NO できる/動詞-非自立可能/NO の/助詞-準体助詞/NO
は/助詞-係助詞/NO やはり/副詞/52112-0-0-0 大学/名詞-普通名詞-一般/30595-0-0-1 だ/助動詞/
NO と/助詞-格助詞/NO 実感/名詞-普通名詞-サ変可能/NO し/動詞-非自立可能/NO 始め/動詞-非自立
可能/NO て/助詞-接続助詞/NO い/動詞-非自立可能/NO た/助動詞/NO から/助詞-接続助詞/NO で/助
動詞/NO ある/動詞-非自立可能/NO . /補助記号-句点/NO
```

図 2: wsd0.mdl による all-word WSD の実行例

いたシステムでは多義語に対してだけ語義が付与される.

4 語義を概念とした all-words WSD

前章では語義を品詞と見なして KyTea を利用することで all-words WSD を実現した. しかし語義と品詞は種類数が大きく異なる. このため語義を正しく識別するためにはかなり大規模な訓練データ（語義タグ付きコーパス）が必要となる. 例えば「声」の語義は「27346-0-0-0 〜 27346-0-0-5」の 6 つだが, これらの語義は訓練データ中の単語「声」の語義だけにしか現れないため,「声」の語義を正しく識別するには単語「声」を数多く含んだ訓練データが必要になる. 一方「声」の品詞は「名詞-普通名詞-一般」の 1 つだけであり曖昧性さえなく, しかも「名詞-普通名詞-一般」という品詞は単語「声」を含まなくても訓練データ中に多数出現している. そのため「声」の品詞を正しく識別するには単語「声」を含まない訓練データであっても可能である.

この問題の対処として語義を概念に一般化する. 論文 [4] では岩波辞書の語義を all-words WSD が可能になる粒度に一般化した概念辞書を作成し, 辞書の語義を付与する all-words WSD を実現している. ここではそこで作成した概念辞書を利用する. 具体的に

```
> kytea -model wsd1.mdl < sample.txt
学問/名詞-普通名詞-サ変可能/276 に/助詞-格助詞/347 基づい/動詞-一般/331.0 た/助動詞/347 本物/名詞-普通名詞-一般/22 の/助詞-格助詞/59 技術/名詞-普通名詞-一般/448 を/助詞-格助詞/307 創出/名詞-普通名詞-サ変可能/158 できる/動詞-非自立可能/394.1 の/助詞-準体助詞/59 は/助詞-係助詞/118 やはり/副詞/258.0 大学/名詞-普通名詞-一般/148 だ/助動詞/352 と/助詞-格助詞/347 実感/名詞-普通名詞-サ変可能/448 し/動詞-非自立可能/307 始め/動詞-非自立可能/324 て/助詞-接続助詞/347 い/動詞-非自立可能/177 た/助動詞/347 から/助詞-接続助詞/118 で/助動詞/118 ある/動詞-非自立可能/286 . /補助記号-句点/318

> kytea -model wsd1.mdl < sample.txt > out.tmp
> python c2s.py out.tmp    # <-- 概念から語義への変換
学問/名詞-普通名詞-サ変可能/7705-0-0-2 に/助詞-格助詞/NO 基づい/動詞-一般/NO た/助動詞/NO 本物/名詞-普通名詞-一般/48389-0-0-0 の/助詞-格助詞/NO 技術/名詞-普通名詞-一般/10703-0-0-2 を/助詞-格助詞/NO 創出/名詞- 普通名詞-サ変可能/29448-0-0-0 できる/動詞-非自立可能/35052-0-0-1 の/助詞-準体助詞/NO は/助詞-係助詞/NO やはり/副詞/52112-0-0-0 大学/名詞-普通名詞-一般/30595-0-0-1 だ/助動詞/NO と/助詞-格助詞/NO 実感/名詞-普通名詞-サ変可能/21538-0-0-0 し/動詞-非自立可能/20314-0-0-2 始め/動詞-非自立可能/41135-0-0-2 て/助詞-接続助詞/NO い/動詞-非自立可能/1609-0-0-1 た/助動詞/NO から/助詞-接続助詞/NO で/助動詞/NO ある/動詞-非自立可能/1381-0-0-0 . /補助記号-句点/NO
```

図 3: wsd1.mdl による all-word WSD の実行例

はその概念辞書を利用して，訓練データ中の語義を概念に変換し，先と同様 KyTea によってモデルを学習する．ここではこのモデルを `wsd1.mdl` と名付ける．`wsd1.mdl` を利用すれば KyTea により語義を概念とした all-words WSD が実現できる．

また利用した概念辞書は，概念 c と単語 w が与えられたとき，単語 w の語義 s が一意に定まるように構築されている [4]．このため `wsd1.mdl` を用いた KyTea の出力を，通常の all-words WSD の出力に変換することができる．

`wsd1.mdl` を用いたシステムの実行例を図 3 に示す．`wsd1.mdl` を用いたシステムでは，後処理により概念と単語から語義を求めているので，名詞と動詞の全ての単語に語義が付与される[2]．

5 システムの評価

まず `wsd0.mdl` を用いたシステムの速度を評価する．CPU Core i5-661，メモリ 12GB，OS Ubuntu 15.04 64bit の上で新聞 1 年間分のプレーンテキスト (約 120MB) の解析に約 313 秒を要した．この程度の速度でも研究目的であれば十分に利用可能だと思われる．

次に精度について述べる．all-words WSD というタスクの関係上，精度の評価は難しい[3]．ここではごく小規模な実験として Senseval-2 における日本語辞書タスク [5] のデータのうち多義語「意味」のタグ付き例文 100 文（評価データ）の語義識別を行った．「意味」の語義は 2843-0-0-1, 2843-0-0-2, 2843-0-0-3 の 3 つであり，それぞれの語義の評価データの数は 36, 37, 27 であった．つまり MFS を用いても正解率は 0.37 である．一方 `wsd0.mdl` を用いたシステムの正解率は 0.48 であった．つまり MFS 以上の正解率は出すと考えられる[4]．また `wsd1.mdl` を用いたシステムの正解率は 0.59 であった． `wsd1.mdl` を用いたシステムの方が `wsd0.mdl` を用いたシステムよりも正解率は高い

[2] ただし動詞の場合，活用形が異なると別の単語と見なされるので，原形ではない動詞に対しては語義が付与されない場合もある．

[3] SemEval-2 の日本語辞書タスクのデータ [3] が使えそうだが，このデータは訓練データ中に含まれているため評価には利用できない．

[4] 訓練データを MFS で識別した場合，精度（マイクロ平均）は 0.777 であり，`wsd0.mdl` を用いたシステムであっても，その程度の精度は確保できると予想する．

が，後述するようにこれは KyTea の後処理の影響である． `wsd0.mdl` を用いたシステムでは「意味」の語義を付与できないものが 15 個存在した． `wsd1.mdl` では概念から語義への変換によって語義を付与できないことはない．それら違いが `wsd0.mdl` と `wsd1.mdl` の正解率の差を生じさせていた．それ以外の部分に関しては 1 例だけ識別結果が異なっただけであった．

最後にカバー率について述べる． `wsd0.mdl` を用いたシステムでは訓練データに出現しない多義語 w に対しては対応できず w の語義が付与できない．また訓練データに多義語 w が出現していても w の語義の中で訓練データに出現しない語義 s が存在する場合，テストデータにおいて w の語義を s と判定することはない．このため訓練データが岩波辞書における多義語や語義をどの程度カバーしているかは重要な問題である．岩波辞書の登録単語数は 56,257 単語である．そのうち多義語になっているものは 13,190 単語である．そして全多義語の語義の総数は 36,354 語義である．一方，ここで用いた訓練データ内の多義語は 4,916 単語，それら多義語の語義の総数は 7,219 語義である．つまり現システムのカバー率は約 2 割である．このカバー率を上げることが今後の課題と言える．

6 考察

前述したようにシステムのカバー率を上げることが今後の課題である．これは訓練データにあたる語義タグ付きコーパスを拡充させることを意味する．このためには all-words WSD のシステムが必要である[5]．語義タグ付きコーパスを作成するためには教師付き学習手法や知識ベース手法を併用しながら all-words WSD を実現するつもりである．注意としてこのように構築した all-words WSD システムがあれば，もはや KyTea を使った all-words WSD に意味がない，というのは間違いである．all-words WSD システムでは，精度の問題よりも頑健性，利用のしやすさ，また拡張の容易性などが重要だと考えている．

また細かな問題点を 2 点あげる．1 つは，評価の際の実験でも確認されたが，語義を付与しなければならない単語に，語義が付与されないことが起こる問題である．現状，訓練データに未出現の多義語や，多義語ではない単語には語義が付与されない．これは辞書を準備しておけば容易に解決できる問題である．辞書を準備する場合 default の語義を設定するはずだが，その際に知識ベース手法（例えば Lesk 手法 [2] など）を利用して default の語義を設定しておけば精度がかなり改善できる．default の語義を考慮した辞書作成が直近の課題である．もう 1 つは動詞などの用言の語義が，活用形毎に与えられる問題である．このため例えば訓練データ中に「音楽を聞く.」が存在しても，テストデータ中の「音楽を聞いて，…」の「聞く」の語義を正しく識別できない可能性がある．この点は問題点ではないとも見なせるので，現在，検討中である．

[5] KyTea で使うのは最終的に得られた語義タグ付きコーパスである．

7 おわりに

本稿では我々が作成した日本語の all-words WSD のシステムを紹介した．語義タグ付きコーパスを訓練データとし，KyTea を利用してテキスト解析用のモデルを構築する．KyTea からこのモデルを利用することで，プレーンなテキストを入力として，通常の形態素解析結果の上に語義が付与された出力を得ることができる．現状，Kytea を利用して all-words WSD が実現できることを示しただけに留まっている．ただし default の語義を記した辞書を用意するだけで，ここで作成した all-words WSD システムは多くの自然言語アプリケーションに使えるものであると考えている．KyTea の訓練データにあたる語義タグ付きコーパスの拡充が今後の課題である．

参考文献

[1] Jun Hatori, Yusuke Miyao, and Jun'ichi Tsujii. Word Sense Disambiguation for All Words using Tree-Structured Conditional Random Fields. In *COLING-2008*, pp. 43–46, 2008.

[2] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *the 5th annual international conference on Systems documentation*, pp. 24–26, 1986.

[3] Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. SemEval-2010 Task: Japanese WSD. In *The 5th International Workshop on Semantic Evaluation*, pp. 69–74, 2010.

[4] 新納浩幸, 古宮嘉那子, 佐々木稔. all-words wsd のための概念辞書の自動作成. 情報処理学会自然言語処理研究会, pp. NL–224–13, 2015.

[5] 白井清昭. Senseval-2 日本語辞書タスク. 自然言語処理, Vol. 10, No. 3, pp. 3–24, 2003.