Vol.2016-NL-227 No.2
2016/7/29

点推定による日本語 all-words WSD システム KyWSD

新納 浩幸[1,a)]　古宮 嘉那子[1,b)]　佐々木 稔[1,c)]　森 信介[2,d)]

概要：ここでは我々が開発・公開している日本語の all-words WSD システム KyWSD を紹介する．KyWSD は点推定を基本にした単語分割学習システム KyTea を利用したものであり，その拡張性に特徴がある．all-words WSD は現実の意味解析にとって必須の技術である．また様々な NLP タスクの学習システムに対して，語義の素性を追加することができる．更に KyWSD は日本語 WSD システムのベースラインとして手軽に利用できる．これらの点から KyWSD は役に立つと考えている．またここでは KyWSD の精度や拡張性を調べた．その実験から KyWSD 及び日本語の all-words WSD の持つ問題点も示す．

Shinnou Hiroyuki[1,a)]　Komiya Kanako[1,b)]　Sasaki Minoru[1,c)]　Mori Shinsuke[2,d)]

1. はじめに

本論文では日本語の all-words WSD について述べ，我々が開発・公開している日本語の all-words WSD システム KyWSD (KyTea for WSD) を紹介する．日本語の意味解析には all-words WSD システムが必須であり，KyWSD の有益性は高い．

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．これは現状の語義曖昧性解消が，主として，教師あり学習のアプローチをとっているため，対象単語が限定されてしまうことが大きな原因である．対象単語を限定せず，すべての単語に語義を付与する語義曖昧性解消は all-words WSD と呼ばれ，古くから研究されている [9]．ただし日本語に関しては，これまで all-words WSD システムは開発されておらず，手軽に利用できるシステムはない．そこで我々は日本語の all-words WSD システム KyWSD を開発し公開している．KyWSD はあらゆる意味解析システムに有用である．また教師あり学習手法を利用する様々なタスクにおいて，語義の素性を加えることができ，識別の精度を向上させることができる．

KyWSD は KyTea[*1] と呼ばれるシステムに対する all-words WSD のモデルである．KyTea にこのモデルを指定することで，プレーンなテキストに対して単語分割を行い，各単語に対してその品詞と語義を付与することができる．KyTea は，簡単に言えば，タスクに依存した形態素解析システムのモデルを学習するシステムである．ここでは all-words WSD を一種の形態素解析と捉えて，KyTea システムを利用して all-words WSD システムを実現している．KyTea は個々のタスクに適合するように学習を行うメカニズムが備わっている．KyWSD もこのメカニズムを利用できるため，拡張性が高い．例えば訓練データを追加する場合にも，全ての単語に語義を与える必要はなく，追加したい箇所にだけ語義を付与しておけば良い．このため領域適応に適したシステムとなっている．また識別した語義に信頼度を付与することも可能であるため，能動学習を用いることでシステムの改善も容易である．

2. 関連研究

all-words WSD はドメインを限定すれば通常の教師あり学習も可能である．実際に SemEval で行われた all-words WSD のタスクでも，いくつかの教師あり学習によるシステムが参加している．ただし教師あり学習はその拡張性に問題がある．

教師あり学習手法を用いない場合，all-words WSD の手法は知識ベースの手法か教師なし学習手法に分類でき

[1] 茨城大学工学部情報工学科
Ibaraki University, Nakanarusawa 4-12-1, Hiachi, Ibaraki 316–8511, Japan
[2] 京都大学学術情報メディアセンター
Kyoto University, Yoshida Honcho, Sakyouku, Kyoto 606–8501, Japan
[a)] hiroyuki.shinnou.0828@vc.ibaraki.ac.jp
[b)] kanako.komiya.nlp@vc.ibaraki.ac.jp
[c)] minoru.sasaki.01@vc.ibaraki.ac.jp
[d)] mori.shinsuke.8u@kyoto-u.ac.jp

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

る [7].

知識ベースの手法として，古典的には Lesk の手法 [8] がある．これは対象単語の周辺の単語集合と，対象単語の各語義の定義文中に現れる単語集合との重なり度合いを調べ，その度合いの大きい語義を選択するというものである．ただし一般に知識ベースの手法は語義の頻度の情報を利用していないために，精度が低いという問題がある．

教師なし学習手法には様々なタイプのものが存在するが [13][4][14]，近年は，語義列の生成モデルを定義し，ある種のヒューリスティックを導入することでプレーンなコーパスから生成モデルのパラメータを推定する手法が採られている [1][12][15][6]．教師なし学習手法は知識に基づく手法よりも精度は高く，更に改善が期待できる魅力的な手法ではある．しかし現状の教師なし学習手法では，付与する語義が概念になってしまうという問題がある．それは教師なし学習手法では，何らかのヒューリスティックを手がかりとしてパラメータを推定する形になっているが，現状の手法では，そのヒューリスティックとして本質的に「語義 a の周辺の文脈と語義 b の周辺の文脈が似ている場合，語義 a と語義 b は似ている」というものを使っているからである．この手がかりを利用しようとした場合，通常 a や b に曖昧性があるために，語義間の距離が必要になってしまう．語義が概念であればその距離は測定可能だが，辞書の語義に対してはその語義間の距離を測ることができない．これは語義の粒度の問題とも見なせる．一般に，辞書の語義は概念よりも粒度が細かいため，辞書の語義を付与する方がタスクとしては難しい． KyWSD が付与する語義は概念としての語義ではなく，辞書（岩波辞書）の語義である．

また教師なし学習手法による all-words WSD システムは，通常の WSD システムとは異なる入出力となっていることも注意すべきである．通常の WSD システムの入力は WSD の対象単語を含む文であり，出力はその対象単語の語義である．一方，教師なし学習手法による all-words WSD システムの入力はコーパスである．入力コーパス中のすべての単語に語義を付与する．しかし新たに対象単語を含む文を単独で入力しても，その対象単語に語義を付与することはできない．

また近年は語義の分散表現を求めることで all-words WSD を実現することも試みられている [10][2]．単語 w に対する i 番目の語義の分散表現 s_i とする．w の周辺文脈を分散表現 v で表現し，s_i と v の類似度を測り，最も類似度の高い i に対する語義を識別結果とすることで，WSD が行える．この手法は知識ベースの手法の一種であり，この手法も精度が低いという問題がある．一般に MFS (Most Frequent Sense) よりも高い精度は得られない．そのため逆に分散表現を用いて，MFS を推定する方法も研究されている [11].

KyWSD は教師あり学習により構築されたものであり，しかも語義としては概念ではなく，辞書の語義である点が従来までのシステムと大きく異なっている．

KyWSD と 類似の研究としては Hatori の研究 [3] がある．Hatori の研究も KyWSD と同様，all-words WSD を系列ラベリング問題と見なし，CRF を用いていることで，それを解決している．ただしこの研究でも語義は概念である．そのため CRF が可能となっている．KyWSD のように辞書の語義を付与する場合，単語 w の語義 s は，w 以外の単語には付与されることはなく，現実的には CRF を用いることはできない．そのため KyWSD は CRF の代わりに点推定を用いている．

3. KyTea

KyTea は，基本的には，単語分割システムを構築するためのツールと見なせる．一般に，単語分割は系列ラベリング問題としてモデル化できるが，KyTea では入力テキストの各文字間に分割されるかされないかのフラグを与える 2 値問題のタスクとしてモデル化する．そして学習にはその位置の前後の文字列の情報のみを用いた SVM あるいはロジステック回帰が使われる．このため KyTea の訓練データは単語を空白区切りで分割したテキストという簡易な形でよいため，モデルの拡張やモデルの領域適応が容易であるという特徴をもつ．

KyTea は更に単語分割する際に，その単語にタグを付与することも可能である．訓練データの単語に所望のタグを付与しておけば，分割した単語に適切なタグを付与するモデルが学習できる．通常，タグとしては品詞が設定されるが，それ以外にもその単語の「読み」あるいは固有表現抽出のための BIO タグなどを与えることで，様々な応用が可能となる．

本論文ではこの単語に与えるタグとして，その単語の語義を設定する．これによって語義タグ付きコーパスから all-words WSD が構築できる．

4. システムの概要

KyWSD は KyTea に語義タグ付きコーパスを訓練データとして与えることで構築される．訓練データとしては，東工大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．

上記コーパスを KyTea の訓練データの形式に変換するのは容易であるが，1 つだけ問題がある．それは日本語の用言の語尾変換の問題である．日本語の場合，一般に，用言（動詞や形容詞）は語幹と語尾から構成され，モダリティやテンスに応じて，語尾が変化する．通常，日本語の単語

```
> cat sample.txt
野球のＤＨの正式呼び名と意味を教えてください。

> kytea -model wsd.mod < sample.txt
野球/名詞-普通名詞/51783-0-0-0 の/助詞-格助詞/0 ＤＨ/UNK/UNK の/助詞-格助詞/0 正式/形状詞-一般/0
呼び名/名詞-普通名詞/53605-0-0-0 と/助詞-格助詞/0 意味/名詞-普通名詞/2843-0-0-1 を/助詞-格助詞/0
教え/動詞-一般-語幹/5541-0-0-2 て/助詞-接続助詞/0 ください/動詞-非自立可能-語幹/13445-0-0-2
い/動詞-非自立可能-語尾/0 。/補助記号-句点/0

>kytea -model wsd.mod -notag 1 -out conf < sample.txt
野球/51783-0-0-0 の/0&39930-0-1-3&40065-0-0-0 ＤＨ/UNK の/0&39930-0-1-3&39930-0-1-1
正式/0 呼び名/53605-0-0-0 と/0&37713-0-0-1&37446-0-0-2
意味/2843-0-0-1&2843-0-0-2&2843-0-0-3 を/0 教え/5541-0-0-2&5541-0-0-1&5541-0-0-3
て/0&35369-0-0-0 ください/13445-0-0-2&0 い/0&1707-0-0-2&52935-0-0-3 。/0
   ・・・ (omit) ・・・
1 0.999999&7.94354e-07&1.23533e-07 1 1&6.47248e-08&3.92486e-08 1 1
1&1.8927e-09&1.8105e-09 0.807761&0.108979&0.0807573 1 0.863406&0.135187&0.0012201
1&4.35077e-09 0.999236&0.00076433 0.999999&1.22639e-07&8.67671e-08 1
```

図 1 KyWSD の実行例

分割では語幹と語尾をまとめて，一つの単語として扱うために，単語分割の情報だけを与えられたデータでは，同じ単語でも別単語として扱われてしまう．例えば “書く” という単語は，否定形にすると “書か” + “ない” となる．“書く” と “書か” は同じ単語であるが，表層の文字列だけの情報だと，別単語として扱われてしまう．この問題の対処のために，コーパス中の用言は語幹と語尾に分割することを行った．ただしサ変動詞「する」の語幹は固定されないので例外である．一方，カ変動詞「来る」の場合，語幹の読みは固定されないが，表層の「来」は固定されるので，これを語幹とした．

KyTea では辞書も学習に利用することができる．訓練データに出現しない単語に対しては，辞書からタグを付与する形になる．このため上記の問題と同様，辞書の用言の見出しも語幹に変換する．

システムの入出力例を図 1 に示す．KyTea によって学習されたモデルは `wsd.mod` である．入力はプレーンな日本語テキストファイル (`sample.txt`) であり，出力は入力テキストを単語分割し，各単語にその単語の品詞と語義を付与したものである．語義は内容語となる名詞，動詞の語幹，形容詞の語幹に付与しており，それ以外の単語には語義として 0 を与えている．また UNK は辞書と訓練コーパスのどちらにも出現しなかった単語を意味する．

またオプション`-notag 1` を指定することで，1 個目のタグを出力しない，つまり品詞タグを出力しないことができる．更にオプション`-out conf` を指定することで，タグの信頼度を出力することができる．`wsd.mod` はロジスティック回帰を用いたタグ推定を行っているために，信頼度は確率となっている．例えば，図 1 では単語「意味」の語義の出力は以下である．

```
意味/2843-0-0-1&2843-0-0-2&2843-0-0-3
```

これは「意味」の語義が，2843-0-0-1, 2843-0-0-2 及び 2843-0-0-3 の 3 つあることを示している．また「意味」の語義の信頼度の出力は以下である．

```
0.807761&0.108979&0.0807573
```

これは語義 2843-0-0-1,2843-0-0-2, 2843-0-0-3 のそれぞれの確率が 0.807761, 0.108979, 0.0807573 であることを示している．この信頼度の出力を利用することで能動学習を容易に導入できる仕組みをもっている．

5. 評価

5.1 精度

KyWSD の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．交差検定で精度は出せそうだが，それは学習手法の評価であり，システムの精度の評価には使えない．ここでは Senseval-2 における日本語辞書タスク [5] のデータを用いて，対象単語を定めた場合の語義識別の精度を調べた．対象単語は上記タスクの対象単語である名詞 50 単語と動詞 50 単語の計 100 単語である．上記タスクでは各対象単語に対してその対象単語を含む 100 文が用意され，合計 10,000 個のテストデー

タが存在する．

まず通常の教師あり学習手法を用いた場合の精度を調べた．上記タスクでは各対象単語に対して平均して 175 個の訓練データが存在する．以下の素性を利用して，SVM により各対象単語に対する分類器を作成した．

利用した素性は以下の 6 つ（e1 から e6）である．

e1: 対象単語の直前の単語
e2: 対象単語の直後の単語
e3: 対象単語の前方にある自立語 2 つ
e4: 対象単語の後方にある自立語 2 つ
e5: e3 のシソーラス番号
e6: e4 のシソーラス番号

10,000 個のテストデータのうち正しく語義を識別できたデータは 7244 個であり，識別精度は 0.7244 であった．この場合，F 値も 0.7244 である．

次に KyWSD にテストデータをプレーンなテキストとして与える．全ての単語に対して語義が付与されるが，対象となっている単語を正しく切り出し，その上で正しい語義を付与できたものを正解とする．10,000 個の対象単語に対して，その対象単語を正しく切り出せたのは 9,935 個であり，そのうち正しく語義を付与できたものは 6,258 個であった．つまり正解率は 0.6571 再現率は 0.6528 であり，F 値は 0.6549 となった．

SVM と比較すると精度は低いが，これは日本語の all-words WSD の問題設定が，通常の WSD の問題設定よりもより困難な形になっていることも原因である．通常の WSD の問題設定では，対象単語 w に対する語義の候補のリスト L_w が与えられている．このため解答となる語義は L_w の中から選べば良い．一方，all-words WSD の問題設定では L_w が与えられていない．日本語の場合，全く同じ表記の異なる単語が多数存在するために，w に対する語義の候補は必ず L_w よりも大きいリストになる．

例えば，単語“間”の読み方は 6 つあり，辞書ではそれぞれ別個の単語として登録されている．“あい (21)”，“あいだ (105)”，“あわい (1432)”，“かん (9518)”，“けん (15147)” そして “ま (48408)” である．括弧の数値は語義の番号を示している．この単語“間”は実際に Senseval-2 における対象単語であったが，その語義のリストは あいだ (105)” に対するものだけであった．この問題は従来までの日本語 WSD では無視されていた問題であるが，all-words WSD では深刻な問題となっている．上記の実験においても L_w に属さない語義を選択したことで識別を誤っているものが 1,372 個存在した．これらの解答を無視した場合，8,563 個の解答中，6,258 個が正解となり，正解率が 0.7623，F 値は 0.7076 となる．

5.2 拡張性

KyWSD の大きな特徴は拡張が容易なことである．単に訓練データを追加して学習させればよい．ここでは先の実験で用いたデータセットの中の訓練データを KyWSD で単語分割し，与えられている対象単語に語義を付与した訓練データを作成した．この訓練データを追加して KyWSD を再構築し，先のテストデータで精度を調べた．

10,000 個の対象単語に対して，その対象単語を正しく切り出せたのは 9,938 個であり，そのうち正しく語義を付与できたものは 6,986 個であった．つまり正解率は 0.7030 再現率は 0.6986 であり，F 値は 0.7008 となった．更に先に述べたように設定された語義候補以外の解答をしたもの 985 個を無視した場合，8,953 個の解答中，6,986 個が正解となり，正解率が 0.7803，F 値は 0.7394 となり，教師あり学習である SVM の F 値を超えることができた．

5.3 文書分類タスクへの利用

ここでは all-words WSD の応用として文書分類を行う．

文書分類は通常，文書を bag-of-words によりベクトル化することで処理される．ここでは all-words WSD を用いて，word を語義に変更し，bag-of-senses により文書分類を行ってみる．ネットニュース記事は 2003 年 11 月 25 日から 12 月 5 日までの 10 日間でニュースサイト http://news.goo.ne.jp/に掲載されたニュース記事である．5 カテゴリ (政治，経済，国際，社会，スポーツ) から集めた総数 316 文書のデータである．評価は Leave-one-out による交差検定で行う．また学習アルゴリズムとしては Naive Bayes を用いた．通常の bag-of-words の場合，正しくカテゴリを識別できたものは 316 個中 246 個であったが，単語を語義に変更した場合，正しい識別数は 247 個になった．

KyWSD により通常の学習システムに語義の素性を容易に導入できる．文書分類外にも様々なタスクに利用できる．

6. おわりに

ここでは我々が開発・公開している日本語の all-words WSD システム KyWSD を紹介した．KyWSD は点推定を基本にした単語分割学習システム KyTea を利用したものであり，その拡張性に特徴がある．Senseval-2 のデータを利用した実験により，簡易に拡張可能であり，拡張されたシステムは通常の教師あり学習のシステムと同等の精度を出すことが確認できた．KyWSD および日本語の all-words WSD の問題点として，表記が同じであるが読みが異なる単語では，可能な語義のリストが大きくなることがあげられる．これは形態素解析で解決できる部分もあるため，この部分の処理をどのようにシステムに組み込むかが当面の課題である．

参考文献

[1] Boyd-Graber, J. L., Blei, D. M. and Zhu, X.: A Topic Model for Word Sense Disambiguation, *EMNLP-CoNLL-2007*, pp. 1024–1033 (2007).
[2] Chen, X., Liu, Z. and Sun, M.: A Unified Model for Word Sense Representation and Disambiguation, *EMNLP-2014*, pp. 1025–1035 (2014).
[3] Hatori, J., Miyao, Y. and Tsujii, J.: Word Sense Disambiguation for All Words using Tree-Structured Conditional Random Fields, *COLING-2008*, pp. 43–46 (2008).
[4] Izquierdo-Beviá, R., Moreno-Monteagudo, L., Navarro, B. and Suárez, A.: Spanish all-words semantic class disambiguation using Cast3LB corpus, *MICAI 2006: Advances in Artificial Intelligence*, pp. 879–888 (2006).
[5] Kiyoaki Shirai: SENSEVAL-2 Japanese Dictionary Task, *SENSEVAL-2*, pp. 33–36 (2001).
[6] Komiya, K., Sasaki, Y., Morita, H., Shinnou, H., Sasaki, M. and Kotani, Y.: Surrounding Word Sense Model for Japanese All-words Word Sense Disambiguation, *PACLIC-29*, pp. 35–43 (2015).
[7] Kulkarni, A., Khapra, M. M., Sohoney, S. and Bhattacharyya, P.: CFILT: Resource conscious approaches for all-words domain specific WSD, *SemEval-2010*, pp. 421–426 (2010).
[8] Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, *the 5th annual international conference on Systems documentation*, pp. 24–26 (1986).
[9] Navigli, R.: Word sense disambiguation: A survey, *ACM Computing Surveys (CSUR)*, Vol. 41, No. 2, p. 10 (2009).
[10] Neelakantan, A., Shankar, J., Passos, A. and McCallum, A.: Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space, *EMNLP-2014*, pp. 1059–1069 (2014).
[11] Sudha Bhingardive, Dhirendra Singh, R. V. H. H. R. and Bhattacharyya, P.: Unsupervised Most Frequent Sense Detection using Word Embeddings, *HLT-NAACL 2015*, pp. 1238–1243 (2015).
[12] Tanigaki, K., Shiba, M., Munaka, T. and Sagisaka, Y.: Density Maximization in Context-Sense Metric Space for All-words WSD, *ACL-2013*, pp. 884–893 (2013).
[13] Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods, *ACL-95*, pp. 189–196 (1995).
[14] Zhong, Z. and Ng, H. T.: Word Sense Disambiguation for All Words without Hard Labor, *IJCAI-2009*, pp. 1616–1622 (2009).
[15] 谷垣宏一，徳本修一，撫中達司，匂坂芳典：文脈・語義対応の階層ベイズ推定による教師なし語義曖昧性解消，情報処理学会自然言語処理研究会，pp. NL–220–5 (2015).