JAIST Repository
https://dspace.jaist.ac.jp/

Title	辞書の語義立てに基づく語義曖昧性解消に関する研究
Author(s)	玉垣, 隆幸
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1804
Rights	
Description	Supervisor:白井 清昭, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

辞書の語義立てに基づく語義曖昧性解消に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

玉垣 隆幸

2004 年 3 月

修 士 論 文

辞書の語義立てに基づく語義曖昧性解消に関する研究

指導教官　白井 清昭

審査委員主査　白井 清昭 助教授
審査委員　島津 明 教授
審査委員　東条 敏 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

110076 玉垣 隆幸

提出年月: 2004 年 2 月

Copyright © 2004 by Takayuki Tamagaki

概 要

単語の意味を決める語義曖昧性解消は、自然言語処理の中でも重要なタスクの一つである。本研究では、人間の文章理解を支援する読解支援システムでの使用を前提とした語義曖昧性解消のための分類器を作成する。読解支援システムでの使用が前提なので、より多くの単語を扱える再現率を重視した分類器が必要である。そのために、２つの異なる知識源を用いることにより、この問題の解決を試みた。一つ目の知識源は、注釈付きコーパスである。注釈付きコーパスとは、新聞記事などに人手で様々な付加情報を付け加えたテキストデータである。注釈付きコーパスから機械学習を行い、分類器を作成する。コーパスを使用した教師あり学習によってつくられた分類器の利点として、一般的に精度が良く、データ量が豊富であれば再現率も高いとされている。しかし、欠点もある。コーパス中に出現回数の少ない語義や文脈は学習に反映されづらいという、データの過疎性の問題がある。この欠点を克服するために、もう一つの異なる知識源 (国語辞典) を用いた分類器と組み合わせることにした。

注釈付きコーパスから機械学習を行うアルゴリズムとして、Support Vector Machine(SVM) を用いた。SVM は二値分類のアルゴリズムで、汎化性が強く、過学習を起こしにくいと言われている。学習を行う素性として以下のものを用意し、様々な素性に対して学習を行い、最もマッチした素性を実験的に求めた。

- 多義語の前後 n 語に含まれる自立語の基本型を抜き出す。n を可変にして、最適な文脈の大きさを調査した 。
- 多義語の直前、直後にある m 語の品詞情報と表記を抜き出す。m を可変にして、最適な m の大きさを調査した。

多義語の前後 n 語以内に現れる自立語の意味クラスを抜き出す。意味クラスはシソーラスの ID を用いた。意味クラスを用いる場合は、以下の２つの点で最適化を試みた。

- 一つは分類語彙表の桁数に関する最適化である。分類語彙表の ID を上位 3 桁から 7 桁まで変化させた
- もう一つは、一つの単語が複数の意味クラスをもつ場合の処理に関する最適化である。複数の ID が存在する場合は展開して素性に加える場合と単独の ID のみを加える場合を考慮した。

本研究では、コーパスから学習をして作成した分類器の他に、岩波国語辞典に記述されている情報を使用して 2 種類の分類器を作成した。

岩波国語辞典では、定義文中に用例が記述されていることがある。用例を用いた分類器は、入力文と語釈文中の用例の類似度を計算し、最も類似度の高い用例を持つ語義を選択する。類似度はシソーラスを使い求めた。一方、岩波国語辞典では、ある語義が出現する条件が文法情報として記述されていることがある。そこで、語義の文法情報を用いて語義曖昧性解消を行う分類器を作成した。この分類器は、候補となる全ての語義について、入力文がその語義の文法情報を満たすかどうかを調べる。そして、文法情報を満たす語義があれば、これを正しい語義として出力する。

さらに、SVM、用例、文法情報を用いたの 3 つの分類器を組み合わせる方法を提案した。最初に、共通のテストデータ (ヘルドアウトデータ) を用意し、それぞれの分類器単体の正解含有率を調べる。正解含有率は、出力した語義に正解が含まれる単語数の分類器によって語義が一つ以上出力された単語対する割合と定める。そして、ヘルドアウトデータにおける正解含有率の一番高い分類器の出力を最終的な出力として選択する。但し、SVM については単語毎に正解含有率を測定し、他の分類器の正解含有率との比較を行った。さらに、ヘルドアウトデータにおける頻度が 10 以下の単語については、正解含有率の信頼性が低いので、全単語の平均の正解含有率を SVM の正解含有率とした。

ヘルドアウトデータ、テストデータを用いて分類器の作成、評価を行った。SVM 分類器の作成・評価では、ベースライン精度 0.7877 に比べ、最高精度が 0.8059 と 1 %強しか上昇しなかった。そのときに用いた素性は、多義語の前後 7 語の読みと表記であった。また、シソーラスの意味クラスなど素性を加えると、かえって精度が落ちた。これは、過学習が起きたためと思われる。

一方、組み合わせの手法を用いた分類器は SVM 分類器と比べて精度は 8 %強、F 値は約 3 %落ちた。これに対し、再現率は 2 %強、適用率は 2 %近く上昇した。本研究の目的は、読解支援システムでの使用を前提とし、再現率を上げ、より多くの単語について語義の曖昧性を解消することにある。本結果から、この目的がある程度達成されたことが確認された。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　研究の背景と目的

単語の意味を決めるタスクは語義の多義性解消 (Word Sense Disambiguation:以下 WSD と略す) と呼ばれ、自然言語処理の中の重要なタスクの中のひとつである。

語義曖昧性解消の応用として、例えば次のような場合が考えられる。音声を認識して、意味を理解する知能やロボットなどを作ろうとしたとき、まず音声認識処理を行い表層的なテキストを得る。仮に「このはしわたるべからず」というテキストが得られたとする。次に形態素解析を行い、「この／はし／わたる／べから／ず」のように単語ごとに入力テキストを区切る。そして、この情報に意味を付加する過程へと送る。この過程において、2 種類の解釈が成り立つ。「はし＝橋」と解釈するか「はし＝端」と解釈するかによって、この文の意味が大きく変わる。

本研究では、このように複数の語義をもつ単語の語義を判別するシステム（分類器）を作成する。分類器の作成の手法としては、注釈付きコーパスを使用した方法が最もよく研究されている。注釈付きコーパスとは、新聞記事などに人手で様々な付加情報を付け加えたテキストデータである。例えば、国語辞典の語義が語義タグとして付与されているコーパスが存在する。このような語義タグが付与された多義語と周辺の文脈をてがかりに、機械学習によって語義曖昧性解消を行う分類器を作成する。コーパスを使用した方法は、一般的に精度が高く有用とされており、コーパスの量を増やすほど良い分類器ができるといわれている。しかし、語義タグ付きコーパスの作成はコストと時間がかかる。また、コーパスを使用する方法では、コーパス中に出現回数の少ない語義や文脈は学習に反映されづらいというデータの過疎性の問題がある。

本研究では、この問題を回避するために、コーパスから作成した分類器と、コーパスとは異なる言語資源から作成した分類器を別に作成する。異なる言語資源とは、語義の定義となる機械可読辞書内に記載されている情報である。辞書には用例や文法情報が語義別に記載されている場合があり、これらの情報を手がかりに、分類器を作成する。

さらに、これらの異なる言語資源から作成した分類器を組み合わせる手法を提案する [19]。コーパス中に出現しない単語は機械学習できない。しかし、辞書中の情報を使えば語義の曖昧性を解消できるときもある。このように、複数の分類器を組み合わせることによって語義曖昧性解消の再現率の向上を目指す。

1.2　本論文の構成

本論文の構成は以下の通りである。2 章では、語義曖昧解消全般の関連研究について述べる。3 章では、コーパスを使用した分類器の作成について述べる。4 章では、国語辞典を使用した分類器の作成について述べる。5 章では、3 章、4 章で作成した分類器を組み合わせる方法について述べる。6 章では、システムの評価実験結果を行い、結果の考察を行う。7 章では結論と今後の課題を述べる。

第2章　関連研究

本章では語義多義性解消の先行研究を紹介する。最後に、これらの研究と本研究の相違について述べる。

関連研究を紹介する前に、SENSEVAL について述べる。SENSEVAL は単語の多義性解消のコンテストである。1998 年の第一回 SENSEVAL-1[7] と、2001 年の第二回 SENSEVAL-2[18] が行われた。日本語タスクは 2001 年の第二回から行われ、辞書タスク [29] と翻訳タスク [10] の問題設定が設けられた。これから紹介する関連研究は、SENSEVAL に関する論文をが多い。

2.1　教師あり学習アルゴリズムによる WSD 分類器の作成

教師あり学習に基づいた方法は数多く発表されているが、本論文に特に関連が深い論文を紹介する。

村田らは、SENSEVAL-2 の日本語辞書タスクにおいて、いくつかの機械学習アルゴリズムと素性の組について実験を行った [27]。使用した素性は以下の通りである。

- 文字列素性
  - 多義語の文字列
  - 直前、直後の文字列 1〜3gram
- RWC 形態素素性
  - RWC コーパスの形態素情報
  - 解析する語の情報
    解析する単語の分類語彙表の 5 桁、分類語彙表の 3 桁、読み、表記、品詞
  - 直前、直後の単語の情報
    直前、直後の単語の分類語彙表の 5 桁、分類語彙表の 3 桁、読み、表記、品詞
- JUMAN 形態素素性
  コーパスを JUMAN で形態素解析をし、その形態素情報を用いる
  - 解析する語の情報

  - 直前、直後の単語の情報

- 構文素性
  コーパスを KNP で構文解析し、その結果を用いる。
  - 同一文節に体言があるかどうか
  - 解析する単語を含む文節内の係り受け先の文節内の自立語の情報
  - 解析する単語を含む文節内の係り受け元の文節内の自立語の情報

- 同一文内共起素性
  コーパスを JUMAN で形態素解析し、その形態素情報を用いる
  - 同一文中の単語の情報

- UDC 素性
  - RWC コーパスに記事ごとに付与している UDC コードの最初の 1 桁、2 桁、3 桁

学習アルゴリズムは以下の 3 つを適用した。

- シンプルベイズ法
- サポートベクトルマシーン (SVM)
- 決定リスト

また、上記のアルゴリズムでつくられた分類器をいくつか組み合わせる方法についても実験を行っている。この研究では、2 種類のシンプルベイズと 2 種類の SVM 分類器を組み合わせた手法が最も精度がよく、78.8 %であった。

2.2 教師無し学習アルゴリズムによる WSD 分類器の作成

正解タグ付きのコーパスの作成には、時間と費用がかかる。そこで、正解の無いプレーンテキストを使って語義曖昧性解消を行う研究もある。Yarowsky は少量のタグ付きコーパスを基にして、タグ無しコーパスから自動的に素性を追加する手法を提案した [17]。まず、

1. ある語義と共起しやすい単語がある
2. 同じ文章では同じ語義が出現する

という 2 つの性質を用いた。1 の性質から、共起語を素性とした決定リストを作成した。その分類器を用いてタグ無しテキスト上の多義語の語義を判別し、信頼度が高い場合はその単語を新たに訓練データに加えた。さらに 2 の性質から、新たに語義が決まった単語が

あるとき、同一記事中にある同じ単語に全て同一語義を与えることによって訓練データを増やした。そして、訓練データを追加するごとに決定リストの尤度を更新し、再学習を行という操作を繰り返し、決定リストを学習するための訓練データを獲得することに成功した。

この手法はCo-trainingの一種と見なせる。Co-trainingの概要は以下の通りである。まず、2つの独立した属性を用いた2つの分類器を作成する。次に、一方の分類器での語義判定結果を訓練データとして、他方の分類器の学習を行う。次は逆の操作を行う。この操作を繰り返し、2つの分類器の精錬を行う。Co-trainingは、2つの独立な素性集合を設定し、ラベル付きデータから2つの分類器を作成し、その分類器を用いてラベル無しデータラベルを与えることで学習データ量を増やす手法である。しかし一方では、2つの独立の素性を定義することへの困難性も指摘されている [24]。Yarrowskyの手法は、対象語の周辺に現れる語などといった決定リストの学習に用いる属性と、同一記事中の同一単語の語義は同じになりやすいという属性の2つを用いている点でCo-trainingの一種とみなせる。

また、EMアルゴリズムを使った方法も、新納によって報告されている [23]。未知のクラスのラベルcが与えられたときの素性fが共起する確率$P(f|c)$が最大になるように、未知のデータを使ってパラメータを決める方法である。その他にAdaBoostを使った手法も提案されている [28]。AdaBoostはBoostingの一種である。Boostingとは、精度の低い分類器と組み合わせて高い精度の分類器を構成する手法である。

2.3 コーパス以外の言語資源を使用したWSD分類器の作成

2.3.1 語釈文を用いる方法

Leskは辞書の語釈文を用いて語義を決める方法を提案した [11]。彼は、文脈と多義語の語釈文の単語が最も多く一致した語義を選択した。この方法では50 %から70 %の精度であったが、語釈文によっては全く一致を得られることなく、有用でないこともあった。また、この方法では計算量が多く、実験は数単語についてのみ行われた。

Cowieらは、Leskの方法を大規模に行う近似法を提案した [3]。語幹が同じ語を同一の単語とし、文中にある多義語について、それらの語釈文中の語の重複度が最大となる語義の組合わせを焼き鈍し法という最適化の手法を用いて近似的に求めた。この手法で、47 %の精度で多義性解消ができたと報告している。

2.3.2 機械可読辞書の様々な情報を用いた研究

Litkowskiは機械可読辞書に記載されている情報を使用する手法を提案し、SENSEVAL-2の語義曖昧性解消タスクで実験を行った [2]。著者は、まずNew Oxford Dictionary of Englinsh(NODE)の語義とWordNetの語義の対応付けを行った。SENSEVAL-2タスクの

語義は WordNet の語釈を用いているが、NODE と WordNet の語義を対応付けすることにより、NODE の熟語や文法情報を使用して WordNet の語義立てによる曖昧性解消が可能になる。また、NODE を語義立てとした語義曖昧性解消の評価も行っている。語義曖昧性解消に使用した NODE 中の情報を以下に挙げる。

- 最も頻繁に出現する語義
- 熟語
- 文型（例:他動詞かどうか）
- 意味的、構造的規則
- 格構造
- 特殊な形（大文字、時制、単複形）
- 選択制限
- 語釈文の一致度

実験の結果、精度は 29.3 %であった。

2.3.3 用例ベースの方法

黒橋らは、機械可読辞書 IPAL[22] から動詞の格フレームを語義ごとに抽出し、語義曖昧性解消のための格フレーム辞書を作成した。そして、シソーラスを用いて格要素の類似度を算出し、格ごとに重みを設定して重みつき足し算を行い、そのスコアが高い語義を選択した [9]。語義 s のスコア付け式 (2.1) に示す。

$$P(s) = \omega_s \sum_c SIM(n_c, \varepsilon_{s,c}) \tag{2.1}$$

c は格、n_c は入力文の格 c の格要素、$\varepsilon_{s,c}$ は格フレーム辞書の用例中の格 c の格要素である。ω_s は重みで、入力と格フレーム辞書で合致した必須格の数を考慮に入れ実験的に決めた。用例データベースは新聞記事から曖昧性を解消するべき動詞の格フレームの用例を抜き出し、それを IPAL の格フレームのいずれかに振り分けて作成した。

藤井らは、以下の 2 つの考えに基づいて、(2.1) 式を改良した [5]。

- 格によって動詞の曖昧性解消への貢献度は異なる。
  (例えば、主格よりも対格の格要素の方が多義性解消の強い手がかりとなる)
- 選択制限の適用範囲を考慮に入れる
  (「生物」のようなゆるい選択制限を満たすよりも、「子供」のような厳しい選択制限を満たす場合の方が、その格フレームを正解にすることが多い)

その結果、黒橋らより精度が向上したと報告している。

2.4 分類器の組み合わせ

一般に、一つの分類器よりも、複数の分類器を作成し組み合わせた方が良い結果が得られると言われている。本節では、過去に行われた分類器の様々な組合わせ方法について述べる。

2.4.1 教師つき学習によって作成した複数の分類器による組み合わせ

高村らは、独立成分分析や主成分分析の手法を用いて素性空間の再構築を行い、英語のタスクに対して語義曖昧性解消を行った [14]。これらの手法は、素性ベクトルの疎な部分を省略し、密な部分だけで素性ベクトルを構築し、学習を行う方法である。学習アルゴリズムに SVM を使い、異なる素性でいくつかの分類器を作成した。これらの分類器の結果による重みつき投票を行い、最終的な語義を選択した。重み w は分類器を作成する際に得られる VC_bound を利用して、$w = 1 - VC\_bound$ とした。VC_bound については 3 章で詳しく述べる。

Florian は 6 種類の語義曖昧性解消を行う分類器を作成し、その組み合わせ方を検証した [4]。６つの分類器はいずれも教師付き学習のアルゴリズムで作成され、語義と事後確率を返す。これらの分類器を組み合わせる 9 つの方法について実験を行った。

まず、分類器の事後確率を使い、重みつき足し算をする方法によりスコア付けを行う方法を検討した。この方法は重みを求めるために、訓練、評価コーパスの他に、重みを調整するためのコーパスが必要となる。

$$P(s|d) = \sum_{k=1}^{N} \lambda_k(d) \cdot P_k(s|d) \tag{2.2}$$

$P(s|d)$ はドキュメント d で語義 s が出現する確率で、 N は分類器の総数で $N = 6$ である。また、$P_k(s|d)$ は分類器の事後確率である。最適な $\lambda_k(d)$ を求めることが必要であり、以下のアルゴリズムを使用した。

- $\lambda_k(d) = 1/N$(N は分類器の個数)
- 最小自乗法
- EM 法
- Perfomance-based
  $\lambda_k(d) = P$(分類器 k が正解 $|d$)

また Voting ベースの組み合わせも試した。これは $P(s|d)$ を式 (2.3) のように定義した手法である。

$$P(s|d) = \sum_{k=1}^{N} \lambda_k(d) \cdot \delta(s, s_k(d)) \tag{2.3}$$

$\delta(s, s_d)$ はクロネッカーのデルタで、$s = s_d$ のとき 1、その他の場合は 0 である。Voting ベースでは語義の事後確率を必要としない。$\lambda_k(d)$ を求めるために、以下のアルゴリズムを使用した。

- 多数決
  $\lambda_k(d) = 1/N$
- Tag Pair
  事後確率を次のように近似する

$$P(s|d) \sum_{k=1}^{N} \delta(s, s_k(d)) + \sum_{j<i} \delta(s, s_{i,j}(d))$$

  $s_{i,j}$ は 2 つの異なる分類器 i, j が同時に同じ語義を返した場合の語義である
- EM 法
- Perfomance-based
  $\lambda_k(d) = P(\text{分類器 } k \text{ が正解 } |d)$

最後の 9 番目の分類器は、式 (2.2) の代わりに式 (2.4) を用いる。式 (2.4) では分類器の事後確率の順位 $rank_k(s|d)$ を使用した。

$$P(s|d) = \frac{\sum_{k=1}^{N} \lambda_k(d) \cdot rank_k(s|d)}{\sum_{\hat{s}} \sum_{k=1}^{N} \lambda_k(d) \cdot rank_k(\hat{s}|d)} \tag{2.4}$$

最も結果が良かったのは、9 個の分類器のうち上位 3 個で投票した分類器 (Stacking) であったと報告している。

Klein らは、教師付き学習のアルゴリズムによって語義曖昧性を解消する 23 の分類器を単語ごとに作成した [8]。この研究では、23 個の分類器の中から精度が良いを上位数個を選択し、多数決、重み付き投票、エントロピー最大モデルの方法を使用して、これらの出力語義を入力とし、語義を一つ選択する分類器を作成した。さらに、その 3 つの中で最も精度が良いものを最終結果として採用した。

詳細なアルゴリズムは次の通りである。

- データをテスト用と調整用に分割する
- 第一段階の各分類器は、調整用のデータを使いすべての単語に対する平均精度を出す
- 第一段階の各分類器は、調整用のデータを使い個々の単語に対する精度を出す
- 単語ごとに分類器を作成し、精度が良い順にランク付けを行う。同点の場合は平均精度を使う

- 上位にランク付けされた分類器の中から決められた個数を選択し、第二段階の分類器を作成する
- 第二段階の分類器の中から最も精度の高い語義を出力する

この手順を図 2.1 に図示する。最終段階の分類器の精度は 63.9 %であった。

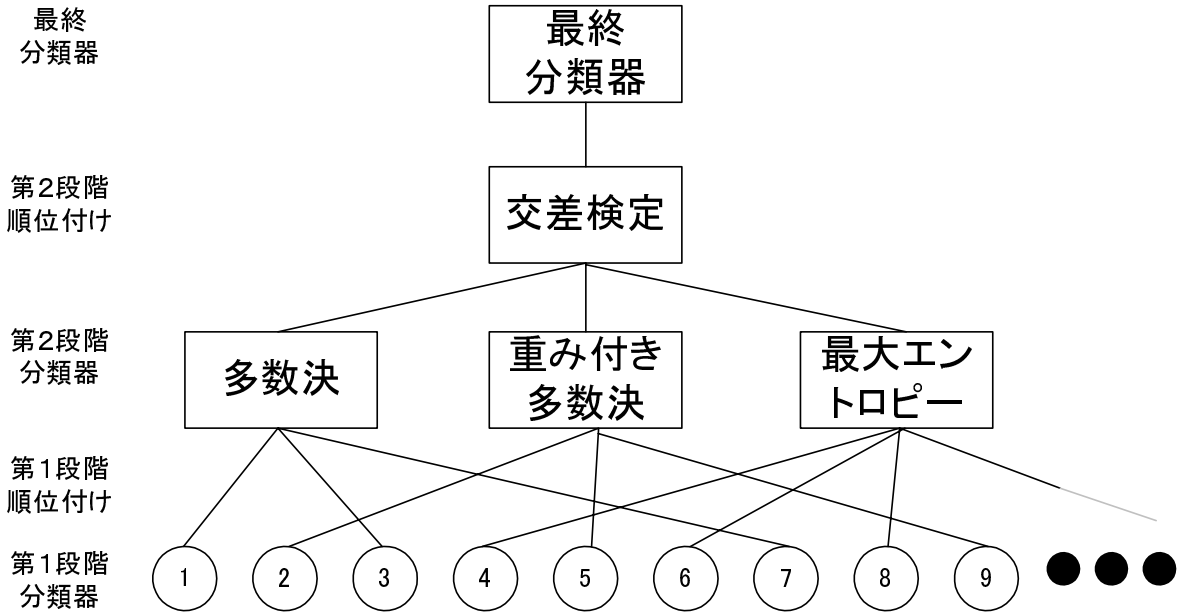

図 2.1: Klein の方法

福本は複数の素性について語義ごとに分類器を作成し、その中から最適な分類器を選択する方法を用いた [30]。語義の数 q、素性の数 r の組に対し、$q \times r$ 個の分類器を SVM を使い実装した。各分類器は語義が妥当か否かを返す。疑似テストを行い、その語義で最も多く正解を返した素性を最終的な分類器とする。例えば、図 2.2 の場合では、s_1 の語義は

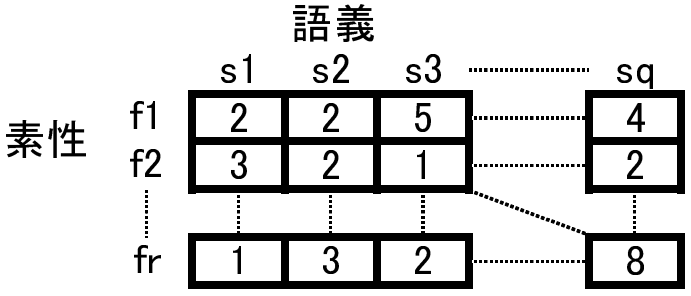

図 2.2: 福本の方法

f_2 の素性を利用し、s_2 では f_3、s_q では f_r とし、これらの素性と語義の組について分類器を作成した。*interest* と *line* の 2 単語で評価を行い、90 %以上の精度を得た。しかし、この方法では単語ごとに語義と素性の数だけ学習を行わなければならず、すべての単語について学習を行うのは現実的に不可能である。

Pedersen は素性の異なる Naive Bayes 法による分類器を 81 個作成した [12]。81 個の分類器を表 2.3 のように 9 つに分類した。表 2.3 中の一つのマトリックスは 9 つの成分からなる。全体のマトリックスは縦、横が narrow,midium,wide の 3×3 のマトリックスからなる。彼の分類器は多義語の左右の文脈の位置する単語を素性として用いるが、マトリックスの縦軸は多義語より右側の文脈の大きさを、横軸は左側の文脈の大きさを表す。マトリックス中に記述されている数字は、それぞれの分類器の精度である。たとえば、縦=25、横=1 のとき 0.81 というのは、文脈の右側 25 語、左側 1 語を素性として学習した分類器の精度は 0.81 であるという意味である。一つのマトリックスから最も精度の高い分類器を選択 (図中の斜体の精度) し、9 つのマトリックスから選ばれた 9 個の分類器により投票を行う。*interest* と *line* の 2 単語で評価を行い、それぞれ 88 %、89 %の精度だった。

	50	.74	.80	.82	.83	.83	.83	.82	.80	.81
wide	25	.73	.80	.82	.83	.83	.83	.81	.80	.80
	10	.75	.82	*.84*	*.84*	.84	.84	*.82*	.81	.81
	5	.73	.83	.85	.86	.85	.85	.83	.81	.81
medium	4	.72	.83	.85	.85	.84	.84	.83	.81	.80
	3	.70	.84	*.86*	*.86*	.86	.85	*.83*	.81	.80
	2	.66	.83	.85	.86	*.86*	.84	*.83*	.80	.80
narrow	1	.63	.82	*.85*	.85	.86	.85	.82	.81	.80
	0	.53	.72	.77	.78	.79	.77	.77	.76	.75
		0	1	2	3	4	5	10	25	50
		narrow			medium			wide		

図 2.3: Pederson の方法

2.4.2 コーパスと用例とそれ以外の知識源を用いた分類器の組み合わせ

Agirre らは機械可読辞書である WordNet から得られる情報から作成した分類器と、コーパスベースの分類器を組み合わせる方法を提案した [1]。学習アルゴリズムは決定リストを用い、訓練やテストは SENSEVAL-1[7] のデータを用いた。SENSEVAL-1 では多義語に WordNet の語義がふられており、WordNet には同義語の情報や、上位下位関係が記述されている。表 2.1 に、この研究で使用した情報を記載した。また、これらの情報を用いて作成した分類器と決定リストによる分類器をを組み合わせた。組合わせの方法として、重み付き足し算を用いた。ある分類器が出力する信頼度を、その分類器が出力する最大の信頼度で割ることにより正規化をし、その値を重みとした。最終的に精度、再現率、被覆率が向上したと報告している。

表 2.1: Agirre ら [1] で使用した辞書の情報

分類器	説明
複合語	多義語が特定の複合語の一部か否か
見出し語の順番	WordNet は出現頻度順に語義が並んでいる
Topic Domain	WordNet の同義語に付与された意味タグを用いる
単語の一致度	Lesk の方法
共起関係	辞書から得られる多義語の共起語と入力文の共起語との一致度を重みつき足し算を行う
共起ベクトル	共起する単語をベクトル表現にし内積の値を求める
概念密度	WordNet において入力文の周辺語を下位語として多く含む語義を選択
決定木	コーパスを使い機械学習をおこなう

2.5 先行研究と本研究との関連

今まで述べた先行研究と本研究の関連について述べる。

まず、最初に 2.3.1 項で使われた、語釈文の単語の一致数によって語義曖昧性解消を行う手法を岩波国語辞典の語釈文を用いて試した。しかし、この方法はほとんど機能しなかった。一致が見られた単語は「こと」、「もの」などの抽象名詞や辞書特有の言い回しによる単語が多く、内容語の一致はほとんど見られなかった。したがって、複数の語義の候補に対する語釈文に対する語釈文中の単語の一致数にほとんど差は見られなかった。よって、本研究では採用しなかった。

また、2.2 節のタグ無しテキストを用いる方法も本研究では用いない。これらの手法は適用率を向上させるために有効な手段であるが、タグ無しテキスト使用した際の、各種のパラメーターの調整やデータの質の確保といった問題はコストと時間がかかるためである。これに対し本研究では、複数の知識源を用いることで適用率を向上させるアプローチをとる。

本研究では、教師あり学習アルゴリズムを使用した分類器と国語辞典の情報から分類器を作成し、これらを組み合わせることにより最終の語義を出力する。本章で述べた先行研究と本研究の相違を述べる。

2.1 節で述べられた先行研究のうち、学習アルゴリズムとして SVM を、素性として RWC の形態素情報の素性などを用いる。本研究では、Klein ら [8] と同様に単語ごとに分類器を作成した。また、学習に用いる素性集合を変化させ、分類器の正解率を基準とした最適な素性の調査を行う。

2.3.2 項の手法と同様に、機械可読辞書から得られる情報を利用し、語義曖昧性解消を行う手法を提案する。具体的には、岩波国語辞典の用例と文法情報を用いた。まず、語釈

文中の用例から、格フレーム抽出し、用例辞書を作成した。そして用例中の格要素と入力文の格要素と類似度を算出し、最大の類似度を持つ用例を含む語義を選択した。これは 2.3.3 項の用例ベースの手法と同じであるが、あらかじめ用例データベースを作成する必要がなく、辞書中の用例をそのまま用いる点が異なる。類似度を計測する方法として (2.1) 式を用いた。但し、重みはヒューリスティックスを用いて独自に設定した。詳しくは、4 章で述べる。

文法情報を用いる分類器は先行研究では行われていない。

また分類器の組合わせの方法として、Klein ら [8] に近い手法を用いる。Klein らはコーパスから単語ごとに学習を行った分類器を、3 種類の組み合わせの方法を用いて出力を行った。その際、コーパスの一部を調整用データとして用いた。本研究では SVM 分類器は単語ごとに学習を行った。そして、調整用データを用いて単語ごとに正解含有率を算出した。他の 2 つの分類器も、調整用データから分類器全体の正解含有率を算出した。テストでは最も正解含有率がよい分類器を選択した。

第3章　教師付き学習アルゴリズムによる分類器の作成

本章では、コーパスを使用した分類器について述べる。機械学習に使用するアルゴリズムとしてSupport Vector Machine(以下SVM) を使った。3.1 節では、SVM のアルゴリズムについて、3.3 節ではSVM の機械学習で使用する素性について説明する。なお、SVM を使用した分類器を **SVM 分類器** と呼ぶ。

3.1　Support Vector Machine

Suppor Vector Machine は Vapnik によって提案された 2 値分類を行うための分類アルゴリズムである [16]。本節では、2 値に分類可能で、線型な分離平面が存在する場合について述べる。訓練事例を $\{\mathbf{x}_i, y_i\}, i = 1, \cdots, l, y_i \in \{-1, 1\}, \mathbf{x}_i \in \mathbf{R}^d$ と書く。この事例に対し、2 値に分離可能な平面 $\mathbf{w} \cdot \mathbf{x} + b = 0$ が存在する。その平面と原点との距離は $\frac{|b|}{||\mathbf{w}||}$ である。いま、$d_+(d_-)$ を分離平面から最も近い訓練事例の距離とする。このマージンを最大化する制約条件は式 (3.1)、(3.2) のように表せる。

$$x_i \cdot \mathbf{w} + b \geq +1 \quad for \quad y_i = +1 \tag{3.1}$$

$$x_i \cdot \mathbf{w} + b \leq -1 \quad for \quad y_i = -1 \tag{3.2}$$

まとめると

$$y_i(x_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i \tag{3.3}$$

となる。分離平面に平行で、最短の正例の側の訓練事例上を通る平面 $H_1 : x_i \cdot \mathbf{w} + b = 1$ と原点との距離は $\frac{|1-b|}{||\mathbf{w}||}$ である。同様に、 負例側の平面 $H_2 : x_i \cdot \mathbf{w} + b = -1$ と原点との距離は $\frac{|-1-b|}{||\mathbf{w}||}$ である。よって $d_+ = d_- = 1/||\mathbf{w}||$ である。平面間のマージンは $2/||\mathbf{w}||$ で、このマージンを最大化するように最適化を行う。この条件は、言い換えると、制約条件式 (3.3) のもとで $\frac{1}{2}||\mathbf{w}||^2$ を最小化する最適化問題になる。図 3.1 にこの問題を図式化した。

この問題を、ラグランジュの未定乗数法を用いて解く。制約条件を考慮した目的関数を L_P とし、未定係数を α_i と書くと L_P は次式で表せる。

$$L_P \equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \tag{3.4}$$

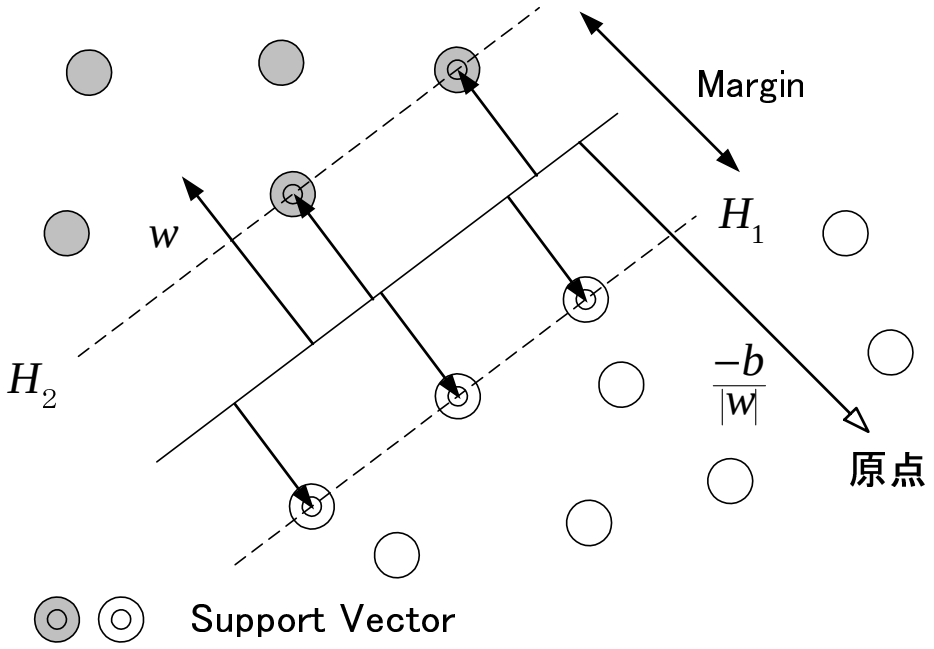

図 3.1: SVM 概要

また、式 (3.4) の双対問題より制約条件 $\alpha_i > 0$ が導出される。さて、L_P が極値をもつためには、$\frac{\partial L_P}{\partial \mathbf{w}} = 0$、$\frac{\partial L_P}{\partial b} = 0$ が必要で、この条件から

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{3.5}$$

$$\sum_i \alpha_i y_i = 0 \tag{3.6}$$

となる。式 (3.5)、式 (3.6) を式 (3.4) に適用すると、L_P と同値の双対問題 L_D は

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{3.7}$$

となる。α_i はサポートベクトルと呼ばれ、訓練事例が H_1, H_2 上に存在するとき $\alpha_i > 0$ となり、それ以外は 0 になる。SVM は、分離平面上に存在する訓練事例のみを考えれば良く、そのため過学習を起こしにくいアルゴリズムと言われている。

Vapnic Chervonenkis(VC) bound　SVM は、ある学習モデルの訓練のしにくさの上限値を学習と同時に求めることができる。今あるベクトル $\mathbf{x}_i, i = 1, \cdots, l$ に対して、y_i が「真」というラベル付けを考える。また、線型独立で単調増加な未知の確率分布 $P(\mathbf{x}, y)$ が与えられているとする。訓練事例に対するテストデータの誤り率の期待値を $R(\alpha)$ とすると

$$R(\alpha) = \int \frac{1}{2} |y - f(\mathbf{x}, \alpha)| dP(\mathbf{x}, y) \tag{3.8}$$

である。$R(\alpha)$ を risk と呼んでいる。実際に観測可能な risk を empirical risk とよび、その期待値 $R_{emp}(\alpha)$ は

$$R_{emp}(\alpha) = \frac{1}{2l}\sum_{i}^{1}|y_i - f(\mathbf{x}, \alpha)| \tag{3.9}$$

となる。Vapnic は、ある変数 η、$0 < \eta < 1$ に対して式 (3.10) が成り立つとした。

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{(\frac{h(\log(2l/h)+1) - log(\eta/4))}{l})} \tag{3.10}$$

h は VC Dimension である。式 (3.10) は、$R(\alpha)$ の値は決定することはできないが、上限値は知ることができる、という意味である。式 (3.10) の左辺の第 2 項を VC confidence と呼び、左辺を VC bound と呼ぶ。

3.2 RWC コーパス

本研究では、SVM を学習するためのコーパスとして RWC コーパス [6, 26] を用いる。RWC コーパスは、毎日新聞の 1994 年の 3000 記事に語義 ID を付与したテキストコーパスである。語義タグ付けの対象となる新聞記事は、計算機で形態素解析された後、人手で形態素情報を修正したコーパスである。またコーパスの自立語には、岩波国語辞典 [21] の語義が付与されている。また、語義の他に以下の情報が付与されている。

- 形態素情報
  読み、表記、基本型、分かち書きの情報
- UDC コード

表 5.1 にコーパスの一部分を掲載する。

3.3 素性選択

本実験では、学習に用いる素性を変化させ、語義曖昧性解消に最適な素性を実験的に調べる。RWC コーパスの多義語周辺の文脈から以下に述べるグローバル素性、ローカル素性、意味クラス素性を抽出し、学習を行った。以下はコーパスから抜き出した多義語「違う」を含む文である。この文を例に、素性の抽出について説明する。

> 今／まで／の／刑事／ドラマ／と／は／ひと味／**違い**／、／主人公／の／刑事／の／心理／描写／や／彼／を／取り巻く／日常／を／丁寧／に／描い／て／いく／。

- グローバル素性
  多義語の前後 n 語に含まれる自立語の基本型の表記を抜き出す。n を可変にして、最適な文脈の大きさを調査した 。

```
<article id="00000810" udc="(046) 631.158 331.584">

<mor pos="1" rd="ノウチ">農地</mor>
<mor pos="268" rd="アリ" bfm="ある" sense="1380-0-1-1-0*">あり</mor>
<mor pos="454" rd="マス" bfm="ます">ます</mor>
<mor pos="490" rd="  ">  </mor>
<mor pos="1" rd="ツキ" sense="34012-0-0-2-0*">月</mor>
<mor pos="15" rd="１５万">１５万</mor>
<mor pos="30" rd="エン">円</mor>
<mor pos="13" rd="シキュウ">支給</mor>
<mor pos="468" rd="、">、</mor>
<mor pos="1" rd="シンチク">新築</mor>
<mor pos="1" rd="ジュウタク">住宅</mor>
<mor pos="24" rd="ツキ">付き</mor>
<mor pos="468" rd="－－">－－</mor>
<mor pos="7" rd="シマネ">島根</mor>
<mor pos="468" rd="・">・</mor>
<mor pos="7" rd="ヨコタ">横田</mor>
<mor pos="24" rd="チョウ">町</mor>
<mor pos="419" rd="ガ">が</mor>
<mor pos="1" rd="ケンシュウセイ">研修生</mor>
<mor pos="419" rd="ヲ">を</mor>
<mor pos="13" rd="コウボ">公募</mor>
                    .
                    .
                    .
</article>
```

図 3.2: RWC コーパス

グローバル素性の例:多義語の前後 5 単語の自立語

cl:刑事 cl:ドラマ cl:ひと味 cr:主人公 cr:刑事

「cl:」は素性のラベルで多義語より左側、「cr:」は右側の文脈に位置することを示す。

- ローカル素性
  多義語の直前、直後にある m 語の品詞情報と表記を抜き出す。m を可変にして、最適な m の大きさを調査した。

  ローカル素性の例:多義語の前後 3 単語の位置情報と形態素情報と表記

  ph1:ひと味 ph2:は ph3:と pp1:1 pp2:423 pp3:419 pp4:1 pp5:1 sh1:、 sh2:主人公 sh3:の sp1:468 sp2:1 sp3:419

  「ph1:」「sh2:」のラベルはそれぞれ多義語の前と後に現れる表記を表す。数字は多義語から何語離れているかを示している。「pp2:423」「sp3:419」のラベルは多義語の前と後に現れる単語の品詞と位置情報を表す。「:」の右側の数字は RWC コーパスの品詞コードである。

- 意味クラス素性
  まず、多義語の前後 n 語以内に現れる自立語の意味クラスを抜き出す。意味クラスは分類語彙表 [20] を用いた。意味クラスを用いる場合は、以下の 2 つの点で最適化を試みた。

  – 分類語彙表の桁数に関する最適化。分類語彙表の ID を上位 3 桁から 7 桁まで変化させた
  – 一つの単語が複数の意味クラスをもつ場合の処理に関する最適化。一つの単語に対して複数の ID が存在する場合は、それらすべての素性に加える場合と、そのような単語については正しいシソーラスの ID は不明なので、意味クラス素性は追加せず、シソーラスの ID を一つだけ持つたんごについてのみ意味クラス素性を用いる場合の 2 つの手法を試した。

```
bgh:11613+10+1 bgh:31650+12+1 (日常 11613+10+1,31650+12+1)
bgh:12000+3+1 (彼 12000+3+1)
bgh:13103+2+1 (描写 13103+2+1)
```

図 3.3: シソーラスの ID が 7 桁で、複数のシソーラスの ID を全て用いる場合

上記の例では「日常」のシソーラスの意味クラスは複数ある。図 3.3 の例では、多義性を考慮し「日常」のすべての意味クラスを 7 桁の ID としてで素性に追加した。図 3.4

```
bgh:12000（彼 12000+3+1）
bgh:13103（描写 13103+2+1）
```

図 3.4: シソーラスの ID が 5 桁で、複数のシソーラスの ID を持つ単語は意味クラス素性を追加しない場合の例

の例では、「日常」のシソーラスの意味クラスは複数あるので、素性に加えない。一方、「彼」や「描写」の意味クラスは一つであるので、この 5 桁の意味クラスを素性に追加しいてる。

第4章　国語辞典を用いた分類器の作成

本研究では、コーパスから学習をして作成した分類器の他に、岩波国語辞典に記述されている情報を使用して2種類の分類器を作成した。なお、4.2節で作成する分類器を**用例分類器**、4.3節で作成する分類器を**文法情報分類器**と呼ぶ。

4.1　岩波国語辞典

岩波国語辞典 [21] の概要を表4.1に示す。

表 4.1: 岩波国語辞典の概要

	動詞	名詞	形容詞	副詞	全体
見出し語数	11,474	50,745	668	1,050	60,321
語義数	18,856	66,943	1,256	1,665	85,870
多義である見出し語数	2,585	9,385	196	258	12,360
多義である語義数	9,967	25,562	784	873	37,909

また、岩波国語辞典のデータ例を図4.1に示す。語釈文は形態素解析され、各形態素にはRWCコーパスと同じ品詞コードが付与されている。また、用例は特別なタグ＜EX＞で囲まれている。ここでは語釈文中の用例や文法情報の記述を語義曖昧性解消の手がかりとして使用する。

4.2　辞書の用例を用いた分類器

本節では、国語辞典の語釈文中に出現する用例を用いて多義性解消を行う分類器について述べる。この分類器は、語義曖昧性解消の対象となる語の品詞によって手法が異なる。

4.2.1　動詞の場合

岩波国語辞典では、定義文中に用例が記述されていることがある。動詞「愛する」の語釈文を図4.2に示す。図4.2において「子を―」、「国を―」、「酒を―」等、括弧で囲まれ

```
<entry id="37" fukugou_id="0" mds="あいえん" knz="愛煙家" pos="名">
<sense id="37-0-0-0-0">
<mor pos="1" rd="タバコ">タバコ</mor>
<mor pos="419" rd="が">が</mor>
<mor pos="14" rd="スキ">好き</mor>
<mor pos="502" rd="ナ">な</mor>
<mor pos="16" rd="コト">こと</mor>
<mor pos="468" rd="。">。</mor>
<mor pos="468" rd="「">「</mor>
&lt;EX&gt;
<mor pos="1" rd="アイエン">アイエン</mor>
&lt;/EX&gt;
<mor pos="24" rd="カ">家</mor>
<mor pos="468" rd="」">」</mor>
</sense>
</entry>
```

図 4.1: 岩波国語辞典

た部分が用例である。

【愛する】
それに対し愛をそそぐ。
(1) かわいがり、**いつくしむ**。「子を—」。心から大切に**思う**。「 国を—」
(2) 異性を恋い**慕う**。
(3) 物事を強く**好む**。「酒を—」

図 4.2: 「愛する」の語釈文

用例を用いた分類器は、入力文と語釈文の用例の類似度を計算し、最も類似度の高い用例を持つ語義を選択する。例えば、入力文が「彼は娘を愛している」のとき、図 4.2 中の 3 つの用例との類似度を計算する。その結果、「子を愛する」との類似度が高ければ、入力文「愛する」の語義として (1) を選択する。

次に、入力文と用例の類似度を計算する方法を説明する。類似度は、同じ格に立つ名詞の意味的類似度から求める。まず、各語義毎に用例から格 c の格要素となる名詞の集合 NE_c を抽出する。図 4.2 からは次のような格要素が抽出される。

【愛する】(1) $NE_{ヲ}$={ 子, 国 }
【愛する】(3) $NE_{ヲ}$={ 酒 }

岩波国語辞典では全ての語義に用例があるわけでなく、また用例から得られる格要素の数も十分ではない。そこで、用例から得られる格要素の数を増やすため、語釈文中の最後の動詞を上位語とみなし、上位語と元の語とでは似ている名詞が格要素として現れると仮定して、上位語の語釈文から格 c の格要素の集合 $NE_{c'}$ を抽出する。例えば、「愛する」の (2) の語義の上位語を「慕う」とし、「慕う」の語釈文の用例（図 4.3）から格要素を抽出する。これを図 4.4 に図示する。

【慕う】
(1) 愛着の心をいだいてあとを追う。「母を―って三千里」。恋しく思って (心の中で) 追い求める。「故国を―」。「彼女がひそかに―青年」。
(2) 徳や学問・技量を敬い、これにならおうとする。「徳を―って集まる」。

図 4.3: 「慕う」の語釈文

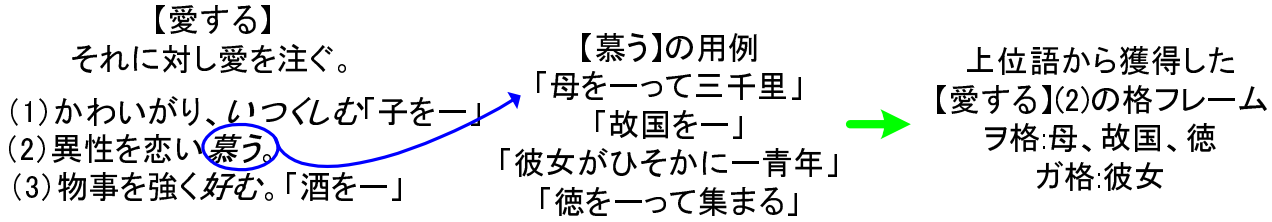

図 4.4: 上位語の語釈文からの格フレームの獲得例

【愛する】(2) $NE_{ヲ'} = \{$ 母, 故国, 徳 $\}$
$NE_{ガ'} = \{$ 彼女 $\}$

他の語義についても同様に格要素を抽出する。また、上位語に複数の語義があるときは、適切な語義を 1 つ選択してその語義の用例から格要素を抽出するべきであるが、本研究では上位語の全ての語義の用例から格要素を抽出することにした。

入力文 s と用例文 e の類似度 $Sim(s,e)$ は式 (4.1) のように定義した。

$$Sim(s,e) = \sum_c w_c s_c(ns_c, NE_c) \tag{4.1}$$

$$s_c(ns_c, NE_c) = max_{ne_c \in NE_c} s(ns_c, ne_c) \tag{4.2}$$

$$s(w_i, w_j) = \frac{2 \times d_k}{d_i + d_j} \tag{4.3}$$

式 (4.1) において、$s_c(ns_c, NE_c)$ は格 c に対する入力文の格要素 ns_c と用例（上位語の用例を含む）から得られた格要素の集合 NE_c の類似度である。また、w_c はその重みである。w_c は経験的に定めた。特に、上位語の用例から抽出された格要素への重み $w_{c'}$ は、元の単

表 4.2: 実験で使用した重み

重み	ガ格	ヲ格	ニ格	その他
w_c	6	5	5	4
$w_{c'}$	3	2	2	1

語の用例から抽出された格要素の重み w_c よりも低くなるようにした。表 4.2 に具体的な重みを示す。

式 (4.2) において、$s_c(ns_c, ne_c)$ は二単語間の類似度で、式 (4.3) で定義される。式 (4.3) における d_i,d_j は単語 w_i,w_j のシソーラスにおける深さ、d_k は w_i と w_j の共通上位ノードのシソーラスにおける深さを表す。シソーラスは日本語語彙体系 [20] を使用した。

岩波国語辞典から得られた格要素の数を表 4.3 に示す。

表 4.3: 格要素が得られた語義数と格要素数

	見出しのみ	上位語使用
多義である動詞の語義数	9,967	
格要素を獲得できた語義数	3,371	4,227
獲得できた格要素数	5,645	30,148

多義である動詞のうち、用例から格要素を獲得できた語義の割合は 33.8%、上位語の用例も使用したときには 42.4%である。したがって、この分類器の適用率は決して高いとは言えない。しかし、他の分類器と組み合わせることにより、システム全体の適用率の向上が期待できる。

4.2.2 名詞の場合

名詞の語釈文に出現する用例も、以下のパターンに用例が該当すれば、その用例を抽出し用例データベースに加えた。以下、語義曖昧性解消の対象となる名詞を下線で表す。

- A の B
  - A(名詞)+B(名詞) (N_{amg}の N)
    例:「烏 の行水」「烏 のお灸」
  - B(名詞)+A(名詞)(N の N_{amg})
    例:「感激の 至り」「光栄の 至り」
- 複合名詞

- 名詞+名詞(NN_{amg})
  例:「選挙 運動」「社会 運動」
- 名詞+名詞 ($N_{amg}N$)
  例:「一日 市長」「一日 乗車券」

- 名詞+格助詞+動詞 (N_{amg}格助詞 V)
  例:「音 を殺して」「音 を立てて歩く」

次に、これら用例と入力文中との類似度を計算し、最も高い類似度をもつ語義を選択する。

$$s(ns, NE) = max_{ne \in NE}\ s(ns, ne) \tag{4.4}$$

$$s(w_i, w_j) = \frac{2 \times d_k}{d_i + d_j} \tag{4.5}$$

用例が「A の B」、「B の A」、「複合名詞」の場合の類似度は式 (4.4) とする。式 (4.4) において、$s(ns, NE)$ は入力文の多義語と共起する名詞と用例から得られた共起名詞の集合 NE の類似度である。式 (4.5) は二単語間の類似度で、式 (4.3) と同じである。

また、名詞+格助詞+動詞の場合は式 (4.6)、(4.7) を用いた。

$$Sim(s, e) = \sum_c w_c s_c(ns_c, NE_c) \tag{4.6}$$

$$s_c(ns_c, NE_c) = max_{ne_c \in NE_c} s(ns_c, ne_c) \tag{4.7}$$

ただし、ns、NE は格要素でなく、名詞の係り先となる動詞に置き換えた。また類似度 $s(ns_c, ne_c)$ の計算は式 (4.5) を使用した。抽出した用例の種類と数を表 4.4 に示す。

表 4.4: 岩波国語辞典から抽出した用例の数

N_{amg}の N	N の N_{amg}	NN_{amg}	$N_{amg}N$	N_{amg}格助詞 V	$ADJ_{amg}N$
659	1034	1532	1039	2068	311

4.2.3 形容詞の場合

多義語が形容詞の場合、以下のパターンにマッチする例を抜き出した。

- 形容詞+名詞 ($ADJ_{amg}N$)
  例「新しく 入社した人」「新しい 学問」「新しい 思想」

類似度は式 (4.4),(4.5) を使用した。抽出した用例の数を表 4.4 の「$ADJ_{amg}N$」に示す。

4.3　辞書の文法情報を用いた分類器

岩波国語辞典では、ある語義が出現する条件が文法情報として記述されていることがある。例を図 4.5 に示す。「さらに」の (2) の語義には、《あとに打消しを伴って》という文法情報の後に語義の定義が記述されている。したがって、入力文が「後悔しているようすなど**さらに**ない」のとき、「さらに」の後に打ち消しの表現があるので、この語義は (2) であると推測できる。このように、岩波国語辞典に記述された文法情報は、語義曖昧性解消の有効な手がかりとなる。

【さらに】
(1) その上に。重ねて。「―懇願する」「―は増援部隊も加わった」ますます。もっと。「―上達する」。
(2) 《あとに打消しを伴って》少しも。いっこうに。さらさら。「―反省の色がない」。

図 4.5: 「さらに」の語釈文（抜粋）

そこで、文法情報を用いて語義曖昧性解消を行う分類器を作成した。この分類器は、候補となる全ての語義について、入力文がその語義の文法情報を満たすかどうかを調べる。そして、文法情報を満たす語義があれば、これを正しい語義として出力する。また、複数の語義が文法情報を満たすときには、その全ての語義を出力する。文法情報を満たす語義がなければ、出力なしとする。

岩波国語辞典では、語義の文法情報は図 4.5 のように二重角括弧で囲まれて記述されていることが多い。そこで、岩波国語辞典おける多義の単語の語釈文から、二重角括弧で囲まれた記述を語義毎に取り出し、入力文がその条件を満たしているかどうかを判定するプログラムを作成した。プログラムとして実装した語義の文法情報の例を以下に挙げる。

- 単語の活用形に関する条件

  【快い】《主に連用形で》
  【眩む】《受身の形で》

- 前後の単語の表記、品詞、活用形に関する条件

  【くれる】《動詞連用形＋「て」を受けて》
  【あがり】《名詞のあとに付く》
  【さっぱり】《多く「と」を伴って》

- 語義が出現する定型表現

  【いっぺん】《「いっぺんに」の形で》
  【否や】《「…や否や」「…と否や」の形で》

- 後に打ち消しの表現を伴うか否か

  【一切】《下に打消しを伴って、副詞的に》
  【てんで】《俗に、打消しを伴わずに》

- 文中での位置に関する条件

  【頂戴】《文末で》

文法情報を取り出すことのできた語義の数は 973 であった。このうち、582 の語義について、条件を満たすか否かを判断するプログラムを実装した。岩波国語辞典における多義語の語義の総数は 37,908 なので、文法情報を判定するプログラムを実装できた語義の割合はわずか 1.5 %である。しかし、文法情報を取り出すことのできた語義は頻出単語の語義が多く、実際にこの分類器を用いるときの適用率はもっと大きくなると予想される。また、文法情報を満たすときには高い精度で正しい語義を選択できると期待される。

文法情報を取り出すことができたが、プログラムとして実装しなかった条件には以下のようなものがある。

1. 二重角括弧で囲まれた記述が語義の文法情報を表していない場合
   (例) 【あたる】《忌み言葉の用法》
2. 単語の品詞に関する条件
   (例) 【一等】《副詞的に》
3. 前後の単語の意味に関する条件
   (例) 【薄】《色・味をさす語にかぶせて》
   【現在】《月日・時を表す語に付けて》

2 については、形態素解析ツールが出力する品詞と文法情報として書かれた品詞が一致していない場合があるので、本研究では用いなかった。例えば、上の例の “一等” の語義には「これが一等いい」という用例文があるが、これを JUMAN や茶筌で形態素解析すると、“一等” の品詞は名詞となる。ただし、「副詞的に」という条件を曖昧性解消に用いることもできる。例えば構文解析を行い、“一等” が用言に係ることがわかれば、この単語が副詞的に用いられていると判断できる。また、3 についても、シソーラスなどを利用して周辺の単語の意味を調べることにより、条件を満たすかどうかを自動的に判定することができる。これらの実現については今後の課題としたい。

第5章　分類器の組み合わせ

本章では、3,4 章で作成した分類器を組み合わせる方法について説明する。まず、コーパスの使い方について述べる。実験に使用する RWC コーパス 3,000 記事のうち、2400 記事を SVM の訓練データ、300 記事をヘルドアウト (パラメータ調整用) データ、300 記事をテストデータとした。SVM による分類器については、訓練データにおける頻度が 10 以上 1585 個の単語について分類器を学習し、それ以外の単語については語義を出力しないことにした。図 5.1 にコーパスの使用法を図示した。

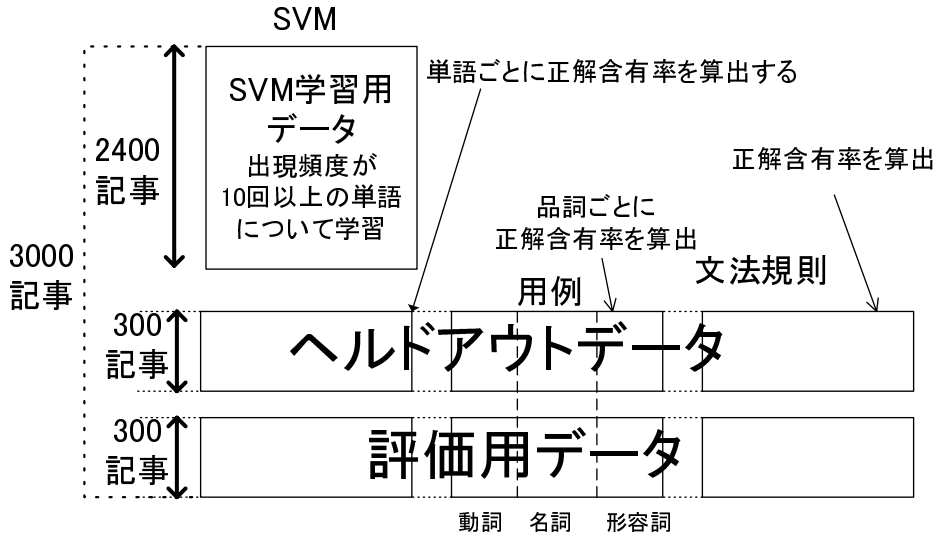

図 5.1: RWC コーパスの使い方

次に、共通のヘルドアウトデータにおいて、それぞれの分類器単体の正解含有率 (式 (5.1)) を算出する。

$$\text{正解含有率} = \frac{\text{出力した語義に正解が含まれる単語数}}{\text{分類器が語義を一つ以上出力した単語数}} \tag{5.1}$$

そして、語義を出力した分類器のうち、ヘルドアウトデータにおける正解含有率の一番高い分類器の出力を最終的な出力として選択する。本研究は、正しい語義のみを表示させたり、表示する語義の数を絞り込む読解支援システムでの使用を前提としている。そのため、分類器が出力する語義の中に正解が含まれていることが重要であると考える。した

がって、たとえ正しくない語義を出力したとしても、正しい語義をより多く出力する分類器を優先的に用いる。精度ではなく、正解含有率の比較を行うのはそのためである。

用例分類器では品詞ごとに正解含有率を設定した。SVM については単語毎に正解含有率を測定し、他の分類器の正解含有率との比較を行った。さらに、ヘルドアウトデータにおける頻度が Oh 以下の単語については、正解含有率の信頼性が低いので、全単語の平均の正解含有率を SVM の正解含有率とした。6 章の実験では Oh=10 とした。

この処理を図 5.2 に図示する。図中 SVM、用例、文法情報の各分類器の正解含有率をそれぞれ Ah_{SVM}、Ah_{EXAM}、Ah_{RULE} とした。また、SVM で学習した単語 ω の正解含有率を Ah_{ω}、そのヘルドアウトデータにおける頻度を $O(\omega)$ とする。Acc_{SVM}、Acc_{EXAM}、Acc_{RULE} は最終的に語義を出力する分類器を選択するために比較を行う評価値を表す。

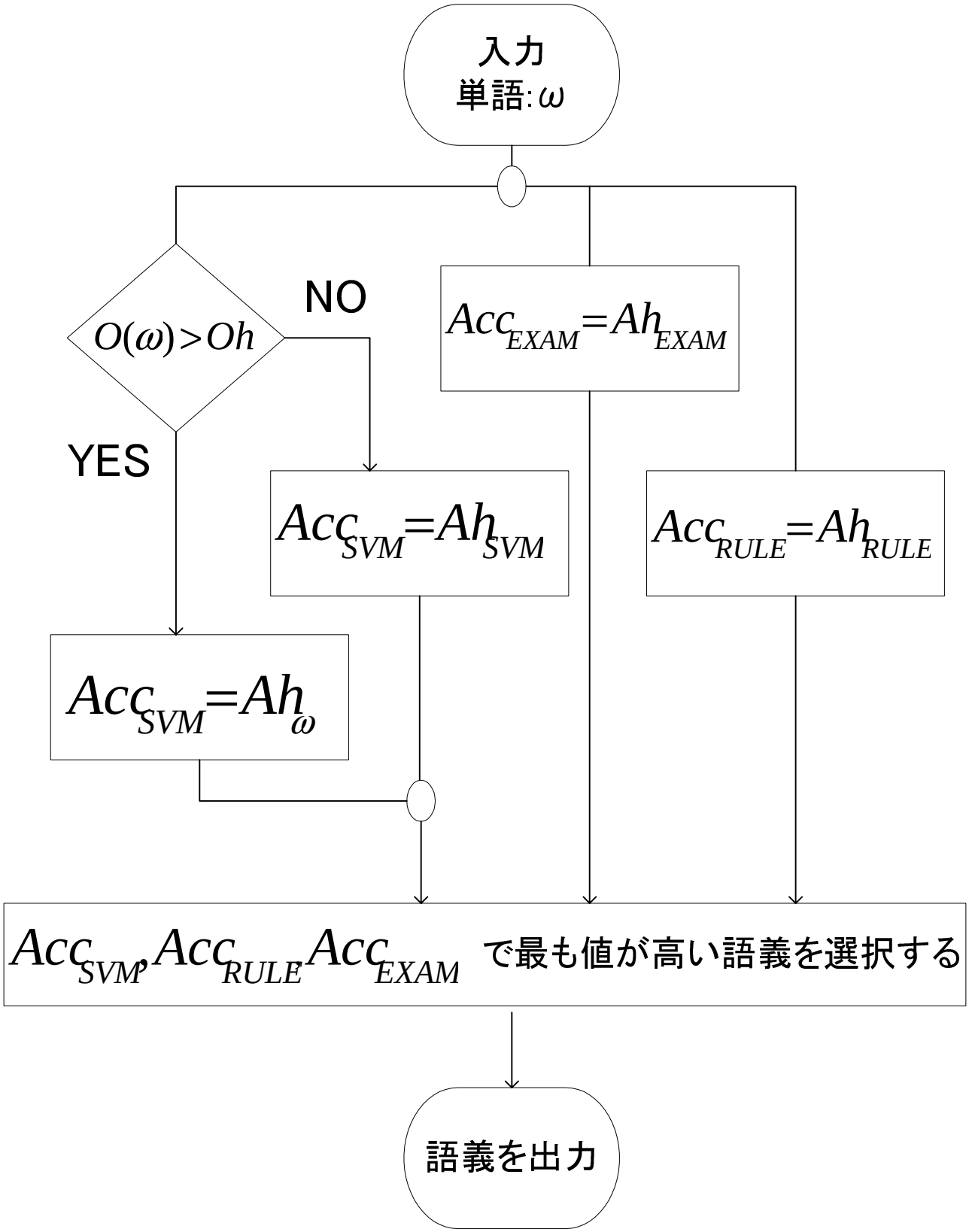

図 5.2: 分類器の組み合わせ

第6章　評価実験

提案手法を評価する実験を行った。まず、それぞれの分類器を実装し、ヘルドアウトデータで正解含有率を算出した。次にテストデータを使用して多義性解消の精度、再現率、F値などを調べ、提案手法の評価を行った。

比較の対象として、ベースラインの分類器を作成した。訓練データ中で頻度10以上の単語について、読み、表記、品詞が同じであれば同一の単語とみなし、同一の単語において出現回数が最も多い語義を出力とした。ベースライン分類器の精度は0.7877であった。

6.1　SVM分類器の作成・最適な素性の選択

SVMの学習にはLIBSVMを用いた[1]。ν–SVM [13]によって学習を行い、カーネルは線形カーネル、$\nu = 0.0001$とした[2]。SVMは二値分類器であるのに対し、本研究における語義曖昧性解消問題は多値問題である。そこで、pairwise法を用いてSVMを多値問題に適用した。

3.3節で述べた複数の素性を使い、分類器の作成を行い、ヘルドアウトデータで正解含有率を求めた。但し、SVM分類器は一つしか語義を出力しないので、正解含有率は精度と同じである。次の素性の組み合わせに対して、素性集合を変化させ、分類器を作成し評価を行った。

- ローカル素性のみ
  複数の文脈の大きさに対して正解含有率を求めた。結果を表6.1に記す。

- グローバル素性のみ
  複数の文脈の大きさに対して正解含有率を求めた。結果を表6.2に記す。

- ローカル素性+グローバル素性+意味クラス素性
  ローカル素性、グローバル素性の文脈の大きさ、及び意味クラス素性のオプションを変化させた。結果を、表6.3、6.4に記す。
  表6.3、6.4中、「BGH桁数」は分類語彙表の桁数、「多義」は単語の意味クラスが複数あるときにその全てを素性として加えた(Y)か、加えずに意味クラスが一意に決

[1] http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm/

[2] 多項式カーネルの次数2,3,4についても実験を行ったが、精度がほぼ半減したので、多項式カーネルを使用をやめた

まるときのみに意味クラスを加えた (N) かを表す。また、空欄は意味クラスを使用しないことを表す。(詳しくは、3.3 節を参照)

表中の正解含有率は、単語ごとに算出した値の平均である。ボールド体で記述されている値は、同じ表の中で最も高い値を示している。

表 6.1: ローカル素性

文脈の大きさ (語)	3	5	7	9	12
正解含有率	0.6596	0.7870	**0.8059**	0.8045	0.7816

表 6.2: グローバル素性

文脈の大きさ (語)	3	5	7	10	15	20	25	30
正解含有率	0.6204	0.7568	0.7706	0.7708	0.7750	0.7741	**0.7817**	0.7785

以上の結果から、次のことが明らかになった。

- ローカル素性では、前後 7 語を素性として加えたときに最も正解含有率が良く、実験を行った中でも最もよい結果になった。また、3 語程度では文脈の情報量が少なく、10 語以上では過学習が起きていると思われる。
- グローバル素性では、文脈の大きさが 7 以上になると、正解含有率の上昇は見られなかった。
- 素性を組み合わせた場合では、正解含有率はほぼ 1 %以内の変化しかみられない。また、意味クラス素性を学習素性として追加することによるパフォーマンスの向上は得られなかった。逆に、ローカル素性のみ場合と比べるとわずかながら精度が下がっている。これは、素性を過剰に追加することの弊害と思われる。
- ベースライン精度 0.7877 に比べ、最高精度が 0.8059 と 1 %強しか上昇しなかった。今回の実験では、原因を追求することができなかった。

素性を様々に変化させた中では、ローカル素性のみを利用し、文脈の大きさを 7 としたときの正解含有率 0.8059 がよかった。以後、この素性を用いたものを分類器を SVM 分類器として使用する。

表 6.3: ローカル素性+グローバル素性+意味クラス素性 (その１)

グローバル素性	ローカル素性	BGH 意味素性		正解含有率
		BGH 桁数	多義	
10	3	3	N	0.7946
10	3	3	Y	0.7807
10	3	5	N	0.7909
10	3	5	Y	0.7864
10	3	7	N	0.7927
10	3	7	Y	0.7952
10	3			**0.7974**
10	5	3	N	0.7769
10	5	3	Y	0.7889
10	5	5	N	0.7922
10	5	5	Y	0.7882
10	5	7	N	0.7894
10	5	7	Y	0.7934
10	5			0.7916
10	7	3	N	0.7907
10	7	3	Y	0.7860
10	7	5	N	0.7887
10	7	5	Y	0.7930
10	7	7	N	0.7922
10	7	7	Y	0.7944
10	7			0.7894
15	3	3	N	0.7907
15	3	3	Y	0.7884
15	3	5	N	0.7944
15	3	5	Y	0.7898
15	3	7	N	0.7841
15	3	7	Y	0.7893
15	3			0.7951
15	5	3	N	0.7853
15	5	3	Y	0.7840
15	5	5	N	0.7940
15	5	5	Y	0.7911
15	5	7	N	0.7890
15	5	7	Y	0.7881
15	5			0.7881
15	7	3	N	0.7927
15	7	3	Y	0.7849
15	7	5	N	0.7884
15	7	5	Y	0.7864
15	7	7	N	0.7906
15	7	7	Y	0.7930
15	7			0.7859

表 6.4: ローカル素性+グローバル素性+意味クラス素性 (その 2)

グローバル素性	ローカル素性	BGH 意味素性		正解含有率
		BGH 桁数	多義	
20	3	3	N	0.7934
20	3	3	Y	0.7930
20	3	5	N	0.7887
20	3	5	Y	0.7935
20	3	7	N	0.7885
20	3	7	Y	0.7857
20	3			0.7954
20	5	3	N	0.7889
20	5	3	Y	0.7897
20	5	5	N	0.7922
20	5	5	Y	0.7934
20	5	7	N	0.7919
20	5	7	Y	0.7880
20	5			0.7894
20	7	3	N	0.7875
20	7	3	Y	0.7841
20	7	5	N	0.7850
20	7	5	Y	0.7936
20	7	7	N	0.7927
20	7	7	Y	0.7897
20	7			0.7875
25	3	3	N	0.7898
25	3	3	Y	0.7932
25	3	5	N	0.7929
25	3	5	Y	0.7913
25	3	7	N	0.7919
25	3	7	Y	0.7916
25	3			0.7924
25	5	3	N	0.7882
25	5	3	Y	0.7940
25	5	5	N	0.7921
25	5	5	Y	0.7895
25	5	7	N	0.7931
25	5	7	Y	0.7909
25	5			0.7901
25	7	3	N	0.7905
25	7	3	Y	0.7899
25	7	5	N	0.7917
25	7	5	Y	0.7887
25	7	7	N	0.7912
25	7	7	Y	0.7907
25	7			0.7903

6.2 国語辞典を使用した分類器のヘルドアウトデータによる評価

用例分類器、文法情報分類器もヘルドアウトデータによる評価を行った。表 6.5,6.6 は、用例分類器のヘルドアウトデータにおける結果である。3 段目の「入力」はヘルドアウトデータにおいて用例とマッチした単語の数である。適用数は分類器が答えを出した単語数である。名詞で複数の用例にマッチした場合は、類似度が高い用例による結果を採用した。このような用例の重複を含めた名詞全体のテスト結果を「名詞全体」に示した。

表 6.5: 用例分類器のヘルドアウトテストの結果 1

	名詞					名詞全体
用例	N_{amg}の N	N の N_{amg}	NN_{amg}	$N_{amg}N$	N_{amg}格助詞 V	
入力	772	1135	1,657	1,975	1,059	5325
適用数	117	251	179	107	261	881
正解含有数	77	140	99	49	169	518

表 6.6: 用例分類器のヘルドアウトテストの結果 2

	動詞	形容詞
用例	N 格助詞 V_{amg}	$ADJ_{amg}N$
入力	3807	178
適用数	1689	68
正解含有数	846	45

次に、式 (5.1) に基づいて正解含有率を算出した。この場合、用例分類器の正解含有率は表 6.5,6.6 の「正解含有数」／「適用数」で求められる。用例分類器は品詞別に正解含有率を求めた。正解含有率を表 6.7 に記す。

表 6.7: 正解含有率

分類器	用例分類器			文法情報分類器
品詞	名詞	動詞	形容詞	
正解含有率	0.58	0.49	0.65	0.81

本研究では、動詞の語釈文の最後の動詞を自動的に上位語とみなすというヒューリスティクスを用いた。しかしながら、必ずしも正しくない用例を獲得したことも少なくな

かった。上位語の語釈文からより正確な用例を得るために、もとの語と上位語の意味的類似を求め、類似度が高い場合には上位語から獲得された用例に対する重みを高くする方法が考えられる。単語間の類似度として例えば、単語間の意味的関連度をコーパスにから抽出する研究 [15] を用いることもできる。これにより、上位語からの類似例文の獲得は、本実験で提案した手法よりも有効に働く可能性がある。今後の課題としたい。

6.3 結果

ヘルドアウトデータでの評価の結果に基づいて、３つの分類器を組み合わせる実験を行った。表 6.8 は、テストデータにおける評価実験の結果である。「混合モデル」は組み合わせアルゴリズムを用いて作成した分類器である。また、比較のため各分類器単体の評価も併記した。評価の基準として精度、再現率、F 値、適用率を求めた。F 値は、

$$\frac{2 \times \text{精度} \times \text{再現率}}{\text{精度} + \text{再現率}}$$

とした。適用率は、

$$\frac{\text{システムが出力を返すことができた数}}{\text{全入力}}$$

とした。本研究は、読解支援システムでの使用を目的として作成されたので、この指標を用いた。各項目で最も数値がよいものをボールド体にした。

表 6.9 は、混合モデル分類器において選択された分類器の数である。

表 6.8: テストデータにおける各手法の評価

	精度	再現率	F 値	適用率
混合モデル	0.7172	**0.7341**	0.7256	**0.9070**
用例	0.4498	0.0898	0.0955	0.1497
文法情報	0.5396	0.1875	0.2783	0.2272
SVM	**0.8016**	0.7107	**0.7534**	0.8867

表 6.9: 混合モデルで選択された分類器の数

用例	文法情報	SVM	回答なし
442	847	11,883	1,350

混合モデル分類器ははSVM分類器と比べて精度は8 %強、F値は約3 %落ちた。一方で再現率は2 %強、適用率は2 %近く上昇した。これは、精度の低い用例分類器、文法情報分類器を組み合わせることによって混合モデルの精度は落ちたが、複数の分類器を同時に使用することによって再現率が向上したと考えられる。本研究の目的は、読解支援システムでの使用を前提とし、複数の知識源を用いた分類器を組み合わせることにより、より多くの単語について語義の曖昧性を解消することにある。言い換えれば再現率の向上を第一の目的としている。本結果から、この目的がある程度達成されたことが確認された。

さらに、再現率を向上させる方法としてベースライン分類器と組み合わせる方法が考えられる。ベースライン分類器の精度は0.7836、再現率は0.766、F値は0.7447、適用率は0.9774であった。これは、混合モデルよりも性能がよい。したがって、本研究で提案した分類器とベースライン分類器と組み合わせることにより更なる性能の向上が期待できる。

第7章　おわりに

本論文では、より多くの単語を扱うことのできる語義曖昧性解消手法を提案した。3種類の分類器を作成し、これらを組み合わせることを試みた。3種類の分類器のうち、SVM分類器では様々な素性集合の中から最適な素性集合を調査した。他2種類は岩波国語辞典の用例と文法情報を用いて分類器を実装した。

最後に、今後の課題を３つ挙げる。

1. コーパスを使用した分類器の最適な素性と学習アルゴリズム選択
   今回の実験では、学習アルゴリズムにSVMを用いたところ、比較的シンプルな素性の方が結果が良かった。一方、村田ら [27] は、本論文と素性は異なるが、SVMよりもシンプルベイズ法のほうが精度がよい、という報告をしている。多義性解消というタスクがSVMに適しているか、その他の学習アルゴリズムの方が適しているかをさらに調べる必要がある。
2. データの過疎化の問題
   本論文では、多くの単語を扱うために複数の異なる言語資源を利用した。しかし答えを返すことができない単語も一割ほど残っている。2章でも言及したが、利用可能な膨大なドキュメントから語義タグの付与された文を獲得し、訓練データに加える方法が提案されている。答えを返す単語を増やすためには、教師なし学習を行う手法を併用するべきかもしれない。しかし、この方法は、学習がうまくいかないと精度を落とす原因になるので、扱いが難しい。
3. 語の区切りの問題
   実際にシステムを実装する際は形態素解析が必要である。しかし、形態素解析システムの辞書と岩波国語辞典では、辞書間の見出しの表記や区切り、品詞体系が異なるため辞書が引けない事態も考えられる。したがって、形態素解析の出力と岩波国語辞典の見出しの対応付けを行う必要がある [25]。

謝辞

本研究を進めるにあたり，終始熱心な御指導を賜わりました白井清昭助教授に心から感謝致します．さらに数多くの御教授を頂きました島津明教授に厚く御礼申し上げます．山田寛康助手ならびに自然言語処理学講座の皆様には，貴重な御意見、討論をして頂きました事を感謝致します．

関連図書

[1] E. Agirre, G. Rigau, L. Padró, and J. Atserias. Combining supervised and unsupervised lexical knowledge methods for word sense disambiguation. *Computers and the Humanities*, Vol. 34, No. 1,2, pp. 103–108, 2000.

[2] Kenneth C.Litkowski. Sense Information for Disambiguation: Confluence of Supervised and Unsupervised Methods. *Proceeding of the SIGLEX/SENSEAVAL Workshop on Word Sense Disambiguation ,Association for Computational Linguistics(ACL)*, pp. 47–53, 2002.

[3] Jim Cowie, Joe Guthrie, and Louise Guthrie. Lexical disambiguation using simulated annealing. *Proceeding on the International Conference on Computational Linguistics*, pp. 359–365, 1992.

[4] Radu Florian and David Yarowsky. Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2002*, pp. 25–32, 2002.

[5] Atsusi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. To what extent does case contribute to verb sense disambiguation? In *COLING-96: The 16th International Conference on Computational Linguistics -Copenhagen-, Vol.1*, pp. 59–64, 1996.

[6] Koiti Hasida, Hitoshi Ishihara, Takenobu Tokunaga, Minako Hshimoto, Shiho Ogino, Wakako Kashino, Jun Toyoura, and Hironobu Takahashi. The rwc text databases. *Proceeding on the first International Conference on Language Resources and Evaluation*, pp. 457–462, 1998.

[7] Adam Kilgarriff. SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *LREC 1998: First International Conference on Language Resources and Evaluation, vol.1*, pp. 581–585, 1998.

[8] Dan Klein, Kristina Toutanova, H.Tolga Ilhan, Sepandar D.Kamvar, and Christopher D.Manning. Combining Heterogeneous Classifiers for Word-Sense Disambiguation.

Proceeding of the SIGLEX/SENSEAVAL Workshop on Word Sense Disambiguation ,Association for Computational Linguistics(ACL), pp. 74–80, 2002.

[9] Sadao Kurohashi and Makoto Nagao. A method of case structure analysis for japanese sentences based on examples in case frame dictionary. *IEICE Transactions on Information and Systems*, Vol. E77-D, No. 2, 1994.

[10] Sadao Kurohashi and Kiyotaka Uchimoto. SENSEVAL-2 Japanese Translation Task. 自然言語処理, Vol.10 No.3, pp. 25–37, 2003.

[11] Michale Lesk. Automated sense disambiguation using machine-readable dictionaryies:How to tell a pine cone from an ice cream cone. In *SIGDOC Connference*, pp. 24–26, 1986.

[12] Ted Pedersen. A simple approach to building ensembles of naive baysian classifiers for word sense disambiguation. *Proceeding on the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 63–69, 2000.

[13] Bernhard Schölkopf, Alex J. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. *Neural Computation*, Vol. 12, pp. 1083–1121, 2000.

[14] Hiroya Takamura, Hiroyasu Yamada, Taku Kudoh, Kaoru Yamamoto, and Yuji Matsumoto. Ensembling based on feature space restructuring with application to wsd. In *NLPRS2001: Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, pp. 41–48, 2001.

[15] Kentaro Torisawa. An unsupervised learning method for associative relationships between verb phrases. In *COLING 2002: The 19th International Conference on Computational Linguistics Vol.2*, pp. 1009–1015, 2002.

[16] Vladimir N. Vapnik. *Statistical Learning Theory*. A Wiley-Interscience Publication, 1998.

[17] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. *Proceeding on the Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, 1995.

[18] David Yarowsky. In *Proceedings of SENSEVAL-2*, 2002.

[19] 玉垣隆幸, 白井清昭. 読解支援システムのための語義曖昧性解消に関する研究. 言語処理学会第 9 回年次大会発表論文集, pp. 481–484, 2003.

[20] 池原悟, 宮崎正弘, 白井論, 横尾昭男, 中岩浩巳, 小倉健太郎, 大山芳史, 林良彦. 日本語語彙体系－全五巻－. 1997.

[21] 西尾実, 岩淵悦太郎, 水谷静夫. 岩波国語辞典 第五版. 1994.

[22] 情報処理振興事業協会. 計算機用日本語基本動詞辞書 IPAL(Basic Verbs), 1987.

[23] 新納浩幸. EM アルゴリズムを用いた教師なし学習の日本語翻訳タスクへの適応. 自然言語処理, Vol.10 No.3, pp. 61–73, 2003.

[24] 新納浩幸 (茨城大). 素性間の共起性を検査する Co-training による語義判別規則の学習. 情報処理学会研究報告 (自然言語処理研究会) 2001-NL-145、2001-FI-64, pp. 29–36, 2001.

[25] 森田勝, 白井清昭. 意味辞書を利用するための形態素変換規則の自動獲得. 言語処理学会第 9 回年次大会発表論文集, pp. 149–152, 2003.

[26] 白井清昭, 柏野和佳子, 橋本美奈子, 徳永健伸, 有田英一, 井佐原均, 萩野紫穂, 小船隆一, 高橋裕信, 長尾確, 橋田浩一, 村田真樹. 岩波国語辞典を利用した語義タグ付きテキストデータベースの作成. 情報処理学会自然言語処理研究会, pp. 2000(9):117–122, 2001.

[27] 村田真樹, 内山将夫, 内元清貴, 馬青, 伊佐原均. SENSEVAL2J 辞書タスクでの CRL の取り組み-日本語単語の多義性解消における種々の機械学習手法と素性の比較. 自然言語処理, Vol.10 No.3, pp. 115–133, 2003.

[28] 平井有三 (筑波大学) 中野桂吾. Adaboost を用いた語義の曖昧性解消. 言語処理学会第 8 回年次大会発表論文集, pp. 659–662, 2002.

[29] 白井清昭. SENSEVAL-2 日本辞書タスク. 自然言語処理, Vol.10 No.3, pp. 3–24, 2003.

[30] 福本文代. 語義の曖昧性解消のための最適な属性選択. 情報処理学会論文誌、Vol.43, No,1, pp. 20–33, 2002.