Vol.2012-NL-207 No.3
2012/7/26

パラレルコーパスから自動獲得した用例に基づく語義曖昧性解消

Pulkit Kathuria[1,a)]　白井清昭[1,b)]

概要：本論文では，日本語学習者向けの読解支援システムで用いることを前提とし，精度を重視した用例に基づく語義曖昧性解消 (WSD) 手法について述べる．提案手法では，コロケーションと統語的関係の 2 つの観点から文の類似度を測り，辞書中の用例の中から最も似ていてかつ類似度が十分高い用例の語義を選択する．再現率を向上させるため，用例に基づく WSD 手法は Naive Bayes モデルと組み合わせて用いる．また，パラレルコーパスから語義ごとに例文を獲得し，用例データベースを拡張することで WSD の性能を向上させる．実験の結果，コーパスから自動獲得された例文の正解率は 85%であった．また，提案手法の WSD の正解率は 65%であり，ベースラインから 7%の改善が見られた．

キーワード：語義曖昧性解消, 用例, パラレルコーパス, 日本語学習支援

Example Based Word Sense Disambiguation with Automatically Acquired Examples from Parallel Corpus

Pulkit Kathuria[1,a)]　Kiyoaki Shirai[1,b)]

Abstract: This paper presents a precision oriented example based approach for word sense disambiguation (WSD) for a reading assistant system for Japanese learners. Our WSD classifier chooses a sense associated with the most similar sentence in a dictionary only if the similarity is high enough, otherwise chooses no sense. We propose sentence similarity measures by exploiting collocations and syntactic dependency relations for a target word. The example based classifier is combined with Naive Bayes model to compensate recall. We further improve WSD performance by automatically acquiring bilingual sentences from a parallel corpus. According to the results of our experiments, the accuracy of automatically extracted sentences was 85%, while the proposed WSD method achieves 65% precision which is 7% higher than the baseline.

Keywords: Word Sense Disambiguation, Example Sentence, Parallel Corpus, Support for Japanese Language Learning

1. はじめに

日本語学習者は辞書を使って単語の意味を調べる機会が多い．単語の意味は一般に複数あるが，辞書に記載されている全ての意味の語釈文を読み，現在読んでいるテキストの文脈内で該当する意味を見つけるのは，日本語を母語としない学習者にとっては負担が大きい．本研究は，日本語学習者のための読解支援システムとして，語義曖昧性解消 (Word Sense Disambiguation; WSD) によって単語の意味を自動的に推測し，その意味の語釈文と用例をユーザに提示するシステムの構築を目指している．WSD の機能を備えた日本語読解支援システムとしては「あすなろ」[*1] がある．あすなろは辞書として EDR 単語辞書 [*2] を用いているが，EDR 単語辞書は機械処理用の辞書であるため，不自然な語釈文も多い．また，あすなろは語釈文をユーザに

[1] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
[a)] pulkit@jaist.ac.jp
[b)] kshirai@jaist.ac.jp

[*1] http://hinoki.ryu.titech.ac.jp/asunaro/index-e.php
[*2] http://www2.nict.go.jp/r/r312/EDR/

提示するが，用例は提示しない．これに対し，本研究では辞書として和英辞書 EDICT [*3] を用いる．EDICT の語釈文は単語もしくは句という単純なものではあるが，語義ごとに日本語の例文とその英訳が用意されている．我々は，日本語学習者が単語の意味を理解する上で，用例を表示することは大きな役割を果たすと考えている．

本論文では，上記のような日本語読解支援システムのモジュールとして実装することを前提とした用例に基づく WSD 手法について述べる．我々の日本語読解支援システムの目標は，語義曖昧性解消を行うと同時に語義の用例をユーザに提示することであるため，用例に基づく手法はごく自然にシステムに組み込むことができる．現在，WSD は教師あり学習に基づく手法が主流である [10], [11] が，学習データとなる語義タグ付きコーパスは作成コストが高い．一方，日本語読解支援システムではテキストに出現する全ての単語の語義を決定できることが望ましいが，全ての単語について十分な量の語義タグ付きコーパスを用意することは難しい．そのため，本研究では，語義タグ付きコーパスを使用せず，EDICT に含まれる語義の例文を用例データベースとして利用する．

本研究の用例に基づく WSD 手法では，対象単語を含む入力文と最も類似した例文を用例データベースから検索し，その例文中の対象語が持つ語義を選択する．また，類似度が低いとき，すなわち十分に似ている例文を発見することができないときには，語義を選択しない．このような手法は，WSD の再現率は低い半面，精度は高いという利点がある．再現率を向上させるために，本研究では用例に基づく WSD 手法と Naive Bayes モデルを組み合わせて用いる．さらに，日英パラレルコーパスから，語義ごとに新しい例文を自動的に獲得し，用例データベースを拡張する手法を提案する．これにより，WSD 手法の性能が向上するだけでなく，日本語学習者に提示する例文数を増やすこともできる．

以下，2 章では本論文で提案する WSD 手法について述べる．3 章では用例データベースを拡張する方法について述べる．4 章では提案手法の評価実験について述べる．5 章では関連研究について議論する．最後に 6 章でまとめと今後の課題について述べる．

2. WSD 手法

テキスト中の単語の語義の曖昧性を解消するために，本研究では 2 つの手法を組み合わせる．1 つは用例に基づく WSD 手法，もうひとつは Naive Bayes モデルである．

2.1 用例に基づく WSD

本研究では語義を定義する辞書として EDICT を用いる．

[*3] http://www.csse.monash.edu.au/%7Ejwb/edict.html

【話】
S_1 { story, talk , conversation, speech, chat }
E_{11}: そんな話は知りたくない。
I don't want to know that kind of story.
E_{12}: もうこれ以上その話を私に聞かせないで下さい。
Please let me not hear of that story any more.
S_2 { discussions, argument, negotiation }
E_{21}: 3 時間議論したが，我々は話がまとまらなかった。
After 3 hours of discussion we got nowhere.

図 1 EDICT における「話」の語釈文と例文

EDICT では，日本語単語を見出しとし，その語釈文が英語で記述されている．さらに，語義毎に，その語義の例文のリストが記載されている．例文は日本語と英語の対訳となっている．EDICT に記載されている「話」の語釈文及び例文 (一部) を図 1 に示す．

ここでは，EDICT に記載されている例文の集合を用例データベースとして用いる．ただし，学習者に提示する際には日英両方の例文を提示するが，WSD の際は日本語の例文のみを利用する．対象語を含む入力文 I に対し，用例データベース中の例文 E との類似度 $sim(I,E)$ を計算し，類似度が最大となる例文をひとつ選択する．ただし，最大の類似度が閾値 T より小さければ，つまり入力文と十分似ている例文が見つからないときは，対象語の語義を決定せずに不明と判定する．すなわち，ここでは WSD の再現率より精度を重視している．

文間類似度 $sim(I,E)$ は，コロケーションの類似度 $col(I,E)$ と，統語的関係の類似度 $syn(I,E)$ の和とする．

$$sim(I,E) = col(I,E) + syn(I,E) \qquad (1)$$

以下，$col(I,E)$，$syn(I,E)$ の定義について述べる．

2.1.1 コロケーションの類似度

コロケーションの類似度 $col(I,E)$ は，対象語の直前・直後の文脈の類似度を測る指標である．まず，対象語を含む n-gram ($n = 4,5,6$) を抽出する．例えば 4-gram の場合，$w_{-3}w_{-2}w_{-1}w_0$，$w_{-2}w_{-1}w_0w_1$，$w_{-1}w_0w_1w_2$，$w_0w_1w_2w_3$ の 4 つの単語列を取り出す．ただし，w_0 は対象語を表わす．抽出した n-gram のうちどれかひとつでも一致するものがあれば，n-gram の長さに応じたスコアを与える．具体的には，$col(I,E)$ は式 (2) のように定義する．なお，n-gram に対して与える重みは直観により決めている．

$$col(I,E) = \begin{cases} 1 & \textbf{if} \quad \text{one of 6-grams is same} \\ 0.75 & \textbf{elif} \ \text{one of 5-grams is same} \\ 0.5 & \textbf{elif} \ \text{one of 4-grams is same} \\ 0 & \textbf{otherwise} \end{cases} \qquad (2)$$

図 2　文節の係り受け解析の例

2.1.2　統語的関係の類似度

まず, 類似度を計算するそれぞれの文に対し, CaBoCha[*4] を用いて文節の係り受け解析を行い，対象語に関する統語的関係 r を抽出する．r の定義を式 (3) に示す．

$$r = w_1 - rel - w_2 \tag{3}$$

$$rel = \begin{cases} p & \textbf{if } \text{係り元文節に助詞 } p \text{ がある} \\ \text{連体修飾} & \textbf{elif } w_2 \text{が名詞} \\ \text{連用修飾} & \textbf{otherwise} \end{cases} \tag{4}$$

w_1 および w_2 は係り受け関係にある文節の係り元および係り先文節の主辞の基本形を表わす．ただし，w_1 もしくは w_2 のどちらか一方は対象語 t であるとする．一方，rel は統語的関係のタイプで,「が」「を」「に」などの助詞,「連体修飾」,「連用修飾」のいずれかとする．

例えば，以下の I_1 中の「話」の語義を決定したいとき

I_1: 犯人が捕まったという話は滅多に聞かない．

図 2 のような文節係り受け解析の結果から式 (5) のような統語的関係が得られる．

$$\begin{aligned} r_1 &: \text{捕まる - 連体修飾 - 話} \\ r_2 &: \text{話 - は - 聞く} \end{aligned} \tag{5}$$

次に，$syn(I, E)$ の定義を以下に示す．

$$syn(I, E) = \sum_{(r_i, r_e) \in R_I \times R_E} s_r(r_i, r_e) \tag{6}$$

$$s_r(r_i, r_e) = \begin{cases} s_w(r_i(w_1), r_e(w_1)) & \textbf{if } r_i(w_2) = r_e(w_2) = t \\ & \quad and\ r_i(rel) = r_e(rel) \\ s_w(r_i(w_2), r_e(w_2)) & \textbf{if } r_i(w_1) = r_e(w_1) = t \\ & \quad and\ r_i(rel) = r_e(rel) \\ 0 & \textbf{otherwise} \end{cases} \tag{7}$$

$$s_w(w_i, w_j) = \begin{cases} 1 & \textbf{if } w_i = w_j \\ \dfrac{x}{8} & \textbf{otherwise} \end{cases} \tag{8}$$

$syn(I, E)$ は，文 I,E から抽出した統語的関係 r_i,r_e の類似度 $s_r(r_i, r_e)$ の和とする (式 (6))．統語的関係の類似度 $s_r(r_i, r_e)$ の定義は式 (7) であり，対象語 t に同じ関係タイプ rel で係る単語の類似度，もしくは同じ関係タイプ rel で対象語 t の修飾を受ける単語の類似度とする．単語間の類似度 $s_w(w_i, w_j)$(式 (8)) は分類語彙表 [14] を用いて求める．式 (8) の x は，2 つの単語 w_i,w_j の分類語彙表のコード番号の共通上位接頭辞の長さである．分類語彙表では単語の意味は 7 桁のコード番号で定義されており，x を 8 で割ることで類似度の値の範囲が [0,1] になるように正規化している．

例えば，入力文 I_1 と例文 E_{11} の統語的関係の類似度は，I_1 の関係「話 - は - 聞く」と E_{11} の関係「話 - は - 知る」の類似度で決まる．一方，入力文 I_1 と例文 E_{21} については，同じ関係タイプの統語的関係が存在しないため，類似度は 0 となる．

$$\begin{aligned} syn(I_1, E_{11}) &= s_r(\text{話 - は - 聞く, 話 - は - 知る}) \\ &= s_w(\text{聞く, 知る}) = 0.375 \\ syn(I_1, E_{21}) &= 0 \end{aligned}$$

2.1.3　共通語の統語的関係を考慮したモデル

2.1.2 で述べたように，統語的関係の類似度を計算する際には対象語を含む統語的関係しか考慮しない．しかしながら，対象語を含みかつ共通のタイプを持つ統語的関係は必ずしも存在せず，そのような場合には類似度を見積ることができない．そこで，2 つの文で共通して出現する単語に着目し，共通語を含む統語的関係も類似度計算の対象とするモデルを提案する．このモデルでは，式 (7) における t は対象語だけでなく 2 つの文で共通して出現する単語となる．

例えば，入力文 I_1 と例文 E_{12} では，対象語「話」の統語的関係のうち共通の関係タイプを持つものは存在しないが,「聞く」という共通語が存在し,「聞く」の統語的関係から類似度 $syn(I_1, E_{12})$ は以下のように計算される．

$$\begin{aligned} &syn(I_1, E_{12}) \\ &= s_r(\text{滅多に - 連用修飾 - 聞く, もう - 連用修飾 - 聞く}) \\ &= s_w(\text{滅多に, もう}) = 0.5 \end{aligned}$$

以下，用例に基づく WSD 手法のうち，$syn(I, E)$ を計算する際，対象語の統語的関係のみを比較するモデルを RTW[*5]，二文間の共通語 (対象語を含む) の統語的関係を比較するモデルを RCW[*6] と呼ぶ．RCW は RTW と比べて再現率は向上するが精度は低下することが予想される．両者の性能の比較については 4 節で述べる．なお，RTW, RCW ともにコロケーションの類似度 $col(I, E)$ も文間の類似度計算に用いられることに注意していただきたい．

2.2　Naive Bayes モデル

2 つ目の WSD 手法は Naive Bayes モデルである [8]. Naive Bayes モデルでは，式 (9) のように，確率モデル $P(s|F)$ が最大となるような語義 s をひとつ選択する．

$$s = \arg\max_s P(s|F) = \arg\max_s P(s) \prod_{f_i \in F} P(f_i|s) \tag{9}$$

本研究では，素性集合 F として，2.1.2 で述べた統語的関係,

[*4] http://code.google.com/p/cabocha/

[*5] a model considering syntactic Relations with respect to Target Word

[*6] a model considering syntactic Relations with respect to Common Words

対象語を含む 2-gram および 3-gram，文中に含まれる自立語の基本形 (Bag of Words) とした．これらは WSD でよく用いられる素性である．確率モデルのパラメタ $P(s)$ および $P(f_i|s)$ は，語義タグ付きコーパスではなく，EDICT に記載されている例文の集合から最尤推定で学習した．Naive Bayes モデルは，入力に対して常に語義を 1 つ選択することから，用例に基づく WSD 手法よりもロバストな手法である．

2.3 混合モデル

混合モデルは，用例に基づく WSD 手法と Naive Bayes モデルを組み合わせた手法である．前者は精度を重視していることから，混合モデルではまず最初に用例に基づく WSD 手法で語義を決定する．閾値 T よりも大きい類似度を持つ例文が見つからず，語義が選択できなかったときには，Naive Bayes モデルを用いて語義を決定する．

3. 用例データベースの拡張

本節では，語義の用例をパラレルコーパスから自動獲得し，用例データベースを拡張する方法について述べる．我々の読解支援システムでは日英両方の例文をユーザに提示するため，ここでの目的はある語義の用例となる日英の対訳対をパラレルコーパスから抽出することである．

本研究では，パラレルコーパスとして JENAAD[13] を用いた．JENAAD は日英の対訳新聞記事に対する文の自動アライメントによって得られた約 150,000 対の文からなるコーパスである．前処理として，BerkleyAligner [*7] を用いて単語のアライメントを行い，Morpha[9] を用いて lemmatization を行った．

語義の用例を獲得するためには，コーパス中に出現する対象語の語義を決定する必要がある．本研究では，語義の語釈文中の語とパラレルコーパスの英文中の語を照合することで語義を決定する．その手続きを以下に述べる．対象語 t の語義 S に対し，日本語文 Ja と英語文 En の組が以下の条件を満たすとき，(Ja, En) を語義 S の用例として獲得する．

- 日本語文 Ja が対象語 t を含む．
- t_e を t と対応関係にある En 中の語とする．t_e もしくは t_e を含む複合語が，語義 S の語釈文中のいずれかの語もしくは複合語と等しい．
  (例) 「出る」の語義 S_1 {to go out, to exit, to leave} について考える．「出る」の訳語 t_e が exit もしくは leave なら，(Ja, En) を S_1 の用例として獲得する．また，t_e が go であり，かつ En において t_e の直後の単語が out なら，その例文を S_1 の用例として獲得する．
- t_e はたかだかひとつの語義の語釈文中の単語とマッチする．
  (例) 「作る」の 2 つの語義 S_1 {to prepare, to brew} と S_2 {to prepare, to make out, to write } について考える．t_e が prepare のときは，その用例の語義が S_1 か S_2 かはわからないため，用例として獲得しない．
- 語釈文の中に括弧で囲まれた注釈があるとき，括弧内の自立語のひとつが En に含まれる．
  (例) 「出す」の語義 S_6 {to produce (a sound)} について考える．t_e が produce であり，かつ英語文 En に sound が出現するとき，(Ja, En) を S_6 の用例として獲得する．

このような制約により，パラレルコーパスに存在する全ての語義の用例を獲得することはできないが，獲得された用例は正しい可能性が高い．すなわち，ここでは信頼性の高い語義の用例のみを獲得する．

上記の手続きにより日英の文の組が獲得され，用例データベースに追加されるが，用例に基づく WSD 手法ならびに Naive Bayes モデルの学習には日本語の用例のみを用いる．用例データベースの拡張によって両方の WSD 手法の性能の向上が期待できる．

4. 評価実験

4.1 実験データ

提案手法を評価するために，開発データと評価データという 2 種類の正解語義付きの例文セットを用意した．開発データ (D_d) における対象単語数は 17(名詞 8，動詞 8，形容詞 1)，テスト文の数は 330 である．これは用例に基づく WSD 手法の設計・開発，およびパラメータ T の最適化に用いた．一方，評価データ (D_e) の対象単語数は 49(名詞 23，動詞 24，形容詞 2)，テスト文の数は 937 である．このデータは WSD 手法の評価に用いた．D_e における対象単語は D_d における 17 個の対象単語を全て含んでいる．ただし，共通の対象単語についても，D_d と D_e のテスト文は互いに異なる．テスト文は毎日新聞 1994 年の記事から抽出し，対象語の正しい語義を人手で付与した．

4.2 用例データベースの自動拡張の評価

パラレルコーパスから自動獲得された用例を追加し，用例データベースを拡張したときの結果を表 1 に示す．T_d，T_e は開発データ，評価データにおける対象単語の集合を表わす．E+ は自動拡張後の用例データベース (EDICT にもともと記載されていた例文も含む) の値を示している．例文の数がおよそ 1.5 倍に増加し，1 つの語義当たりの例文数も増えている．また，例文が 1 つもない語義の数も 70 から 65 に減少している．語義の例文が 1 つもない場合，本研究における WSD 手法ではそのような語義は選択されることがないため，深刻な問題を生じるが，例文の自動獲得によりこの問題が多少緩和されている．なお，T_e の 49 個

[*7] http://code.google.com/p/berkeleyaligner/

表 1　用例データベースの自動拡張の結果

	対象語数	1 語当たりの平均語義数	例文数		1 語義当たりの平均例文数		例文のない語義数	
				E+		E+		E+
T_d	17	3.41	4,252	7,763	73.3	134	10	8
T_e	49	4.65	10,998	16,468	48.2	72.2	70	65

の対象単語における語義の総数は 228 であるが，このうちのおよそ 36%の語義に対してパラレルコーパスから新しい例文を獲得することができた．

表 2　自動獲得された例文の評価

文数	適切	不適切
652	553 (85%)	99 (15%)

自動獲得された 5,470 の例文のうち，それぞれの語義について最大 10 個の例文をランダムに選択 (例文数が 10 以下なら全て選択) し，獲得された例文が適切であるかを人手で判定した．結果を表 2 に示す．およそ 85%の例文は用例として適切であることがわかった．提案手法では，対象語と対訳関係にある英単語が語釈文中の単語と一致するかをチェックすることで対象語の語義を決めているが，表 2 はこの方法が有効であることを示唆する．

不適切と判定された 99 の例文のうち，5 つの例文については，日本語の形態素解析の誤りによって対象単語を正しく認識できていなかった．残りの 94 個の例文については対象語の語義の判定が誤っていた．この誤りの多くは，1 つの英単語が複数の語義の訳語として使われているときに生じている．例えば，「中」という単語に S_1 { inside, in } という語義がある．この語義は，「机の中」のように，何かが空間的に他のものの内部にあるという意味である．しかし「中」の他の語義も in に訳されることがある．例えば「自由の中」が「in freedom」に訳されていることがあったが，この「中」の語義は S_1 ではないのにも関わらず，in と訳されているために，S_1 の語義の例文として誤って抽出される．また，「人」という語には S_1 { man, person }，S_2 { mankind, people } という 2 つの語義がある．語義 S_1 の「人」が person の複数形 people に訳されているとき，この例文は誤って語義 S_2 の例文として抽出される．これらのようなケースでは，語釈文をチェックするだけでは対象語の語義を区別することが難しい．

4.3　WSD の実験結果

表 3 は，用例に基づく WSD 手法 (RTW，RCW)，Naive Bayes モデル (NB)，ベースライン (BL)，および混合モデル (RTW+NB，RCW+NB) の開発データ D_d における精度 (P)，再現率 (R)，F 値 (F) を示している．ベースラインは，用例データベース中の例文の数が最も大きい語義を常に選択するシステムである．ただし，例文数最大の語義が複数あるときは，その中から語義をランダムに選択する．また，NB, BL, RTW+NB, RCW+NB は常に語義を 1 つ選択するため，精度，再現率の区別をせず，単に正解率 (正解語義とシステムの出力語義が一致した割合) を示している．

用例に基づく WSD 手法 RTW，RCW は，類似度最大の例文の類似度が T より大きいときのみに語義を出力するが，容易に予想されるように，T の値を大きく設定すると，精度は向上したが再現率は低下した．また，混合モデル RTW+NB，RCW+NB の正解率は，個々の手法，すなわち RTW(もしくは RCW) の F 値および NB の正解率よりも高いことから，精度を重視した用例に基づく手法と Naive Bayes モデルを組み合わせる手法は有効であると言える．

RTW と RCW を比較すると，精度は RTW の方が高いが，再現率や F 値は RCW の方が高い．しかし，混合モデルで両者を比較すると，RTW+NB の方が RCW+NB よりも正解率が高い．これは，Naive Bayes モデルと組み合わせるときは，より精度の高い RTW の方が適しているためと考えられる．

用例データベースの拡張により，RTW，RCW ともに再現率と F 値が向上した．精度については，再現率がほぼ同じとなる閾値で比較すると，拡張前と拡張後ではそれほど大きな変化は見られない．例えば，RTW で $T = 0.3$ のときと RCW で $T = 0.9$ のときはともに再現率は 0.35 程度だが，精度も前者が 0.76，後者が 0.77 と大きな差はない．4.2 項で示したように，自動獲得した例文の正解率は 85%と比較的高いことから，WSD の再現率だけでなく精度の向上も見込まれるが，実験結果からは精度の顕著な改善は見られなかった．一方，混合モデルについて用例データベースの拡張の効果を調べると，RTW+NB では正解率はほぼ同じだが，RCW+NB では正解率が低下している．RTW と RCW を同じ閾値 T で比較すると，RCW の精度は RTW よりも悪く，このことが混合モデルでの正解率の低下を招いていると考えられる．

閾値 T の最適化については，混合モデルの正解率は T に大きく依存しないものの，RTW+NB，RCW+NB ともに $T = 0$ が最適値であると言える．

表 4 は評価データ D_e における WSD 手法の結果である．開発データ D_d の実験結果とは異なる傾向がいくつか見ら

表 3　D_d における WSD の実験結果

	RTW			RCW			RTW+NB	RCW+NB
T	P	R	F	P	R	F		
0.0	0.72	0.48	0.58	0.66	0.53	0.59	0.67	0.66
0.3	0.76	0.35	0.48	0.68	0.45	0.54	0.67	0.66
0.6	0.83	0.26	0.39	0.73	0.36	0.48	0.66	0.66
0.9	0.90	0.16	0.28	0.79	0.29	0.42	0.64	0.66
E+								
0.0	0.70	0.57	0.63	0.62	0.59	0.60	0.67	0.61
0.3	0.71	0.55	0.62	0.63	0.55	0.58	0.67	0.61
0.6	0.74	0.45	0.56	0.67	0.49	0.56	0.65	0.60
0.9	0.77	0.36	0.49	0.68	0.40	0.51	0.62	0.59

	NB	BL
	0.62	0.59
E+	0.60	0.61

表 4　D_e における WSD の実験結果

	RTW			RCW			RTW+NB	RCW + NB
T	P	R	F	P	R	F		
0.0	0.64	0.44	0.52	0.60	0.49	0.53	0.57	0.57
0.3	0.66	0.32	0.43	0.64	0.42	0.51	0.56	0.57
0.6	0.73	0.22	0.34	0.67	0.33	0.44	0.55	0.56
0.9	0.81	0.12	0.21	0.71	0.24	0.36	0.55	0.56
E+								
0.0	0.68	0.56	0.62	0.65	0.60	0.63	0.65	0.64
0.3	0.68	0.52	0.59	0.66	0.55	0.60	0.65	0.63
0.6	0.70	0.43	0.53	0.68	0.46	0.55	0.65	0.63
0.9	0.77	0.30	0.44	0.72	0.37	0.49	0.64	0.62

	NB	BL
	0.54	0.51
E+	0.60	0.58

れる．まず，用例データベースの自動拡張後 (E+)，全ての手法について WSD の精度，再現率，F 値または正解率が向上している．特に，開発データでは RTW，RCW については精度の大きな向上は見られなかったが，評価データでは顕著な改善が見られる．評価データの対象単語数，テスト文数は開発データと比べて多いことから，評価データの実験結果は開発データの結果よりもおそらく信頼性が高いだろう．したがって，パラレルコーパスから例文を抽出し用例データベースを自動拡張する手法は，WSD の性能向上に有効であるといえる．また，2 つの混合モデルを比較すると，用例データベースの自動拡張後の RTW+NB は RCW+NB よりも正解率が高いが，その差は開発データほどは大きくない．開発データで最適化された閾値 $T = 0$ のとき，自動拡張後の RTW+NB の精度は 0.65 であり，ベースラインよりも 7%高かった．

4.3.1　考察

評価データの評価値は開発データと比べて全般的に低い．これは，以下に挙げる要因から，評価データの WSD タスクは開発データよりも難しかったためと考えられる．

- 対象単語が持つ語義の数が多い．(D_e で 4.7，D_d で 3.2．表 1 より)
- 1 つの語義当たりの例文数が少ない．(拡張前のとき D_e で 48.2，D_d で 73.3．拡張後のとき D_e で 72.2，D_d で 134．表 1 より)
- ベースラインの値が低い．(拡張前のとき D_e で 0.51，D_d で 0.59．拡張後のとき D_e で 0.58，D_d で 0.61．表 4, 3 より)

Naive Bayes モデルの正解率は，用例データベース拡張前，拡張後のいずれのときも，ベースラインと比べてそれほど向上していない．この要因のひとつは，語義タグ付きコーパスではなく，辞書に記載されている例文を訓練データとして使用しているためと考えられる．式 (9) の Naive Bayes モデルにおける $P(s)$ は語義の頻度分布を反映している．語義タグ付きコーパスを訓練データとしたときは語義の頻度分布も学習できるが，EDICT で語義ごとに記載されている例文の数は語義の頻度分布に従っていることは保証されていないと考えられる．したがって，語義の頻度分布を EDICT の例文集合から学習することは難しい．また，提案手法で自動獲得された例文から語義の頻度分布を学習することも困難である．提案手法では信頼性の高い例文のみをパラレルコーパスから抽出しているため，抽出後の語義の分布が真の分布に近いとは言えない．他の機械学習アルゴリズムについて検討するため，サポートベクターマシーン (Support Vector Machine; SVM) を同じ訓練データから学習したが，その精度はベースラインよりも悪かった．実際に SVM の出力を調べてみると，全てのテスト文に対して同じ語義を選択することが多く，有効な分類器を学習できなかった．以上から，辞書の例文もしくは提案手

法でコーパスから自動獲得した例文は，語義タグ付きコーパスと比べて，教師あり機械学習アルゴリズムの訓練データとしては適切でないといえよう．

本論文では，用例に基づく WSD 手法のパラメタ T は全ての対象単語に対して同一の値と定めている．開発データでの実験結果から $T = 0$ を最適としているが，調べてみると，対象単語によっては $T = 0$ では精度が低いこともあった．したがって，対象単語毎に最適な T が調整できれば，WSD の更なる性能向上が期待できる．この際，開発データのような語義タグ付きコーパスを使わないでパラメタを調整する方が望ましい．語義の数や，語釈文の意味的な近さなどから WSD の難易度を定量化し，難易度の高い対象語については T を高く設定することで精度を高く保つ方法などが考えられる．

5. 関連研究

本研究における WSD は，機械翻訳における訳語選択の問題とみなすことができる．Dagan は，目標言語のコーパスにおける語の共起情報をもとに語義の曖昧性を解消する手法を提案している [3]．Lee らは，語の訳語を決定する際，まず語義を決定し，その語義に対応する候補の中から最終的な訳語を決定する二段階の訳語選択手法を提案している [7]．また，一般に WSD は機械翻訳のために必要な要素技術と位置付けられており，WSD によって翻訳の質が向上したという報告もなされている [1], [2]．これに対し，提案手法は原言語の文の情報のみで語義を決定しており，目標言語の情報は用いていないが，原言語，目標言語の両方の情報を併用する手法は今後検討するべきである．

用例に基づく WSD も過去に研究が行われている．日本語を対象とした研究としては Fujii らによる報告がある [5]．彼らの手法は，動詞の語義を決定する際，格要素となる名詞の類似度と，格がどの程度 WSD に影響を及ぼすかを定量化した格の重みを基に入力文と用例データベース中の例文の類似度を計算する．さらに，語義付与のコストを軽減するために，用例集合から WSD の用例データベースとして有効な部分集合を選択する selective sampling の手法を提案している．彼らの手法では対象語は動詞に限られているが，本研究では名詞，形容詞も WSD の対象とする．一方，Shirai らは，岩波国語辞典に記載されている例文と入力文との類似度を測ることで語義曖昧性解消を行う手法を提案している [12]．岩波国語辞典にはごく簡単な例文しか載っていないため，文間の類似度を正確に測れないことも多い．これに対し，本研究で用いている EDICT の方が比較的長い例文が多いため，用例に基づく WSD 手法に適している．また，文の類似度を測る際，Shirai らの手法では統語的関係のみを利用するが，提案手法ではコロケーションと統語的関係の両方を用いる．

語義の用例をコーパスから自動獲得する研究も過去にいくつか報告されている．Fujita らは，岩波国語辞典の例文を利用し，その例文を含む (例文よりも長い) 文をコーパスから自動的に抽出することによって，訓練データとなる語義付き例文の数を増加させる手法を提案している [6]．しかし，彼女らの手法で抽出される例文は，辞書の例文と似ている文しか抽出できないという問題点がある．WSD の訓練データとしては多様な文から構成されたコーパスの方が望ましく，そのような観点からは本研究の方が優れている．また，Melo らは，パラレルコーパスおよび語義が対応付けられた多言語辞書を用いて，語義ごとに新しい例文を獲得する手法を提案している [4]．彼らの手法では，機械学習された WSD の分類器を用いて原言語，目標言語の両方の単語の語義の曖昧性を解消し，2 つの言語の語義を同時にチェックすることで獲得される例文の品質を向上させている．彼らの手法では原言語，目標言語の両方について作成コストの高い語義タグ付きコーパスを必要とするのに対し，本研究では語義タグ付きコーパスを必要としない．また，Melo らは，獲得した例文を辞書に追加することを目的とし，得られた例文集合の中から代表的な例文の部分集合を決める手法を提案しているが，獲得した例文を WSD に利用していない．

6. おわりに

本研究では，精度を重視した用例に基づく WSD 手法を提案した．コロケーションと統語的関係に着目して文の類似度を定量化し，用例データベースの中から最も似ている例文を選択することで語義の曖昧性を解消する．このとき，閾値 T 以上の類似度を持つ例文が見つかったときのみ，つまり確信度の高いときのみ語義を選択する．また，よりロバストな Naive Bayes モデルと組み合わせることで WSD の再現率を向上させた．また，パラレルコーパスから語義を識別した上で用例を獲得し，用例データベースを拡張する手法を提案した．例文の拡張により WSD の性能が向上することが確認できた．提案手法の最終的な WSD の正解率は 65%であり，ベースラインよりも 7%高かった．

今後の課題としては，4.3.1 で議論したように，用例データベースから語義の分布を学習することは難しいことから，これを語義タグ付きコーパスなしに推定する方法を探究することが重要である．また，本研究の用例データベースは日英の対訳文の集合であるが，WSD の際には日本語の文しか利用していない．今後は英語文から得られる情報も利用することで WSD の性能の向上を図りたい．

参考文献

[1] Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72,

Prague, Czech Republic, 2007.

[2] Yee Seng Chan and Hwee Tou Ng. Word sense disambiguation improves statistical machine translation. In *45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 33–40, Prague, Czech Republic, 2007.

[3] Ido Dagan and Alon Itai. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*, 20(4):563–596, December 1994.

[4] Gerard de Melo and Gerhard Weikum. Extracting sense-disambiguated example sentences from parallel corpora. In *Proceedings of the 1st Workshop on Definition Extraction*, WDE '09, pages 40–46, 2009.

[5] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597, 1998.

[6] Sanae Fujita and Akinori Fujino. Word sense disambiguation by combining labeled data expansion and semi-supervised learning method. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 676–685, 2011.

[7] Hyun Ah Lee and Gil Chang Kim. Translation selection through source word sense disambiguation and target word selection. In *Proceedings of the 19th International Conference on Computational Linguistics*, volume 1 of *COLING '02*, pages 1–7, 2002.

[8] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*, chapter 7. MIT Press, 1999.

[9] Guido Minnen, John Carroll, and Darren Pearce. Robust, applied morphological generation. In *Proceedings of the First International Natural Language Generation Conference*, pages 201–208, 2000.

[10] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69, 2009.

[11] Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. SemEval-2010 task: Japanese WSD. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 69–74, 2010.

[12] Kiyoaki Shirai and Takayuki Tamagaki. Word sense disambiguation using heterogeneous language resources. In *First International Joint Conference on Natural Language Processing*, pages 614–619, 2004.

[13] Masao Utiyama and Hitoshi Isahara. Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1 of *ACL '03*, pages 72–79, 2003.

[14] 国立国語研究所. 分類語彙表. 大日本図書, 2004.