修 士 論 文

辞書定義文中の上位概念を用いた頑健な語義曖昧性解消

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

小川 千隼

2005年3月

修 士 論 文

辞書定義文中の上位概念を用いた頑健な語義曖昧性解消

指導教官　白井 清昭 助教授

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　鳥澤 健太郎 助教授

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

310024 小川 千隼

提出年月: 2005 年 2 月

概 要

語義曖昧性解消 (Word Sense Disambiguation,WSD) は文中に現れる単語の意味 (語義) を決める処理である．現在，語義曖昧性解消の手法として，語義タグ付きコーパスを利用した教師ありの機械学習による手法が主流であるが，訓練データ量を必ずしも十分に確保できないというデータの過疎性の問題がある．

このような問題に対処する手法として，語義タグのないコーパスを用いる教師なし学習を行う手法も提案されているが，本研究では 辞書定義文から語義の上位概念を抽出し，抽出した上位概念を反映した確率モデルを学習することにより，低頻度語の語義曖昧性解消の正解率を向上させる．例えば,「筆者」の 1 つの語義の辞書定義文,「その文章・書画を書いた人」から「人」という上位概念を抽出する．この例のように「人」という上位概念を持つ単語は「筆者」以外にもコーパスに存在する．このため，語義と文脈の共起情報を用いるよりも上位概念と文脈の共起情報を利用することで，語義自体はコーパスにあまり現れない単語でも上手く学習が出来る可能性がある．低頻度語用のモデルとして，上位概念と文脈の共起性を反映した Naive Bayes モデルを用いる．モデルに使う素性は，対象語の前後の表記，品詞，係り受け関係にある単語の基本形，同一文中にある自立語の基本形など，WSD に一般的に用いられるものを用いた．

本研究に近い研究を行った八木は，EDR 概念辞書を用いて上位概念を抽出した．ところが，EDR 概念辞書は機械処理に特化しているため，辞書定義文が単純でわかりづらい．例えば，EDR 概念辞書の「犬」の定義文は「犬という動物」である．これに対し，岩波国語辞典における「犬」の定義文は「古くから人間が家畜として飼い親しむ，いぬ科のけだもの」であり，犬に関してより多くの情報を得られる．定義文の品質が重要視されるアプリケーションにおいては，辞書定義文を理解しやすい一般の国語辞典を用いる方が望ましい．本研究では，より人にとって有益な表現の多い一般の国語辞典，具体的には岩波国語辞典を用いる．

次に，辞書定義文から語義の上位概念を抽出する手法について述べる．一般に，辞書定義文の末尾にある単語がその語義の上位概念を表していることが多い．したがって，原則として，辞書定義文の末尾の単語を上位概念として抽出する．しかし，末尾の単語では上位概念としてふさわしくない場合もある．例えば,「拝借」の定義文の「借りることをへりくだって言う語」というような「N を V て言う語」で終わる辞書定義文から末尾の「語」を上位概念とするのは適切ではないので，N の部分を取り出すパターンを適用して上位概念「借りること」を取り出す．このような上位概念抽出パターンを 116 個人手で作成し，岩波国語辞典の辞書定義文から上位概念を抽出した．そして，岩波国語辞典の全語義のうち，97.25%の上位概念の抽出に成功した．

また，岩波国語辞典は EDR 概念辞書と異なり，1 つの語義に対して複数の辞書定義文が存在する場合がある．個々の定義文から上位概念を取り出すとすると，1 つの語義に対して複数の上位概念が抽出される．一方，上位概念と文脈の共起性を反映した Naive Bayes

モデルでは語義の上位概念は 1 つであることを仮定している．そのため，抽出した複数の上位概念から最適なものを選択する必要がある．具体的には，辞書定義文の第 2 文以降を第 1 文の上位概念や文頭または文末のキーワードを手がかりに 3 つのタイプに分類し，そのタイプにしたがって上位概念を選択する．上記のプロセスのタイプ分類に用いるキーワードは全部で 38 種類ある．1 つの語義に対して複数の辞書定義文が存在している語義の辞書定義文をランダムに 200 語義取り出し，その第 2 文以降の分類タイプが適切なものであるかを人手で確認したところ，全体の 93.5%に相当する 187 語義の辞書定義文の分類タイプが適切であった．したがって，定義文の分類の精度は十分高いと言える．

本研究では，最終的に高頻度語のための教師あり学習モデルの SVM による分類器と，低頻度語のための上位概念を用いた分類器の 2 つを組み合わせた．組み合わせる手法は以下の通りである．

- 単語ごとの訓練データにおける出現頻度により分類器を選択
- 単語ごとの調整用データにおける正解含有率により分類器を選択
- スタッキングによる手法により分類器を選択

  スタッキングの手法を用いて，SVM による分類器と上位概念を用いた Naive Bayes モデル分類器を 1 次分類器とし，それらの出力を素性とする 2 次分類器を学習し，語義曖昧性解消を行う．2 次分類器では学習アルゴリズムとして SVM を用いる．また，1 次分類器の作り方や 2 次分類器で用いる素性について，3 つの異なるスタッキング手法を提案した．

また，単語と文脈の共起情報と，上位概念の共起情報を同時に利用する新たな分類器を作成し，分類器を組み合せた場合との比較も行った．

最後に提案手法を評価する実験を行った．単体の分類器同士の SVM，NB，BL(ベースラインモデル) の比較や，混合モデルである SVM+BL (SVM とベースラインモデルの組み合わせ)，SVM+NB (提案手法)，2 つの共起情報を同時に反映した NB モデルの比較を行った．この結果，本研究の提案手法である NB は SVM には及ばなかったものの，BL モデルと比べて F 値で 2 %以上の上昇がみられた．また，SVM と NB の混合モデルの中で最も F 値の高いモデルは交差検定を用いたスタッキングで，全ての混合モデルの中で最も高い F 値が得られた．

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

文中に現れる単語の意味 (語義) を決める処理を語義曖昧性解消 (Word Sense Disambiguation,WSD) という．例えば,「のむ」という単語について,「薬をのむ」と「要求をのむ」の２つの文章があったとき，我々は普通に「のむ」の意味がそれぞれ違うことを理解できる．語義曖昧性解消とは，このように「のむ」の意味がどちらであるかということを機械で自動的に判別させる処理のことである．語義曖昧性解消は読解支援システムや機械翻訳，情報検索など様々な分野に応用可能な，自然言語処理の中でも重要なタスクの一つであり，様々な手法で行われてきているが，現在でも正しい語義選択を高い精度で行うことは困難である．

語義曖昧性解消を行う手法として，現在語義タグ付きコーパスを利用した教師ありの機械学習による手法が主流である．これらの手法は，単語周辺の文脈を手がかりにして語義曖昧性解消を行う分類器を学習する．語義タグ付きコーパスとは，あらかじめその文章中での単語の正しい意味をコーパスと呼ばれる電子テキストに付与したデータである．

語義タグ付きコーパスは人手で作成する必要がありコストがかかるという問題がある．また，この手法ではコーパス中に出現回数の少ない語義や文脈については語義判定のモデルの学習が難しいというデータの過疎性の問題がある．

また，人間の文章理解を支援する読解支援システムなどのような実用的なアプリケーションで語義曖昧性解消を行う際には，より多くの単語を扱えること，すなわち再現率と適用率の向上が必要である．そのため，読解支援システムで使用することを考えた場合には，教師あり学習による手法をそのまま用いることは適切ではない．

このような問題に対処するために，本研究では，低頻度語の語義曖昧性解消モデルの学習に有効である，辞書定義文から得られる上位概念を利用する手法を用いて，頑健な語義曖昧性解消を実現することを目的とする．ここで言う頑健性とは，多くの単語に対して正しい語義を選択できるという意味であり，実用的な応用を考えたときに重要である．頑健性を向上させるために，教師なし学習を行う手法 [1][2][28] も考えられるが，本研究ではコーパス以外の知識源を併用する手法 [4][30] として，辞書定義文を利用する手法をとる．

本研究における低頻度語用の語義曖昧性解消モデルの概要は以下の通りである．辞書定義文から抽出した上位概念と周辺語の共起関係を反映した分類器を学習して語義曖昧性解

消を行う．例えば,「具象」の 1 つの辞書定義文,「物が実際にそなえている形」から「形」という上位概念を抽出する．この例のように「具象」という単語自体よりも「形」という上位概念の単語の方がコーパスにより多く出現しやすいため，上位概念と文脈の共起情報を利用することで，語義自体はコーパスにあまり現れない単語でも上手く学習が出来る可能性がある．

EDR 概念辞書 [20] を用いて上位概念を抽出した先行研究があるが [29]，それには EDR 概念辞書は機械処理に特化しているため，辞書定義文が単純でわかりづらいという問題がある．例えば，EDR 概念辞書の「犬」の定義文は「犬という動物」である．これに対し，岩波国語辞典 [21] における「犬」の定義文は「古くから人間が家畜として飼い親しむ，いぬ科のけだもの」であり，犬に関してより多くの情報を得られる．人の文書理解を支援する読解支援システムなどを構築する場合には，辞書定義文を理解しやすい一般の国語辞典を用いる方が望ましい．そこで本研究では，より人にとって有益な表現の多い岩波国語辞典を用いる．ただし，岩波国語辞典の辞書定義文は EDR 概念辞書よりも多様であるため，辞書定義文から語義の上位概念を抽出することは難しくなる．

本研究では，このような上位概念を用いた分類器と高頻度語の語義曖昧性解消に有効な分類器を組み合せることにより，より多くの単語に対して適用可能な語義曖昧性解消を行う．そして，分類器の組み合せる方法について検討し，最適な組み合わせ方法を見つけることを目的とする．また，単語と文脈の共起情報と，上位概念の共起情報を同時に利用する新たな分類器を作成し，分類器を組み合せた場合との比較，検討も行う．

1.2 論文の構成

本論文の構成は以下の通りである．

2 章では，語義曖昧性解消の手法とその関連研究について述べる．

3 章では，辞書定義文から上位概念を抽出する手法について述べる．

4 章では，語義曖昧性解消を行う分類器の作成と分類器の組み合わせ法について述べる．

5 章では，分類器を実装し，その評価実験を行い，結果の考察を行う．

6 章では，結論と今後の課題を述べる．

第2章　語義曖昧性解消に関する研究

本章では，語義曖昧性解消の先行研究を紹介し，また本研究との関係を述べる．また 2.3 節では，本研究と最も関連の深い先行研究を紹介してその問題点を示し，本研究でその問題にどのように対処するかについて述べる．

2.1　教師あり学習による語義曖昧性解消

本節では，語義曖昧性解消の一般的な手法である教師あり学習の概要について述べる．

$S \in \{s_1, ..., s_k\}$ を曖昧性を解消したい単語の語義の集合とし，$\mathbf{x} = \langle x_1, ..., x_n \rangle$ を文脈の素性を表すベクトルとする．パラメータ θ を持つ分類器 $f(\mathbf{x}, \theta)$ が語義の集合 S のうち $s_j \in S$ を最も多く選んだとき，その語義 s_j を最も可能性の高い語義とし，その単語の正しい語義 s^* として選択する．パラメータ θ は正解付きデータの集合 $L = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}(y_i \in S)$ からの訓練により決定される．

$$s^* = \arg\max_{s_j \in S} \{f(\mathbf{x}, \theta) = s_j\} \tag{2.1}$$

分類器 f を訓練するためには正解付きデータが必要である．しかし，語義を人手で与えなければならないため，大量の正解付きデータを得るのは多大な労力を要し，時間がかかる．それゆえ，実用的なアプリケーションとして使うにはコストの削減が必要となる．

文脈の素性ベクトル $\mathbf{x}$ は，対象語の前後の表記，品詞，係り受け関係にある単語の基本形，同一文中にある自立語の基本形や意味クラスなどが一般的に用いられる．分類器を学習するアルゴリズムには様々なものがある．分類器の学習アルゴリズムとして一般的な統計ベースの学習で用いられる以下のようなものが用いられる．

- 決定木 (C4.5)[14]
- 決定リスト [15]
- 最大エントロピーモデル [16]
- Naive Bayes モデル [17]
- Support Vector Machine(SVM) モデル [19]

本研究で用いる Naive Bayes モデル，Support Vector Machine(SVM) モデルについては，4.1 節で詳しく紹介する．

2.2 関連研究

語義曖昧性解消の手法として，前節で述べたような教師あり学習に基づいた方法は数多く発表されている．これに対し，本節では，本論文に特に関連が深い，あまりコーパスに現れない単語に対して様々な手法で対処した論文や，精度向上のために分類器の組み合わせを行った論文をいくつか紹介する．

2.2.1 正解タグなしコーパスを用いた 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 [1]．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，再学習を行う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している．

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に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，２つの独立な素性を定義することへの困難性も指摘されている [27]．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．Co-training に似た手法として，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 [28]．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．

Boosting を用いた分類器

AdaBoost を使った手法も提案されている [2]．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．

Park は，複数の分類器の重み付きの多数決を行うことによって，与えられた訓練例を学習すべきかどうかを判定することにより，訓練例の数を減らした．分類器は初めに少量の正解付きのデータの集合から訓練され，大量の正解なしのデータの中から訓練集合を増やしていく．具体的なアルゴリズムについて以下に示す．

L を正解付きデータの集合，C_j を j 番目の分類器，C を分類器が結合された committee（分類器の判定を行うもの）とする．入力は 2 つのデータの集合で，1 つは正解付きデータをもつ集合 L，もう 1 つは素性ベクトル $\mathbf{x}_i$ を持つ正解なしデータの集合 $D = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$ である．まず初めに，ベースの分類器 C_j を作るために，正解付きデータの集合 L から無作為に抽出した訓練集合 $L_j^{(1)}(1 \leq j \leq M)$ を用意し，L_j で分類器 C_j を訓練する．

次に，以下のステップを正解なしデータが尽きるまで繰り返す．

1. 正解なしデータの特性ベクトル $\mathbf{x}_t \in D$ に対して，分類器 C_j がそれぞれ語義 $y_i \in S$ を決める．
2. 正解なしデータ $\mathbf{x}_t$ の語義は C_j の重み W つきの多数決によって決定される．ただし，$W_1(j) = 1/M$(M は分類器の数).
3. 語義が y_t でない分類器の数の割合 ε_t を求め，その誤り率から確実性 α_t を式 (2.2) で計算する．

$$\alpha_t = (1 - \varepsilon_t)/\varepsilon_t \tag{2.2}$$

4. α_t の値が閾値 θ よりも大きいとき，α_t を重み $W_t(j)$ に掛けて新たな重み $W_{t+1}(j)$ を計算する．

$$W_{t+1}(j) = \frac{W_t(j)}{Z_t} \times \begin{cases} \alpha_t & \text{if } y_j = y_t, \\ 1 & \text{otherwise.} \end{cases} \tag{2.3}$$

Z_t は正規化するための定数である．これにより語義が正しい y_t であった分類器の重みが大きくなり，そうでない語義を選択した分類器の重みが小さくなる．閾値 θ は試行錯誤的に導かれる．

5. α_t の値が閾値 θ 以下のとき，語義が分散しているために例 $\mathbf{x}_t$ がデータの分布の分散を減らすのに役立つと考えられるので，この例を重み付き多数決で求めた語義 y_t を正解として，正解付きコーパス L_j に加える．
6. 分類器 C_j を新たな訓練集合 $L_j^{(t+1)}$ で再構成する．

$$L_j^{(t+1)} = L_j^{(t)} + \{(\mathbf{x}_t, y_t)\}. \tag{2.4}$$

これらのステップにより，最終的に以下の分類器ができる．

$$f(\mathbf{x}) = \arg\max_{y \in S} \sum_{j: C_j(\mathbf{x}) = y} W_T(j). \tag{2.5}$$

この手法で，4つの多義語に対しての実験の結果，最大で多義語の WSD の精度が 20.2%向上したと報告している．

このように，様々な手法で正解タグなしコーパスを用いた WSD が行われている．それに対して，本研究では正解タグなしコーパスを用いないで，岩波国語辞典の辞書定義文を用いることにより，低頻度語の WSD を上手く行うというアプローチを取る．

2.2.2 コーパス以外の言語資源を使用した WSD 分類器の作成

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 [3]．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 [4]．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．

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 [5]．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 に記載されている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．

辞書による語義立てにおける語義曖昧性解消

玉垣は 2 つの異なる知識源を用いて再現率の向上を図っている [30]．1 つは正解タグ付きのコーパスで，Support Vector Machine(SVM) を用いて分類器を作成した．素性に関しては，以下のような様々な素性に対して学習を行い，適切な素性を実験的に求めている．

- 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

  n を可変にして，最適な文脈の大きさを調べる．

- ローカル素性：多義語の直前，直後にある m 語の品詞と表記

  m を可変にして，最適な m の大きさを調べる．

- 意味クラス素性：多義語の前後 n 語以内に現れる自立語の意味クラス

  意味クラスは分類語彙表 [26] のシソーラス 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ている．

  - 1 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 桁まで変化させている．
  - もう 1 つは，1 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ている．

また，コーパス以外の知識源として岩波国語辞典を用いている．即ち，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用いて 2 種類の分類器を作成した．それらは，用例を用いた分類器と文法情報を用いた分類器で，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している．文法情報を用いた分類器は，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ている．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，その全てを正しい語義として出力する．

これらの複数の分類器について，ヘルドアウトデータにおける分類器単体の正解含有率を求め，一番高い分類器の出力を最終的な出力として選択している．実験の結果，組み合わせの手法を用いた分類器は SVM 単体の分類器と比べて，精度と F 値は落ちたが，再現率と適用率が向上した．

以上のように，コーパス以外の言語資源を用いる手法も様々であるが，本研究では，岩波国語辞典の辞書定義文から上位概念を抽出することにより，低頻度語の WSD の精度を上げている．

2.2.3 分類器の組合せ

WSD は様々なアルゴリズムで行われているが，それぞれ異なる特徴を持ち，異なる素性空間に対して有効であるため，分類器の組み合わせは精度を向上させるのに有効な手法である．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簡潔に紹介する．

近年，分類器の組み合わせは以下のような様々な手法で行われている．

- *Kilgarriff, Rosenzweig*[6]：SENSEVAL-1 に参加したシステムの出力を組み合わせる．
- *Pedersen*[7]：文脈の大きさを変えて，複数の Naive Bayes モデル分類器を作り組み合わせる．
- *Hoste* ら [8]：異なる文脈で訓練された 4 つの memory-based の学習器からなる分類器を学習する．
- *Florian* ら [9], *Florian, Yarowsky*[10]：6 つの異なる分類器をいくつかの異なる手法で組み合わせる．
- *Klein* ら [11]：多数決，重み付き投票，エントロピー最大モデルを用いて分類器を選択し，第 2 段階の分類器を作成する．
- *Frank* ら [12]：局所的重み付き Naive Bayes モデル．入力文に似た文を訓練データから k 近傍法で集め，それらから Naive Bayes モデル分類器を作成する．
- *Wang* ら [13]：文脈の大きさを変化させた一連の Naive Bayes モデルの分類器を作成し，その一連の分類器の判定を K 近傍法により比較し，語義を判定する．

このうち，Wang らの研究について詳しく紹介する．Wang らは文脈の大きさを変化させた Naive Bayes モデルの分類器を複数作成し，その複数の分類器の判定を K 近傍法により比較し，類似度を測ることにより，語義を選択した [13]．文脈の大きさとは，目標の多義語の左側の単語数と右側の単語数のことであり，同じアルゴリズムが使われたとしても，異なる文脈の大きさによって異なる語義選択が行われる．そこで文脈の大きさを固定せずに文脈の大きさを変えた分類器のリストを作成し用いることで，その文脈から取り出せる全ての語義選択の素性を獲得する．

詳しいアルゴリズムは以下の通りである．w を曖昧性を持つ単語とし，その n 個の語義を $s_1, ..., s_i, ..., s_n$ とする．q 個の訓練例 $S_1, ..., S_j, ..., S_q$ があり，q_i 個の例が s_i の語義タグを持つと仮定すると，$\Sigma q_i = q$ という式が成立する．この手法では 2 つの段階があり，それは Training stage と Test stage である．以下にアルゴリズムを述べる．

$p_k = (l_k, r_k)$ は文脈の大きさを表し，l_k は単語 w の左側の単語数，r_k は単語 w の右側の単語数である．T_m は trajectory（一連の分類器）であり，p_k を要素として持つ列である．

$C(p_k)$ は文脈の大きさを p_k とした，全ての語義タグ付きデータによって学習された Naive Bayes モデルによる分類器であり，文脈の大きさを変化させることにより，以下のようなベクトルが得られる．

$$C = (C(p_1), ..., C(p_k), ..., C(p_m)). \tag{2.6}$$

語義タグ s_i を持つ例 S_j に対して，分類器 $C(p_k)$ を用いて S_j を分類した結果を $\omega_j(p_k)$ とする．$\omega_j(p_k) = s_i$ であるとき，例 S_j を分類器 $C(p_k)$ の固有例と呼び，s_i を固有値とする．

それぞれの訓練例 S_j に対して，分類器 C を適用し，その結果を以下のようなベクトルで表す．

$$\omega_j = (\omega_j(p_1), ..., \omega_j(p_m)). \tag{2.7}$$

これを例 S_j の文脈の長さの trajectory T_m による decision trajectory と呼ぶ．全ての ω_j の要素が s_i であるとき，つまり $\omega_j = (s_i, ..., s_i)$ であるとき，S_j を分類器 C の固有例と呼び，ω_j を C の固有 trajectory とする．

このようにして，訓練データを訓練 decision trajectory に変換し，これを最後の k 近傍法による語義選択でのインスタンスとして用いる．

次に，新しい例が与えられたとき，まず分類器 C を用いて decision trajectory ω を計算する．

$$\omega = (\omega(p_1), ..., \omega(p_m)). \tag{2.8}$$

そして，式 (2.9) を用いて，ω と ω_j の類似度を計算する．

$$Sim(\omega, \omega_j) = \frac{\sum_{i=1}^{m} \delta(\omega(p_i), \omega_j(p_i))}{m} \tag{2.9}$$

δ は，$x = y$ のとき $\delta(x, y) = 1$，$x \neq y$ のとき $\delta(x, y) = 0$ となる関数である．

そして，ω に最も近い h 個の訓練 decision trajectory を選択し，これらのうち h_i 個の例が語義タグ s_i を持つとすると，$\Sigma h_i = h$，$h_i \leq q_i$ となる．式 (2.10) を解くことにより，新しい例の最終的な語義選択として，語義 s_i を選択する．

$$i^* = argmax_{1 \leq i \leq n} \sum_{1 \leq j \leq h_i} Sim(\omega, \omega_j) \tag{2.10}$$

全ての訓練データが固有例であった場合，同じ語義に対しての ω と ω_j の類似度は同じになり，式 (2.10) は式 (2.11) のように書き直せる．

$$i^* = argmax_{1 \leq i \leq n} \{|\{\omega(p_k) = i\}|, k = 1, ..., m\} \tag{2.11}$$

このとき，最後の k 近傍法による語義選択は新しい例の decision trajectory による多数決に簡略化されている．

この式 (2.10)，(2.11) による手法と Klein[11]，Frank[12] の手法との比較実験の結果，式 (2.10) を用いる手法が最もよい結果を示した．

以上のように，分類器の組み合わせも様々な手法が提案されている．本研究でも高頻度語に有効な分類器と低頻度語に有効な分類器の 2 つの分類器を様々な手法で組み合わせる．

2.3 定義文から抽出した語義の上位概念を用いる手法

本節では本研究と最も関係の深い八木 [29] の研究について紹介し，その問題点と本研究での解決点について述べる．

本研究では，辞書定義文から上位概念を抽出し，抽出した上位概念を反映した確率モデルを学習することにより，低頻度語の語義曖昧性解消の正解率を向上させる．上位概念を用いる理由は，擬似的に訓練データを増やす効果が得られるためである．八木の研究では，EDR 概念辞書を用いて上位概念を抽出している．しかし EDR 概念辞書は機械処理に特化して作られた辞書であり，語義の語釈文が人にとって分かりづらい表現であることがある．例えば，EDR 概念辞書の「犬」の語釈文は,「犬という動物」となっている．これは WSD を，定義文の品質が重要な場合，例えば単語の語義の語釈文を表示することによって文章の読解を支援するような読解支援システムなどに応用する場合を考えると，語釈文を人に見せても有益であるとはいえない．それと比べて岩波国語辞典では，犬の語釈文は「古くから人間が家畜として飼い親しむ，いぬ科のけだもの」となっており，人にとって非常に分かりやすい．そこで本研究では，より人にとってわかりやすく表現の豊かな岩波国語辞典を用いる．

EDR 概念辞書では 1 つの語義に対し辞書定義文は 1 つだけであり，またその形式も単純なものであったが，岩波国語辞典では 1 つの語義に複数の辞書定義文が存在する場合がある．例えば，単語の詳しい説明を記述した文や言い替え表現の文，例文，文法情報のタグ，などである．EDR 概念辞書に無いような複雑な表現に関しては，上位概念抽出パターンを追加することにより対応する．また，上位概念の抽出法を改良することにより，更なる精度の向上を目的とする．

また八木の研究では，高頻度語にも低頻度語にも有効な分類器を作るために，高頻度語に有効な SVM モデルによる分類器と，低頻度語に有効な上位概念を用いた Naive Bayes モデルを組み合わせている．しかし，その組み合わせ方は単純で，単語の頻度によって分類器を選択するものと，単語ごとに調整用データを用意し，その単語のそれぞれのモデルの正解含有率を求めて，その値の大きい方のモデルの分類器を選択するという手法を提案している．それに対し本研究では，より適切な分類器の組み合わせ方を検討するとともに，上位概念を用いた Naive Bayes モデル自体にも語義と素性の共起情報を組み込み，低頻度語，高頻度語の両方に有効な新たなモデルを提案する．

第3章　上位概念の抽出

本章では辞書定義文からの語義の上位概念の抽出に関して述べる．まず 3.1 節で語義の上位概念を抽出することにより，なぜ低頻度語の WSD に有効な分類器の学習が上手く出来るのかを明らかにする．次に 3.2 節では，どのようにして辞書定義文から上位概念を抽出するかについて述べ，3.3 節では，1 つの語義に対して複数の辞書定義文が存在する場合について，どのように上位概念を決定するかについて述べる．最後に 3.4 節では，上位概念の抽出の改良について検討する．

3.1　上位概念抽出の目的

本節では，国語辞典の辞書定義文から上位概念を抽出し，低頻度語の語義曖昧性解消に利用する基本的な考えを述べる．

分かりやすく説明するために，実際に岩波国語辞典の辞書定義文を引用し説明する．低頻度語の例として,「歳暮（せいぼ）」という単語に対して語義曖昧性解消を行うタスクを考える.「歳暮」の岩波国語辞典の辞書定義文は以下のようになっている．

27841-0-0-1-0 としのくれ．年末.
27841-0-0-2-0 この一年世話になった礼の意味で，年末に贈物をすること．その 贈物.
「お―」▽↓ちゅうげん（中元）

ここで，定義文の前に書かれた数字とハイフンの文字列は，国語辞典に割り振られた語義タグ ID を表している．この辞書定義文から具体的にどのように上位概念を抽出するかについては後の 3.2 節で述べるとして，ここでは語義 ID〔27841-0-0-1-0〕の上位概念は「年末」，語義 ID〔27841-0-0-2-0〕の上位概念は「贈物」であるとする．

このとき「歳暮」の語義と同様に,「年末」「贈物」を上位概念として持つ単語が岩波国語辞典には表 3.1 のように存在する．この表を見ると，例えば「年末」という上位概念を持つ「歳末」「歳晩」という単語の語義は,「歳暮」の 1 つ目の語義とほぼ同じような意味を持っていると言うことが出来る．このことを利用して,「歳暮」という低頻度語がコーパス中に使われていなくても，同じ上位概念を持つ単語の語義「歳末」「歳晩」がコーパス中で使われていれば，それを「歳暮」の語義曖昧性解消を行う分類器の訓練データとして使うことが出来るようになる．同じように,「贈物」を上位概念として持つ語義の単語として,「中元」「御礼」が辞書中に存在し，これらも「歳暮」の WSD を行う分類器の訓

表 3.1: 「年末」「贈物」を上位概念として持つ単語とその定義文

単語	語義タグ ID	辞書定義文
【歳末】	18911-0-0-0-0	年のくれ．年末.
【歳晩】	18882-0-0-0-0	年のくれ．年末.
【中元】	33081-0-0-0-0	七月十五日．そのころに行う贈物.「お―」
【御礼】	6358-0-0-0-0	感謝の気持を表すこと．その言葉やその気持を表す贈物.

練データとして使うことができる．ここで，以下のような文の「歳暮」の語義曖昧性解消を行う場合を考える．

1. 歳暮 の時期は毎年故郷に帰る．
2. 今年の夏は部長への 歳暮 にビールを贈った.

文 1 は「時期」という周辺の単語から「年末」を上位概念として持つ語義が正しいと判定できると思われる．文 2 は「贈った」という周辺の単語から「贈物」を上位概念として持つ語義が正しいと判定できると思われる．このように「歳暮」が訓練コーパスに現れなくても,「年末」「贈物」を上位概念として持つ単語が他に存在しているので,「歳暮」の語義判定が上手く出来る可能性が高くなる．

このように，辞書定義文から上位概念を抽出し，上位概念と周辺後の共起性を学習することにより，訓練コーパスに出現しない単語でも語義を判定できる．また，上位概念はより一般的な言葉であることが多く，複数の語義で共有されるため，同じ上位概念を持つ語義が増えて，低頻度語の訓練例の数を増やすことが出来る．これこそが上位概念を抽出する目的であり，本研究の核となる重要な考え方の 1 つである．なお，上位概念を用いた WSD モデルの詳細については 4.1.2 項で述べる．

3.2 上位概念抽出パターン

この節では，辞書定義文から上位概念を抽出する手法について述べる．基本的には辞書定義文の末尾にある単語がその語義の上位概念を表していることが多い．しかし，中にはそうではない場合が存在する．例えば，以下のような定義文がある．

【言い尽くす】　言い得ることをすべて言ってしまう.

この定義文から，単純に末尾にある単語を取り出すと,「しまう」という単語が上位概念として取り出される．しかし，これは非自立な動詞であり，明らかに誤った上位概念である．この定義文から正しく取り出すべき上位概念は「言う」であり，これを取り出すためには「＜動詞＞＋てしまう」で終わる定義文が現れたとき,「＜動詞＞」を上位概念と

して取り出すというルールを用意する必要がある．本研究ではこのような場合に適用するルールを以下のように記述する．

```
v(V てしまう)
動詞
<*><動詞><*> <て><助詞><接続助詞> <しまう><動詞><*>
b:1
```

この記述について詳しく説明する．

1 つのパターンは 4 行から成り立っており，それぞれの行が以下のような意味を表している．

1. パターン名
2. 品詞
3. パターンの内容
4. 抽出する語の位置

1 行目はパターンを識別するための名前で，実装上呼び出すために使われる．2 行目はパターンを適用する品詞で，名詞，動詞，形容詞，接尾語，その他があり，辞書定義文が定義している単語がこの品詞以外の品詞である場合にはこのパターンは適用しない．

3 行目はパターンの内容であり，辞書定義文の末尾の単語及び品詞がこの内容に一致していれば，このパターンを適用する．スペースで区切られた 1 つの文字列が 1 つの単語を表わしており，それぞれの単語は 3 つの＜＞で区切られた要素で表される．その 1 つ目の要素は**単語の基本形**で，2 つ目の要素は**単語の品詞 (第１レベル)** で，3 つ目の要素は**単語の品詞 (第２レベル)** である．単語の品詞は茶筌 [24] の品詞体系の第 1 レベル，第 2 レベルを用いている．

「v(V てしまう)」のパターンを例に説明すると，*印は任意の要素にマッチするという意味で，1 つ目の単語は動詞であれば何でもよいという意味となる．2 つ目の単語は「て」という単語でかつ品詞が「助詞-接続助詞」でなければならない．3 つ目の単語は「しまう」という動詞を表している．このパターンは 3 つの単語から成り立っているので，定義文の末尾 3 単語に対してパターンマッチが成功すれば，4 行目に指定された単語を上位概念として抽出する．

4 行目はこのパターンが適用された場合に，どの単語を上位概念として取り出すかについての記述である.「v(V てしまう)」のパターンを例に説明すると,「b:1」というのは「b」が**単語の基本形**を上位概念として取り出すことを表しており,「1」が 3 行目のパターンのうち，1 つ目の単語を上位概念として取り出すことを表している．すなわち，この「v(V てしまう)」のパターンから取り出される上位概念は，<*><動詞><*>の基本形ということになる．4 行目のコロンの前の記号として,「b」以外にも以下の記号を用いる．

- h：単語の表記を取り出す
- b：単語の基本形を取り出す
- o：他のパターンを適用した結果を取り出す
- oh：他のパターンを適用した結果のうち，最後の単語のみ表記を取り出す

他のパターンとは，パターンの中に単語の代わりに他のパターンが記述されることがあり，その際は，そのパターンを呼び出した位置を文の末尾と考えて，パターンを適用し，適用できたら上位概念の抽出結果を呼び出したパターンの抽出結果として用いることができる.「o」を用いた具体的なパターンの例を以下に示す.

```
n(N のこと)
名詞
[o(連体詞),o(形名),o(名詞)] <の|に関する|にわたる|に対する|に当たる><助詞><格助詞> <こと|もの><名詞><*>
o:1
```

このパターンの 1 つ目の単語が［］で区切られたものとなっている．これが他のパターンを呼び出すことを表しており，他に存在する,「o(連体詞)」のパターンを呼び出して，パターンマッチすればそのパターンが返す上位概念を「n(N のこと)」のパターンが返す上位概念とする.「o(連体詞)」のパターンがマッチしなければ，次に「o(形名)」のパターンを呼び出し同様に処理を行い，それでもマッチしなければ,「o(名詞)」のパターンを呼び出し同様に処理を行う.［］内の全てのパターンが適用できなかった場合には,「n(N のこと)」自身のパターンが適用できないということになる．このようにして，パターン内に他のパターンを呼び出すことにより，パターンの数を減らすことができる．なお，＜＞内の｜は,「または」の意味を表す.

実際にこのパターンを適用すべき定義文の例を以下に示す.

【雨上（が）り】　雨がやんだばかりのこと.

この定義文で，もし［o(連体詞),o(形名),o(名詞)］の部分を<*><名詞><*> として表したとすると，抽出される上位概念は「ばかり」というふうになり，適切な上位概念を取り出すことはできない．しかし，以下に示す「o(名詞)」のパターンを呼び出すことにより,「s(N ばかり)」のパターンが呼び出され,「やんだばかり」という正しい上位概念を抽出することができるようになる.

```
o(名詞)
その他
```

```
[s(「N」),s(N など),s(N ばかり),s(V たばかり),...,o(V)]
oh:1

s(N ばかり)
接尾語
<*><名詞><*> <ばかり|だけ><助詞><*>
h:1 + h:2
```

4 行目で抽出する上位概念を記述する際には，直接文字列を記述したり，文末表現タグを記述する場合もある．文末表現タグとは，主に文末に現れる表現のうち，同じ概念を表わすものをまとめる働きをする．以下に用いた文末の表現タグをまとめる．

- 単位：単位を表す
- 否定：否定を表す
- 完了：完了を表す
- 可能：可能を表す
- 受身：受身を表す
- 使役：使役を表す
- コピュラ：「〜である」「〜だ」などの総称

例として，2 つの表現が文末表現タグによりまとめられる例を以下に示す．

【肝心】とりわけ 大切であること． → 大切＜コピュラ＞+こと
【第一】一番 大切なこと． → 大切＜コピュラ＞+こと

また，パターンにはあらかじめ順番があり，その順番に従って抽出パターンを適用させる．全てのパターンにマッチしなかった定義文に関しては，上位概念を取り出さない．本研究ではこのような上位概念抽出パターンを 116 種類人手で作成した．その全てのパターンを付録 A として添付する．なお，本研究の先行研究である八木の研究では，EDR 概念辞書から上位概念を抽出するために，64 種類の抽出パターンを作成している．本研究では岩波国語辞典から上位概念を抽出しているために，定義文により複雑なパターンが多く，多くの抽出パターンが必要になったものと思われる．

以上で上位概念を抽出するパターンの記述について説明した．次に，品詞別に抽出パターンの例を取り上げる．

名詞

- 名詞で終わる文

  名詞の辞書定義文は多くの場合名詞で終わるので，文末にある名詞を上位概念として取り出す．ただし複数名詞の場合は最後の名詞のみを上位概念として抽出する．これをおこなうのがパターン「n」であり，名詞のパターンの中で最も優先度が低いパターンである．

  n
  名詞
  <*><名詞><*>
  b:1

  例：【愛】 その価値を認め，大事に思う 心．→ 心

- 「〜物」「〜法」で終わる文

  「〜物」「〜法」で終わる文では，最後の名詞だけでは上位概念が曖昧でありふさわしくないため複合名詞として取り出している．例えば，以下の辞書定義文の末尾における「法」だけを取り出すと，上位概念としては意味が広すぎる．そのため，複合名詞「攻撃法」を抽出している．

  n(複合名詞)
  名詞
  <*><名詞><*> <物|法><名詞><*>
  h:1 + b:2

  例：【アッパーカット】 ボクシングで，相手のあごを下から突き上げて打つ 攻撃法．
  → 攻撃法

- 動詞で終わる文

  名詞の辞書定義文が体言ではなく動詞で終わる場合がある．特に 1 文字の単語の辞書定義文に良くみられる．名詞の辞書定義文が動詞で終わっているとき，動詞のパターンを使って上位概念を取り出し，そして動詞の上位概念に「こと」をつけて補い，名詞として扱う．例として,「苦」の辞書定義文の上位概念抽出法を以下に示す．このパターンで使われている「o(動基 3)」とは，動詞基本パターン 3 の略で，動詞のパターンを使われ方でまとめたものである．詳細は付録 A を参照していただきたいが,「o(動基 3)」とは動詞を取り出すためのパターンである．

  n(V)
  名詞

```
[o(動基3)]
o:1 +こと
```

例：【苦】 堪えがたい圧迫を感ずる．→ 感ずる＋こと

- 「＜動詞＞ということ」で終わる文

  このパターンは「という」の部分を除くことにより，より一般的な上位概念になると思われる．

```
n(Vということ)
名詞
[o(動基3)] <という><助詞><格助詞> <こと|もの><名詞><*>
o:1 +こと
```

例：【淘汰】 生存競争によって環境に適応しない種 (しゅ) が死滅し適応するものだけが残るということ．→ 残る＋こと

動詞

- 動詞で終わる文

  動詞の辞書定義文は多くの場合動詞で終わるので，文末に現れる動詞を上位概念として取り出す．ただし，複合動詞の場合は先頭の動詞のみ取り出す．これを行うのがパターン「v」「v(複合動詞)」である．

```
v(複合動詞)
動詞
<*><動詞><*> <*><動詞><*>
b:1
```

```
v
動詞
<*><動詞><*>
b:1
```

例：【あきらめる】 とても見込みがない，しかたがないと思い切る．→ 思う

- 「＜名詞，形容詞，副詞＞する」で終わる文

  「＜名詞，形容詞，副詞＞する」で終わる文の場合は,「する」だけでなく＜名詞，形容詞，副詞＞も加えて上位概念として取り出す．

```
v(N|A|ADV する)
動詞
[o(名詞),a,o(ADV)] <する><動詞><*>
oh:1 +する
```

例：【早める】 期日・時刻などを予定より 早くする．→ 早く+する

形容詞

- 形容詞で終わる文

  形容詞の辞書定義文も多くの場合形容詞で終わるので，文末に現れる形容詞を上位概念として取り出す．

```
a
形容詞
<*><形容詞><*>
b:1
```

例：【麗しい】 よく整って 美しい．→ 美しい

- 動詞で終わる文

  形容詞の辞書定義文は動詞で終わるものも多い．この場合，動詞に「さま」を付けることにより，形容詞として扱い，上位概念として取り出す．

```
a(V)
形容詞
[o(動基 3)]
o:1 +さま
```

例：【拙 (つたな) い】 劣っている．→ 劣る＋さま

- 「＜名詞＞だ」で終わる文

  形容詞の辞書定義文は形容動詞のような形の「＜名詞＞だ」で終わる文が多い．この場合は文末表現タグの「コピュラ」(〜である，〜だなどの総称) と「さま」を付けて，上位概念として取り出す．

```
a(N だ)
形容詞
[o(形名),o(名詞)] <だ><助動詞><*>
o:1 +コピュラ+さま
```

例：【空恐ろしい】 理由は，はっきりわからないが，こわくて 不安だ.
→ 不安＜コピュラ＞＋さま

接尾語

辞書定義文には文末が上位概念以外の表現で終わるものもある．例として，多く見られる「〜の一つ」「〜の単位」「〜を表す語」について紹介する．

```
s(N の一つ)
接尾語
[o(形名),o(名詞)] <の><助詞><格助詞> <一つ|一種|総称|敬称|類|名|称|俗称|名称><名詞><*>
o:1
```

```
s(N の単位)
接尾語
[o(名詞)] <の><助詞><格助詞> <単位><名詞><*>
o:1 +単位
```

```
s(N を表す語)
接尾語
[o(形名),o(名詞)] <を><助詞><格助詞> <表す|示す><動詞><*> <語><名詞><*>
o:1
```

例：【青写真】 図面などの複写に使う 写真の一つ. → 写真
例：【アンペア】 電流の 強さの単位. → 強さ＜単位＞
例：【位】 等級・程度を示す語. → 程度
「位」の定義文での「等級」と「程度」のように並列関係にある単語が上位概念の候補となる場合，どちらがより優れているかは決められないので，本研究では後者を選ぶことにする．

その他

その他に，品詞のパタンマッチには使わないが，他のパタンマッチ規則の部品として使うパターンがある．これらはいくつかのパターンをひとまとめにしておいて，パターン内で呼び出すときに使われる．例として形式名詞のパターンを集めたパターン「o(形名)」を示す．

```
o(形名)
```

その他
[n(V ている+形名),n(V ていない+形名),n(V|A|N ない+形名),n(V た+形名),n(N である+形名),n(V+形名),n(A+形名),n(C+形名)]
oh:1

なお，本研究では，名詞，動詞，形容詞以外の品詞の単語からは上位概念を抽出しない．

3.3 複数の定義文の取り扱い

先行研究の八木の研究 [29] で上位概念抽出に用いられた EDR 概念辞書では，1 つの語義に対し辞書定義文は 1 文だけであり，またその形式も単純なものであったが，岩波国語辞典では 1 つの語義に複数の辞書定義文が存在する場合がある．さらに，より詳細な意味を記述した文や，由来・性質・使い方など，直接意味を表現するものではない文などが混在する．また，EDR 概念辞書では別の語義として記述されていたものが，岩波国語辞典では 1 つの語義として記述されているような場合もある．具体的な例として，それぞれの辞書の「教育」の定義文を以下に示す．

> EDR：教えこんで身につけさせること
>
> 岩波： 教えて知能をつけること．人の心身両面にわたって，またある技能について，その才能を伸ばすために教えること．

「教育」は，岩波国語辞典の定義文では，1 つの語義に対し 2 つの文があり，少し違った表現で「教育」を定義している．

このように岩波国語辞典では，1 つの語義に対して複数の定義文が用意されていることがある．このことによって，辞書定義文から上位概念を抽出する際に，ある問題が起きる．それは，1 つの語義に対して複数の定義文が存在するため，そのまま上位概念抽出パターンを適用すると，複数の上位概念が取り出されることである．4.1.2 項で述べる WSD の確率モデルでは，語義の上位概念は 1 つであることを仮定している．そのため，複数の定義文が存在し，複数の上位概念が抽出される場合には，上位概念を 1 つに絞り込む必要がある．本節では，複数の定義文の対処について詳しく説明する．

複数の定義文の分類タイプ

1 つの語義に対し複数の辞書定義文が存在する場合には，辞書定義文の第 2 文以降を以下のように分類し，それに基づいて処理を行う．

- 同語義定義文：1 つ前の文と同じもしくは似た意味，言い替え
  例：【全治】 病気や傷が，すっかり直ること．全快.

- 別語義定義文：1つ前の文とは異なる意味，別の表現
  例：【代行】 本人に代わって物事を行うこと．また，その人．
- 非定義文：例・由来・性質・使い方など，直接意味を表現するものではないもの
  例1:【安山岩】火成岩の一種．暗灰色で，緻密 (ちみつ)．建築・土木用に多く使われる．
  例2：【蜂】 膜翅 (まくし) 目の昆虫．丈夫な膜質の羽があり，高等なはちでは産卵管が毒針となって，ふだんは腹の中にしまわれている．ミツバチ・スズメバチなど種類が多い．

なお，今回岩波国語辞典の辞書定義文を調べた結果，本研究で想定した非定義文は名詞の定義文にのみ存在しており，動詞，形容詞の定義文には見当たらなかった．よって，動詞，形容詞の第2文以降の定義文に関しては，同語義定義文，もしくは別語義定義文のどちらかに分類する．

タイプ分類パターン

第2文以降の文をこれらのタイプに分類するには，以下のような分類パターンを用いる．

- 非定義文
  - 第1文から抽出した上位概念が「動物・植物・昆虫・魚・高木」などであれば，その語義の第2文以降は全て非定義文．
  - 第1文の定義文の文末が「〜の一つ・一種・名称・単位」などであれば，その語義の第2文以降は全て非定義文．
  - 第2文以降の定義文の文頭が「例」であれば，その語義の次の文以降は全て非定義文．
  - 第2文以降の定義文の文末が「〜使う・用いる・言う・読む・用」などであれば，その語義の次の文以降は全て非定義文．
- 別語義定義文
  - 第2文以降が「また・もと・転じて・比ゆ的に」などで始まれば，その定義文は別語義定義文．
- 同語義定義文
  - 非定義文，別語義定義文と判定された文以外の文全て

これらは実際の辞書定義文と上位概念を見て人手で作成した．パターンマッチで用いるキーワードは全部で 38 種類ある．付録 B に全てのパターンを掲載する．

これらのパターンを適用することにより，ランダムに選んだ複数の定義文から成る 200 個の語義の定義文において，誤ったタイプの分類を行った語義は 13 個であった．したがって，上記のパターンによる定義文の分類はある程度信頼できる．

複数の定義文からの上位概念の選択

複数の定義文をパターンによって分類した後，以下の順序で抽出すべき上位概念を決定する．

1. 非定義文であれば，その定義文は直接意味を表現するものではないので，その定義文からは抽出しない．
2. 同語義定義文（似た意味の上位概念）であれば，複数の定義文から得られる上位概念は似た意味を持つとみなせる．ここでは，各上位概念の辞書中の出現頻度を調べ，一番大きいものを選択する．なぜなら，多くの語義で共有される方が訓練データの数を増やす効果が高いからである．その辞書中の出現頻度によるスコアは，1/(その語義の定義文の数) とした．例えば，1 つの語義の定義文から 4 つの異なる上位概念が抽出されれば，そのそれぞれの上位概念のスコアに 0.25 を加える．これを辞書中の全ての語義に対して行い，そのスコアを合計したものを用いる．
   例：【選評】 多くの作品からよいものを選んで 批評すること．その 批評.
   辞書中の頻度：「批評+する+こと」4.5,「批評」16.17
   よって,【選評】の上位概念は「批評」となる．
3. 別語義定義文（全く別の上位概念）であれば，得られる複数の上位概念は全く別の意味とみなせる．
   例：【タンゴ】 アルゼンチンから起こった四分の二拍子のダンス 曲．また，それにあわせて踊る ダンス.
   これは 1 つの語義が複数の意味を持っているとみなせる．したがって，単語がどちらの意味で使われるかは単語が出現する文脈によって異なるため，どちらの候補が正しいかを決めることは出来ない．
   4.1.2 項で後述するように，上位概念を用いた分類器は，候補となる語義の出現確率を求め，最大の確率を持つ語義を選択する．このような場合は，複数の上位概念のそれぞれについて確率を計算し，より高い確率を持つ上位概念を選択する．

具体的に「解釈」の辞書定義文に対して，この手順で上位概念を選択する例を示す．
まず,「解釈」の辞書定義文は以下の通りである．

> 【解釈】 文章や物事の意味を，受け手の側から理解すること．また，その理解したところを説明すること．その内容.

これを分かりやすく各文ごとに記述する。

1. 文章や物事の意味を，受け手の側から 理解すること． → 理解+する+こと
2. また，その理解したところを 説明すること． → 説明+する+こと
3. その 内容． → 内容

矢印の右側はそれぞれの文から抽出される語義の上位概念である．ここで，以下のパターンが適用され，第 2 文が別語義定義文，第 3 文が同語義定義文と分類される．

第 2 文以降：「また」で始まる → 別語義定義文

したがって，この定義文中には，第 1 文で定義される語義と，第 2,3 文で定義される語義の 2 つがあるとみなせる．後者からは 2 つの上位概念「説明+する+こと」と「内容」が抽出されるが,「説明+する+こと」の辞書中の頻度は 8.67,「内容」の辞書中の頻度は 36.08 であり，辞書中の頻度が高い「内容」が選択される．最終的に「理解+する+こと」と「内容」が「解釈」の上位概念として選択される．曖昧性解消を行う際，2 つの上位概念のそれぞれについて確率を計算し，大きい方を選択する．

3.4 上位概念の抽出法の改良

これまで上位概念を抽出する方法について述べてきたが，今まで述べてきた方法では，あまり適切でない上位概念が抽出される場合がある．それは，上位概念抽出の目的である，低頻度の語義を上位概念に置き換えることにより，多くの語義を同じものとみなすことによりデータを増やすはずが，抽出した上位概念自体も低頻度である場合には抽出した上位概念が他の語義のものと一致せず，データを増やすことができないため上位概念に置き換えた効果があまりない場合である．よって，本節では上位概念の頻度が低い場合に，より頻度の高い上位概念に置き換えることによりデータを増やし，上位概念を用いる効果を得ることを試みる．

表 3.2 は実際に抽出した上位概念の頻度別の異なり数とのべ数を示したものである．これを見れば分かるように，抽出した上位概念のうち，異なり数で見たとき 18964(92.82%) が頻度 10 未満，17307(84.71%) が頻度 5 未満である．全体ののべ数 (79481) で見ても約 1/2 が頻度 10 未満となっている．

このように低頻度の上位概念は多く存在する．これらを少しでもより高頻度の上位概念に置き換えるために，次のような手法を試みる．

1. 頻度 T_l 未満の低頻度の上位概念に対して，先頭から 1 文字削っていく．
2. 削った語が他の頻度 T_h 以上の高頻度の上位概念のどれかと一致すれば，それを新たな上位概念として置き換える．
3. ステップ 1，2 を T_d 回だけ，新しい上位概念が見つかるまで行う．ただし，T_d =ALL のときには最後の 1 文字まで行う．最後まで一致しなければ置き換えない．

表 3.2: 上位概念の頻度別異なり数とのべ数

頻度	1855	···	9	8	7	6	5	4	3	2	1	ALL
異なり数	1	···	193	228	295	385	556	863	1551	3153	11740	20430
のべ数	1855	···	1737	1824	2065	2310	2780	3452	4653	6306	11740	79481

表 3.3: 上位概念の抽出法の改良

改良前	頻度	改良後	頻度
研究所	2	所	47
長いもの	2	もの	30
挽き肉	1	肉	35
田舎者	1	者	418
煎茶	3	茶	20
品目	2	目	60
けじめ	2	め	10
自給自足	1	足	24

T_l, T_h, T_d の値はいくつか変化させ，その上位概念を用いた WSD の実験を行い最適な値を探す．なお，この手法を適用するのは名詞の上位概念のみであり，動詞，形容詞の上位概念については適用しない．また，＜受身＞，＜コピュラ＞などの文末表現タグや，「+こと」，「+さま」などのように削ると形式名詞のみが残ってしまうような場合は，この手法は適用しない．この手法で置き換えられた上位概念の具体例を表 3.3 に示す．この表で挙げたもののうち，上 5 つは正しく上位概念が得られた場合で，下 3 つは誤った上位概念が得られた場合である．このように，本手法では誤った上位概念が得られることもあり，完璧な手法とはいえない．しかし，正しく上位概念が得られる場合の方が誤る場合よりも多く，また低頻度の上位概念のままではどちらにしてもデータを増やすことができず上位概念を用いる効果が得られないということを考慮して，この改良案を採用した．

表 3.4 は $T_l = 10, T_h = 10, T_d =$ALL のときの上位概念の頻度別の異なり数とのべ数を示したものである．表 3.2 と比較すると，抽出した上位概念のうち，異なり数で見たとき

表 3.4: 改良抽出法による上位概念の頻度別異なり数とのべ数

頻度	2007	···	9	8	7	6	5	4	3	2	1	ALL
異なり数	1	···	149	188	252	328	458	699	1276	2609	9758	17183
のべ数	2007	···	1341	1504	1764	1968	2290	2796	3828	5218	9758	79481

15717(91.47%) が頻度 10 未満，14342(83.47%) が頻度 5 未満とあまり割合は変わらないが，のべ数で見ると頻度 10 未満の数が 6400 個減っており，全体ののべ数 (79481) で見ると 0.4 以下にまで頻度 10 未満の数が減っていることになる．

本研究では，上位概念の抽出法の改良についてこれ以上の検討を行っていないが，今後はより良い改良の手法を見つけて，誤った上位概念が得られる場合を減らす必要がある．

第4章　語義曖昧性解消モデル

本章では，語義曖昧性解消の具体的な手法について述べる．4.1 節では，特に高頻度語に有効な教師あり学習の代表的なアルゴリズムである SVM を用いた分類器と，上位概念を用いることにより低頻度語の WSD を目的とした Naive Bayes モデルによる分類器について詳しく述べる．4.2 節では，前節で述べた SVM による分類器と Naive Bayes モデルによる分類器を組み合わせることによって，両者の利点を活かした分類器を作成する様々な手法について述べる．4.3 節では，新たな試みとして，語義と上位概念を同時に反映する Naive Bayes モデルを提案する．

4.1　分類器の概要

4.1.1　Support Vector Machine(SVM)

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 [19]．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 (4.1)，(4.2) のように表せる．

$$x_i \cdot \mathbf{w} + b \geq +1 \quad for \quad y_i = +1 \tag{4.1}$$

$$x_i \cdot \mathbf{w} + b \leq -1 \quad for \quad y_i = -1 \tag{4.2}$$

まとめると

$$y_i(x_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i \tag{4.3}$$

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 (4.3) 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 4.1 にこの問題を図式化した．

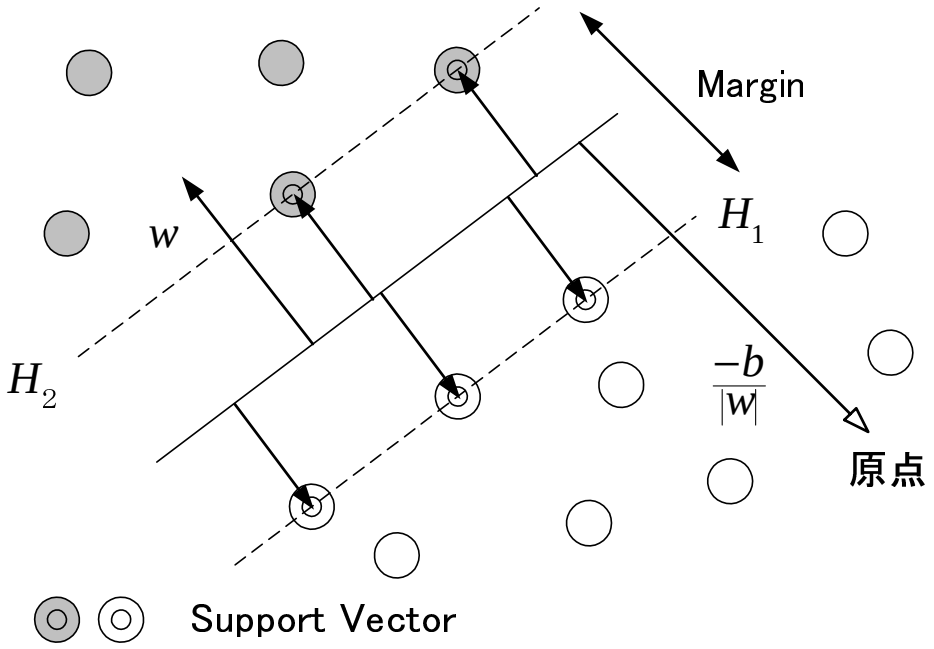

図 4.1: SVM 概要

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．

$$L_P \equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \tag{4.4}$$

また，式 (4.4) 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

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{4.5}$$

$$\sum_i \alpha_i y_i = 0 \tag{4.6}$$

となる．式 (4.5)，式 (4.6) を式 (4.4) に適用すると，L_P と同値の双対問題 L_D は

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{4.7}$$

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．

本研究では，SVM の学習には TinySVM [1]を使用した．使用したカーネルは線形カーネルである．また，SVM は二値分類器であるが，多値分類問題である WSD に適用するために one vs rest 法を用いた．one vs rest 法では，k 値分類問題において，入力ベクトル $\mathbf{x}$ がクラス $i(1 \leq i \leq k)$ であれば正例，それ以外ならば負例と判別する分類器 $g_i(\mathbf{x})$ を各クラスについて SVM で学習する．ここで $g_i(\mathbf{x})$ の絶対値は予測の信頼度 (決定境界からの距離) を表すので，最終的な k 値分類関数 $f(\mathbf{x})$ は

$$f(\mathbf{x}) = \arg\max_i g_i(\mathbf{x}) \tag{4.8}$$

となる．

素性

本節では学習に用いた素性を述べる．以下に挙げる素性を得るため，コーパスに形態素解析器として茶筌 [24] を，文節の係り受け解析器として Cabocha[25] を用いている．

- $S(0), S(-1), S(-2), S(+1), S(+2)$
  対象語及びその周辺にある語の表記．括弧内の数値は対象語からの位置を表わす．
- $P(-1), P(-2), P(+1), P(+2)$
  対象語の周辺にある語の品詞．品詞は茶筌の品詞体系を用いた．
- $S(-2) \cdot S(-1), S(+1) \cdot S(+2), S(-1) \cdot S(+1)$
  対象語の周辺にある 2 つの語の表記の組
- $P(-2) \cdot P(-1), P(+1) \cdot P(+2), P(-1) \cdot P(+1)$
  対象語の周辺にある 2 つの語の品詞の組
- B_{sent}
  同一文中にある自立語の基本形．但し，数字については “NUM” という特別なシンボルを素性として用いた．
- C_{sent}
  同一文中にある自立語の意味クラス．意味クラスは日本語語彙体系 [26] を用いた．ただし，自立語が多義 (複数の意味クラスを持つ) のときには素性に加えないこととした．

[1] http://chasen.org/%7Etaku/software/TinySVM/

- B_{bs_head}, B_{bs_mod}

  対象語が文節の主辞であるとき，その文節の係り先文節の主辞の基本形 (B_{bs_head}) と係り元文節の主辞の基本形 (B_{bs_mod}).

- B_{bs_in}

  対象語が文節の主辞でないとき，その文節の主辞の基本形.

- $(B_{case}; B_{noun})$

  対象語が動詞のとき，その動詞の格 (B_{base}) と格要素 (B_{noun}) の基本形の組.

- $(B_{case}; C_{noun})$

  対象語が動詞のとき，その動詞の格 (B_{base}) と格要素の意味クラス (C_{noun}) の組．意味クラスは日本語語彙体系を用いた．ただし，自立語が多義のときには，その全てを素性に加えた.

- $(B_{case}; B_{verb})$

  対象語が名詞で，かつある動詞の格要素となっているとき，その格 (B_{base}) と動詞の基本形 (B_{verb}) の組.

以下はコーパスから抜き出した多義語「違い」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／。

- $S(0) =$ 違い，$S(-1) =$ ひと味，$S(-2) =$ は，$S(+1) =$ 、，$S(+2) =$ 主人公
- $P(-1) =$ 名詞-一般，$P(-2) =$ 助詞-係助詞，$P(+1) =$ 記号-読点，$P(+2) =$ 名詞-一般
- $B_{sent} =$ ていねい，ひと味，ドラマ，刑事，今，主人公，取り巻く，心理，日常，彼，描く，描写
- $C_{sent} =$ 1037，1057，1238，1771，2699 (シソーラスの ID)
- $B_{bs\_head} =$ 描く
- $(B_{case}, B_{noun}) =$ (トハ，ドラマ)
- $(B_{case}, C_{noun}) =$ (トハ，1057)

4.1.2 上位概念を用いたNaive Bayesモデル

本節では，辞書定義文から抽出された上位概念を用いた語義曖昧性解消モデルに付いて述べる．まず，ある単語 w の語義を決定するために，以下のような確率モデルを用いる．

$$P(s, c|F) \tag{4.9}$$

式 (4.9) において s は w の語義を示し，c は辞書定義文から抽出された s の上位概念である．また，F は w を含む入力文から得られる素性集合であり，w の周辺語の表記や品詞などである．素性の具体的な内容については後で述べる．

次に式 (4.9) を以下のように近似する．

$$P(s, c|F) = P(s|c, F)P(c|F) \simeq P(s|c)P(c|F) \tag{4.10}$$

ここでは式 (4.10) の $P(s|c, F)$ を $P(s|c)$ として近似している．$P(s|c, F)$ は入力文から得られる素性集合 F と上位概念 c から語義 s を予測するモデルであり，F から s を予測するという点で，通常の Naive Bayes モデルによる語義曖昧性解消モデルとほぼ同じである．しかし，低頻度語については語義 s の出現頻度が低いため，統計的に信頼できるモデルが学習できないと考えられる．そのため，語義 s は語義の上位概念 c のみに依存するとみなして，$P(s|c)$ のように近似する．一方，式 (4.10) の $P(c|F)$ は素性集合 F から語義の上位概念 c を予測するモデルである．3.1 節で述べた通り，語義の上位概念は複数の単語で共有されることから，語義 s よりもコーパスにおける出現頻度は高いため，$P(c|F)$ は低頻度語の語義曖昧性解消を行う場合でも十分学習可能である．

次にベイズの定理を用いて以下のような変形を行う．

$$\begin{aligned} P(s|c)P(c|F) &= \frac{P(s)P(c|s)}{P(c)}\frac{P(c)P(F|c)}{P(F)} && (4.11)\\ &= \frac{P(s)P(F|c)}{P(F)} && (4.12)\\ &\simeq \frac{P(s)\prod_{f_i \in F} P(f_i|c)}{P(F)} && (4.13) \end{aligned}$$

式 (4.11) から式 (4.12) の変形では $P(c|s) = 1$ とした．これは語義 s の辞書定義文から抽出される上位概念 c は常に一意に決まるためである．また，式 (4.12) から式 (4.13) の変形において，F 中の各素性 f_i の出現は互いに独立であると仮定して近似している．これは Naive Bayes モデルで用いられる一般的な近似である．

本研究では，式 (4.13) の確率を最大にする語義 s を選択することによって語義曖昧性解消を行う．式 (4.13) が最大となる語義が複数存在するときは，その全てを出力する．ここでは，全ての語義について F は同じであるため，$P(F)$ の計算は省略可能である．

$$\begin{aligned} & \arg\max_{s \in S_w} \frac{P(s)\prod_{f_i \in F} P(f_i|c)}{P(F)} && (4.14)\\ = \ & \arg\max_{s \in S_w} P(s) \prod_{f_i \in F} P(f_i|c) && (4.15) \end{aligned}$$

式 (4.15) の S_w は辞書に登録されている w の語義の集合である．直感的に言えば，式 (4.15) の第 1 項 P(s) は語義の出現頻度を学習するモデルであり，第 2 項 $\prod P(f_i|c)$ は語義の上位概念 c と素性 f_i の共起性を学習するモデルである．

素性

式 (4.15) のモデルに用いる素性 f_i として以下のものを用いた．これらは SVM のときと同様，解析対象となる文に対し，形態素解析器として茶筌 [24] を，文節の係り受け解析器として Cabocha[25] を用いて取り出したものである．またこれらは，4.1.1 項で述べた SVM の素性からいくつかの素性を除いたものである．これは，予備実験で検討したところ，SVM の素性を全て用いるよりも結果が良かったためである．

- $S(0), S(-1), S(+1)$

  対象語及びその周辺にある語の表記．括弧内の数値は対象語からの位置を表わす．

- $P(-1), P(+1)$

  対象語の周辺にある語の品詞．品詞は茶筌の品詞体系を用いた．

- B_{sent}

  同一文中にある自立語の基本形．但し，数字については “NUM” という特別なシンボルを素性として用いた．

- B_{bs_head}, B_{bs_mod}

  対象語が文節の主辞であるとき，その文節の係り先文節の主辞の基本形 (B_{bs_head}) と係り元文節の主辞の基本形 (B_{bs_mod})．

- B_{bs_in}

  対象語が文節の主辞でないとき，その文節の主辞の基本形．

- $(B_{case}; B_{noun})$

  対象語が動詞のとき，その動詞の格 (B_{base}) と格要素 (B_{noun}) の基本形の組．

- $(B_{case}; B_{verb})$

  対象語が名詞で，かつある動詞の格要素となっているとき，その格 (B_{base}) と動詞の基本形 (B_{verb}) の組．

パラメタ推定

式 (4.15) に示すように，本研究で推定するべき確率モデルは $P(s)$ と $P(f_i|c)$ である．まず，$P(s)$ は以下の式のように加算スムージングで推定する．

$$P(s) = \frac{O(s) + \alpha}{\sum_s O(s) + \alpha V} \tag{4.16}$$

$O(s)$ は語義 s の出現頻度，α は全ての事象に足すべき頻度，V は辞書中の語義の総数を示している．ここでは $\alpha = 0.5$ とした．一方，$P(f_i|c)$ は線形補完法で推定した．すなわち以下の式 (4.17) のように素性の出現頻度確率 $P(f_i)$ との混合モデルとして推定する．両者の重みづけ β は，調整用データにおいて値を変化させて，より精度の高かったものを選択する．$P_{MLE}(f_i|c)$ は式 (4.18) の最尤推定によって推定を行った．一方 $P(f_i)$ は式 (4.19) で推定した．T は訓練データの総数であり，分子と分母にそれぞれ 1 と 2 を加えているのはスムージングを行うためである．また $O(f_i)$ は素性の頻度を示し，$O(f_i, c)$ は素性と上位概念の共起関係の頻度を示している．

$$P(f_i|c) = \beta P_{MLE}(f_i|c) + (1 - \beta)P(f_i) \tag{4.17}$$

$$P_{MLE}(f_i|c) = \frac{O(f_i, c)}{\sum_{f_i} O(f_i, c)} \tag{4.18}$$

$$P(f_i) = \frac{O(f_i) + 1}{T + 2} \tag{4.19}$$

訓練コーパスから各素性を取り出した後，語義の頻度 $O(s)$，素性の頻度 $O(f_i)$，素性の頻度と上位概念の共起頻度 $O(f_i, c)$ をカウントし，モデルを学習した．

4.2 分類器の混合モデル

本研究は，高頻度語のための SVM による分類器 (4.1.1 節) と低頻度語のための上位概念を用いた Naive Bayes モデル分類器 (4.1.2 節) の 2 つを組み合わせる．組み合わせ手法として，大きく分けて以下の 3 つを試みた．

4.2.1 頻度に基づく混合モデル

訓練データにおける出現頻度がある閾値以上なら SVM による分類器を，それ以外は上位概念を用いた Naive Bayes モデル分類器を選択する．5 章の実験ではこの閾値を 5 とした．

4.2.2 正解含有率に基づく混合モデル

共通の調整用データを用意し，それぞれの分類器単体の正解含有率 (式 (4.20)) を調べる．

$$\text{正解含有率} = \frac{\text{出力した語義に正解が含まれる単語数}}{\text{分類器が語義を一つ以上出力した単語数}} \tag{4.20}$$

単語毎に調整用データにおける正解含有率を求め，それが高い分類器の出力を最終的な出力として選択する．また，調整用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比較する．5 章の実験では $Oh = 10$ とした．

正解含有率ではなく精度 (適合率) を比較することも考えられる．しかし，本研究は再現率の向上を第一の目標としている．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れば効果があるとして，正解含有率の比較を行った．

4.2.3 スタッキング

スタッキングとは Wolpert によって提案された手法で，複数の分類器を同じ訓練データで訓練し，それらの分類器の出力を集め，それらの出力に元の素性に加えたものを素性として新たな分類器を学習することにより，分類の精度を向上させる手法である [18]．

本節ではスタッキングの手法を用いて，SVM による分類器と上位概念を用いた Naive Bayes モデル分類器の出力を素性とする新たな分類器を学習し，語義曖昧性解消を行う手法について述べる．

シンプルスタッキング

スタッキングで新たに学習する分類器は SVM モデルを用いる．素性は 4.1.1 項で述べた SVM の素性に，以下の素性を加えることにより，スタッキングを行う．

- SVM による分類器の判別語義
- 上位概念を用いた Naive Bayes モデル分類器の判別語義

分類器の判別語義は岩波国語辞典の語義 ID をそのまま用いる．

図 4.2 はこのスタッキングの手法の概念を図で表したものである．SVM は SVM モデル分類器，NB は Naive Bayes モデル分類器の略である．SVM と NB の 1 次分類器の判別語義を SVM の 2 次分類器の素性に加えて用いる様子を表している．

本研究では，1 次分類器で訓練データに少量のテストデータを加えたデータに対して語義の判別を行い，その判別語義を新たに素性に加えた訓練データと少量のテストデータを用いて 2 次分類器を訓練した．語義の判別を行う際には，学習した 1 次分類器で判別した語義を素性に加えて，学習した 2 次分類器を用いて最終的な語義を決定した．また，2 次

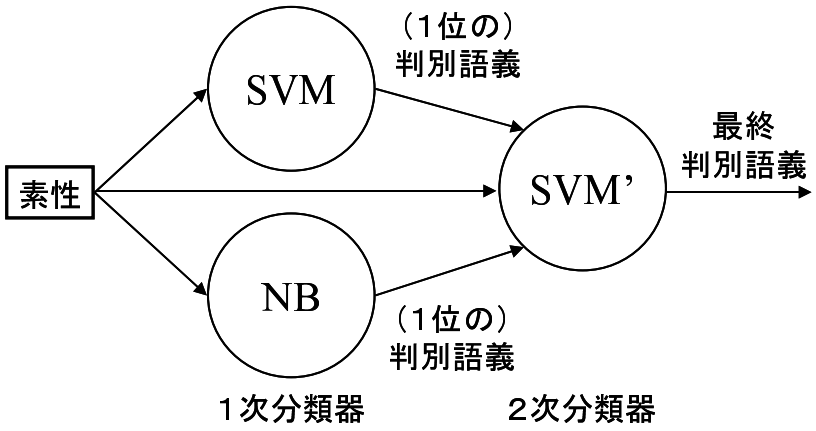

図 4.2: スタッキング概念図

分類器は TinySVM を用いて作成した．カーネルは線形カーネルを用いた．本来のスタッキングの手法では，最終的な分類器の役割は複数ある分類器のどれを選択するかというものである．しかし，本研究では分類器は SVM と NB の 2 つしかなく，この方式では精度の高い SVM しかほとんど選択しなくなってしまう．そこで，単語ごとに語義タグ候補の中から正解の語義を選択するという，1 次の SVM 分類器の方式と同様な手段をとった．

より詳しい実験の内容に関しては，5 章で述べる．

交差検定を用いたスタッキング

さらに本研究では交差検定 (cross validation) を用いる手法でもスタッキングを行った．元の訓練データを単語ごとに 5 分割し，1 つをテストデータ，残り 4 つを訓練データとして 1 次分類器を学習し，そのテストデータの判別語義をマージして，2 次分類器の訓練データの素性に加えた．評価の際，2 次分類器に素性として加える 1 次分類器の判別語義は，訓練データ全体を用いて訓練された分類器に判別させたもの (即ち，通常の分類器の判別結果) を用いた．2 次分類器は TinySVM を用いて作成し，カーネルは線形カーネルを用いた．

単語ごとに 5 分割するとは，例えば元の訓練データに 10 の例がある単語に対して，テストデータとして 2 つの例，訓練データとして 8 つの例を用いる組み合わせを 5 回行い，10 の全ての例の判別結果を得るということである．

なお，訓練データを判別した結果を用いて新たな分類器を作るというスタッキングの本来の手法とは異なっており，厳密に言えば，この手法はスタッキングとは言えない．しかし，本来のスタッキングの手法では，訓練データ自体を判別した結果を新たな分類器に素性として加えるため，2 つの分類器を組み合わせるときに，片方の分類器がもう一方よりも全体的に精度が高いと，その精度の高い方しかほとんど選択しないという問題がある．本研究では，SVM が全体として精度の高い分類器にあたり，Naive Bayes は精度の低い分

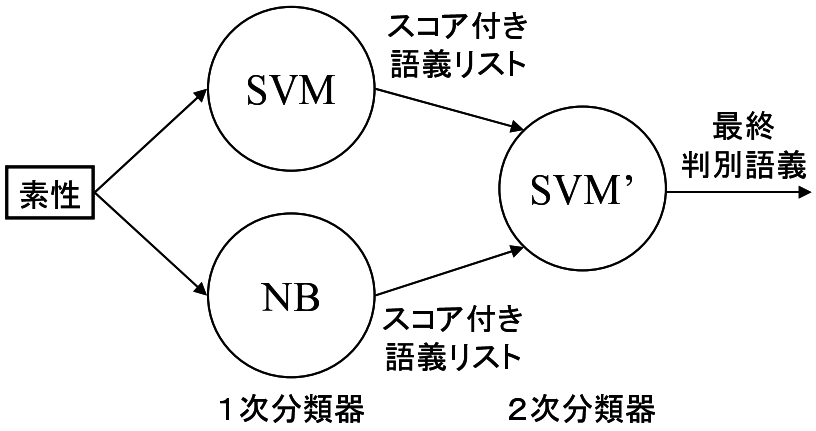

図 4.3: スコアを用いたスタッキング概念図

類器にあたる．それを解決するために，シンプルスタッキングでは調整用データを加えることにより，Naive Bayes が有利に働く例を見つけたかったが，結果的にあまり効果がなかった．そこでこの問題を和らげるために，交差検定を行うことによって，1 次分類器の訓練データとテストデータを別にすることにより，全体としては精度の低い分類器が有利に働く場合を見つけることができる可能性があると考え，この手法を行った．

より詳しい実験の内容に関しては，5 章で述べる．

スコアを用いたスタッキング

これまで説明したとおり，スタッキングは元の素性に複数の分類器の出力を加えて新たな分類器を学習するのが一般的だが，本研究ではこれと異なる手法も試みる．その手法とは，複数の分類器の出力として判別した語義だけではなく，語義候補全てにスコア付けをし，それを素性として新たな分類器を学習する手法である．図 4.3 はスコアを用いたスタッキングの手法の概念を図で表したものである．

具体的には以下の素性を用いて新たな分類器を学習する．

- SVM における各語義候補の分離平面との距離
- Naive Bayes モデルにおける各語義候補の確率値 (式 (4.13) の値)

ただし，これらの値を訓練事例毎に最上位の語義のスコアが 1 となるように正規化して用いる．

例として,【漂う】という単語に対するスコアを以下に示す．記述形式は，(出力した分類器):(語義候補)/(スコア) である．ちなみに，コーパスにあるこの例の正解の語義は 31544-0-0-4-0 である．

素性のスコアの例:【漂う】

SVM:31544-0-0-3-0/1,SVM:31544-0-0-4-0/0.9280,SVM:31544-0-0-1-0/0.7304,
NB:31544-0-0-3-0/1,NB:31544-0-0-4-0/0.5836,NB:31544-0-0-1-0/0.5467,…

このように，今までの手法では各分類器が第 1 候補 (この【漂う】の例でいうと〔31544-0-0-3-0〕) しか選択できなかったが，この手法ではスコアを付けることで全ての語義候補の確率を学習でき，第 1 候補に近い確率である第 2 候補，第 3 候補なども選択する可能性がある．よって，今までの手法では分類器が正解の語義を選択できなかった場合でも，正解の語義を選択できる可能性がある．

この素性を用いて，TinySVM を用いて SVM モデルによる分類器を作成し，語義を選択する．このとき，線形カーネルでは TinySVM での学習が収束しない場合が多かったので，多項式 (3 次) カーネルを用いた．それでも収束しない場合が全体の 2%ほど存在したので，5 章の実験ではその単語を評価から除くこととした．また，スコアを用いたスタッキングでも，先ほど説明した交差検定を用いる手法で分類器を作成した．

より詳しい実験の内容に関しては，5 章で述べる．

4.3 語義と上位概念を同時に反映する Naive Bayes モデル

4.2 節では，多くの単語に対して適用可能な頑健な語義曖昧性解消を行うために，高頻度語に有効な SVM モデル分類器と，低頻度語に有効な上位概念を用いた Naive Bayes モデル分類器を，様々な手法で組み合わせた．

本研究では，この分類器の組み合わせによる手法だけではなく，1 つの分類器において頑健な語義曖昧性解消を行う手法についても試みた．2 つの共起情報を同時に反映した 1 つのモデルを作成し，そのモデルを学習する．即ち，高頻度語に有効である語義と素性の共起情報と，低頻度語に有効である上位概念と素性の共起情報を別々に学習し，それらを掛け合わせることにより，2 つの共起情報を両方とも反映した 1 つのモデルを作ることができるのではないかと考えた．この手法では，2 つの分類器を組み合わせずに多くの単語に対して多義性解消を行えるので効率が良い．語義を s，上位概念を c，素性集合を F とすると，単語の語義を決定する確率モデルは $P(s|F)P(c|F)$ と表され，これをベイズの定理を用いて変形すると式 (4.23) のようになる．

$$P(s|F)P(c|F) = \frac{P(s)P(F|s)}{P(F)}\frac{P(c)P(F|c)}{P(F)} \tag{4.21}$$

$$= \frac{P(s)P(c)P(F|s)P(F|c)}{P(F)^2} \tag{4.22}$$

$$= \frac{P(s)P(c)\prod_{f_i\in F}P(f_i|s)P(f_i|c)}{P(F)^2} \tag{4.23}$$

ここで，現実に得られる量のコーパスで確率の推定を可能にするため，語義 s と上位概念 c が独立であるという特殊な仮定を行っている．この仮定は，4.1.2 項での仮定とは全く

異なるものであり，そういう意味で式 (4.23) は，4.1.2 項で述べた上位概念を用いた Naive Bayes モデルの式とは全く別のものである．つまり，この確率モデルは $P(s, c|F)$ を直接求めるのではなく，語義と素性の共起情報と上位概念と素性の共起情報を同時に活かすための指標として，上記の確率モデルを用いている．

式 (4.23) の値を最大にする語義 s を分類器が選択する語義とする．式 (4.23) が最大となる語義が複数存在するときは，その全てを出力する．ここでは，全ての語義について F は同じであるため，$P(F)^2$ の計算は省略可能である．

$$\arg\max_{s \in S_w} \frac{P(s)P(c)\prod_{f_i \in F} P(f_i|s)P(f_i|c)}{P(F)^2} \tag{4.24}$$

$$= \arg\max_{s \in S_w} P(s)P(c) \prod_{f_i \in F} P(f_i|s)P(f_i|c) \tag{4.25}$$

なお，用いる素性は 4.1.2 項の上位概念を用いた Naive Bayes モデルのものと同様である．

パラメタ推定

式 (4.25) に示すように，本研究で推定するべき確率モデルは $P(s)$，$P(c)$，$P(f_i|s)$，$P(f_i|c)$ である．それぞれの値は 4.1.2 項の上位概念を用いた Naive Bayes モデルのものと同様に求める．よって，詳しい説明は省略する．

$$P(s) = \frac{O(s) + \alpha}{\sum_s O(s) + \alpha V} \tag{4.26}$$

$$P(c) = \frac{O(c) + \alpha'}{\sum_c O(c) + \alpha' V'} \tag{4.27}$$

ただし，α' は全ての事象に足すべき頻度，V' は抽出した上位概念の総数を示している．ここでは $\alpha' = 0.1$ とした．両者の重みづけ β' は，調整用データにおいて値を変化させて，より精度の高かったものを選択することにする．他は 4.1.2 項で述べたものと同様である．

$$P(f_i|s) = \beta P_{MLE}(f_i|s) + (1 - \beta')P(f_i) \tag{4.28}$$

$$P_{MLE}(f_i|s) = \frac{O(f_i, c)}{\sum_{f_i} O(f_i, s)} \tag{4.29}$$

$$P(f_i|c) = \beta P_{MLE}(f_i|c) + (1 - \beta)P(f_i) \tag{4.30}$$

$$P_{MLE}(f_i|c) = \frac{O(f_i, c)}{\sum_{f_i} O(f_i, c)} \tag{4.31}$$

$$P(f_i) = \frac{O(f_i) + 1}{T + 2} \tag{4.32}$$

訓練コーパスから各素性を取り出した後，語義の頻度 $O(s)$，上位概念の頻度 $O(c)$，素性の頻度 $O(f_i)$，素性の頻度と語義の共起頻度 $O(f_i, s)$，素性の頻度と上位概念の共起頻度 $O(f_i, c)$ をカウントし，モデルを学習した．

第5章　評価実験

本章では，これまで述べた手法を実装し，語義曖昧性解消を行った結果について述べる．5.1 節において上位概念の抽出手法の評価実験結果を示し，5.2 節において語義曖昧性解消を行う分類器に関する実験結果を示す．

5.1　上位概念の抽出

3.2 節で述べた抽出パターンを用いて，岩波国語辞典にある辞書定義文から上位概念を抽出した．岩波国語辞典のデータフォーマットを図 5.1 に示す．

辞書定義文は形態素解析され，各形態素には 5.2 節で述べる RWC コーパスと同じ品詞コードが付与されている．図 5.1 の<mor>タグ内の pos="1"がそれにあたる．また，用例は

図 5.1: 岩波国語辞典

```
<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>
```

表 5.1: 岩波国語辞典の上位概念の抽出

	名詞	動詞	形容詞	計
単語数	51,197	4,165	668	56,885
語義数	66,342	8,841	1,256	76,439
抽出語義数	65,041	8,240	1,055	74,336
割合	0.9804	0.9320	0.8400	0.9725
上位概念の種類	16,690	3,148	642	20,480
平均語義数	3.8970	2.6175	1.6433	3.6297

特別なタグ<EX>で囲まれている．図 5.1 ではメタ文字 (<EX>) で表されている．上位概念の抽出には辞書定義文中の用例や文法情報の記述は用いないので，あらかじめ取り除いておく．

5.1.1 上位概念と抽出パターン

上位概念を抽出した結果を，表 5.1 に示す．本研究では，名詞，動詞，形容詞以外の品詞（副詞など）からは上位概念を抽出しない．上位概念抽出パターンを用いて，実際に上位概念を抽出できた語義の数を「抽出語義数」の列に示した．岩波国語辞典の全語義 76,439 のうち，74,336 語義 (97.25%) の上位概念の抽出に成功した．なお，抽出に失敗した辞書定義文はほとんど文法情報，用例など直接語義を表さないものである．ただし，3.3 節で述べたように，岩波国語辞典の辞書定義文には，1 つの語義に対して複数の定義文がある場合がある．この場合，その定義文のうち 1 つでも上位概念が抽出できたなら，その語義からは上位概念が抽出できたとした．また，各品詞ごとの上位概念の種類の数を集計し，それを語義数で割った平均語義数を求めた．この値が大きいほど，上位概念を用いることによって訓練データの数を擬似的に増やすことが出来るといえる．この結果から，上位概念を用いることで，全体で見ると訓練データの量を 3〜4 倍に増やす効果が得られることになる．ただし，3.3 節で述べたように，複数の定義文がある場合で上位概念が 1 つに決められない場合，どちらも上位概念の種類の数に加えているので，実際のモデルの学習における上位概念の平均語義数は表 5.1 の値よりも大きくなると考えられる．

次に，上位概念の抽出に関するデータを示す．表 5.2 は上位概念の抽出に用いられたパターンのうち，各品詞ごとに適用回数上位 10 位のパターンを示している．パターン名の後の数字はそのパターンが使われた回数である．これを見ると，各品詞のほとんどは単純に末尾の語を取り出したものであることが分かる．また，抽出された上位概念のうち，出現回数の多いもの 10 個を表 5.3 に示す．このように，上位概念を用いることで，多くの語義を同一視することにより，訓練事例が少ない語義についても，訓練データ量が増えてうまく学習が出来るようになる．

表 5.2: 上位概念抽出パターンの使用（適用回数上位 10 個）

順位	名詞	数	動詞	数	形容詞	数
1	n	62,086	v	7,281	a	640
2	n(V+形名)	18,375	v(N|A|ADV する)	1,173	a(N だ)	349
3	n(V)	7,925	s(括弧内)	574	a(V)	347
4	s(N の一つ)	2,045	v(複合動詞)	570	a(V|A 様子だ)	70
5	n(V た+形名)	1,797	v(V せる)	332	s(括弧内)	67
6	n(A+形名)	1,321	v(V 状態になる)	310	a(N な様子だ)	28
7	n(V|A|N ない+形名)	1,065	v(N になる)	291	a(N がある)	19
8	s(「N」の略)	1,029	v(N にする)	244	a(N がいい)	18
9	n(C+形名)	913	v(N サ変接続+をする)	225	a(N している)	16
10	n(A)	864	v(V れる)	162	a(N がする)	14

表 5.3: 抽出された上位概念（出現回数上位 10 個）

順位	名詞	数	動詞	数	形容詞	数
1	人	1,855	出す	93	ない	18
2	者	492	つける	87	悪い	17
3	部分	478	する	77	小さい	16
4	植物	338	行く	77	大きい	15
5	言葉	273	言う	76	高い	14
6	場所	225	出る	71	はなはだしい	13
7	道具	216	入れる	66	強い	12
8	家	189	つく	55	つらい	11
9	作る+こと	181	与える	54	多い	11
10	する+こと	180	思う	51	やすい	9

5.1.2 上位概念の有効性

ここからは抽出した上位概念が語義曖昧性解消に対してどれだけ有効であるかという観点で，抽出した上位概念を評価する．ある単語があった時，その全ての語義の辞書定義文から上位概念を抽出することができない場合，その語義に対しては上位概念を用いて学習が出来ないため，語義曖昧性解消にとって有効ではない．表 5.4 は，全ての語義について，上位概念の抽出に成功した単語の割合である．表 5.4 の「全部抽出」は，単語が複数の語義を持つとき，全ての語義から上位概念を抽出できた単語数である．「一部抽出」は一部の語義からのみ上位概念を抽出できた単語数である．「抽出失敗」は全ての語義について上位概念を取り出すことができなかった単語数である．ただし，複数の語義を持たな

表 5.4: 上位概念を抽出できた単語数

全部抽出	一部抽出	抽出失敗
8,386(0.6786)	3,037(0.2458)	934(0.0756)

表 5.5: 同じ上位概念が重複して抽出された単語数

重複なし	一部重複	全て重複
10,730(0.8683)	1,005(0.0813)	622(0.0503)

い単語はあらかじめ除いてある．

表 5.4 を見ると，全単語の 6 割以上が全ての語義で上位概念が抽出できているが，抽出に一部もしくは全部失敗する単語が 3 割を超えた．特に，一部しか抽出できない場合が多いが，これは岩波国語辞典ではよく用いられる一般的な単語 (例えば,「する」「行く」「つける」) などは 10 よりも多い語義を持っており，それらのうちには，品詞情報のみが書かれていたりする語義が存在する．そして，実際にコーパス内で用いられるような語義はほとんどが上位概念を抽出できていると思われる．したがって，実際に WSD を行うときには抽出が全て失敗した約 7.5%だけが問題であるとみなすと，それほど WSD の精度向上に影響を及ぼすことはないと思われる．

また，表 5.5 は複数の語義を持つ単語のうち，各語義の上位概念が重複しているかどうかをまとめたものである．同じ単語の語義から同じ上位概念が重複して抽出される場合は，上位概念を用いた Naive Bayes モデルで WSD を行うのが困難である．例えば，以下の「中子」という単語の 2 つの語義の定義文は，抽出される上位概念が同じである例である．〔〕内の数字は岩波国語辞典の語義 ID を表している．

【中子】〔38060-0-0-2-0〕 刀身の，つかの中にはいった 部分． → 部分
【中子】〔38060-0-0-3-0〕 果実の内部の柔らかい 部分． → 部分

この例のように意味が異なるのに上位概念が同じになる語義は，上位概念を用いたモデルでは上位概念と素性の共起情報を元に語義を判別しているので，同じ共起情報を用いることになり語義の判別が難しいという問題がある．

表 5.5 はこのような場合がどのくらい存在するのかを調べたもので,「重複なし」は全ての語義の上位概念が別のものである単語数,「一部重複」はいくつかの語義の上位概念が同じものである単語数,「全て重複」は全ての語義の上位概念が同じものである単語数を表している．表 5.5 を見ると，ほとんどの単語は語義の上位概念に重複はないが，1 割強の単語は一部，もしくは全体の語義に重複がある．しかし，語義が全て重複していなければ，ある文脈に現れる単語に該当する語義の数を絞り込む効果があるので，約 9 割 5 分の単語で上位概念が有効に働くと言える．

さらに，実際に上位概念を抽出した辞書定義文をランダムに 200 文取り出し，上位概念が適切なものであるかを人手で確認したところ，196 の辞書定義文から抽出した上位概念

表 5.6: 1 つの語義に対する複数の辞書定義文

	名詞	動詞	形容詞	計
第 1 文	65,756	8,342	1,123	75,221
第 2 文	31,216	3,086	470	34,790
第 3 文	9,264	850	129	10,254
第 4 文	2,739	216	26	2,982
第 5 文以降	1,231	110	8	1,349
非定義文	5,796	0	0	5,796
別語義定義文	4,530	420	33	4,983
同語義定義文	34,124	3,842	600	38,566

が適切であった．全体の 98%の上位概念が適切であり，モデルの学習にとって十分な精度であると思われる．

5.1.3 複数の定義文からの上位概念の選択

ここからは，1 つの語義に対して複数の定義文がある場合の上位概念の抽出結果について述べる．手法に関しては 3.3 節を参照のこと．

表 5.6 は，岩波国語辞典の全語義のうち，用例などを除いた定義文の数を示したものである．上 5 行は品詞ごとの 1 つの語義の文の数を表しており，例えば名詞は第 1 文があるのは 65,756 語義で，そのうち第 2 文もあるのは 31,216 語義であるという意味である．二重線の下の 3 行は，第 2 文以降の文において，3.3 節で述べたタイプ分類パターンによって分類された 3 つのタイプ,「非定義文」「別語義定義文」「同語義定義文」の数を表わす.

表 5.6 から，第 2 文以降の定義文のうちほとんどは同語義定義文であり，上位概念の抽出に用いない非定義文は全体の約 10%ほどに過ぎないということが分かる．

次に，表 5.7,5.8,5.9 は各品詞別のどの定義文の上位概念を選択したかを集計したものである.「第 1 語義」「第 2 語義」の区別では，第 2 文以降が別語義定義文である場合に，その文以降から抽出された上位概念を第 2 語義の上位概念として集計している.「1 つ目の文」「2 つ目の文以降」の区別では，その語義の定義文のうち 1 つ目の定義文の上位概念を選択した場合と，それ以外の定義文の上位概念から選択した場合を集計している.「1 つ目の文」が必ずしも岩波国語辞典における語釈文の先頭の文ではないことに注意していただきたい．別語義定義文により，先頭の文で表わされる語義とは別の語義が定義されているとき，その語義を定義する最初の文も語義を表わす「1 つ目の文」とみなす．なお，1 つ目の文と 2 つ目の文が同じ上位概念を抽出している場合は，1 つ目の文に加算している．具体的に，3.3 節で述べた例である，第 2 文が別語義定義文，第 3 文が同語義定義文である「解釈」について述べると,「理解+する+こと」は第 1 語義のうち 1 つ目の文から取り出

表 5.7: 複数の定義文の上位概念の選択率（名詞）

	1つ目の文	2つ目の文以降
第1語義	56,656(0.8702)	8,453(0.1298)
第2語義	4,372(0.9525)	218(0.0475)
計	61,028(0.8756)	8,671(0.1244)

表 5.8: 複数の定義文の上位概念の選択率（動詞）

	1つ目の文	2つ目の文以降
第1語義	7,336(0.8884)	922(0.1116)
第2語義	410(0.9624)	16(0.0376)
計	7,746(0.8920)	938(0.1080)

表 5.9: 複数の定義文の上位概念の選択率（形容詞）

	1つ目の文	2つ目の文以降
第1語義	912(0.8563)	153(0.1437)
第2語義	29(0.8788)	4(0.1212)
計	941(0.8570)	157(0.1430)

された上位概念,「説明+する+こと」は第2語義のうち1つ目の文から取り出された上位概念,「内容」は第2語義のうち2つ目の文以降から取り出された上位概念というようにして，実際に上位概念として選択されたものを表 5.7,5.8,5.9 で集計している．

これを見ると，上位概念が第2文以降から選択されている場合は全体の 10〜15%ほどだが，第1文しかない場合[1]を除くと，全ての品詞で約半分程度が第2文以降から選択されている．したがって，複数の定義文がある場合に，第1文だけでなくそれ以降の文からも上位概念を抽出することは，上位概念を用いた分類器に貢献しているといえる．

さらに，1つの語義に対して複数の辞書定義文が存在している語義の辞書定義文をランダムに 200 語義取り出し，その第2文以降の分類タイプが適切なものであるかを人手で確認した．その結果，187 語義の辞書定義文の分類タイプが全て適切であり，それは全体の 93.5%に相当するため，十分な分類精度であると思われる．なおタイプが適切でなかったもののほとんどが，非定義文で顕著に見られない単語が第1文の上位概念の場合であった．これを分類するには多くの単語をパターンに追加する必要がある．

[1]表 5.6 の第 1 文と第 2 文のある文の差により求められる．名詞は $65756 - 31216 = 34540$ 文．動詞は $8342 - 3086 = 5256$ 文．形容詞は $1123 - 470 = 653$ 文．

5.2 分類器の評価

本節では，本論文が提案する手法を用いて学習した分類器を用いて語義曖昧性解消の実験を行った結果を示す．まず，実験の手順を説明した後，SVM，Naive Bayes(以下，NBと略する) 単体の分類器の比較，ならびにSVMとNBを組み合わせた様々な手法の比較を行う．

5.2.1 単体分類器の評価

まず，本研究で用いた分類器において，単独で語義曖昧性解消を行った場合の精度について述べる．

本研究では，分類器を学習するためのコーパスとしてRWCコーパス[22]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，語義の他に形態素の情報やUDCコードが付与されている．しかしながら，本研究ではこれらの情報は用いず，新たに形態素解析器として茶筌[24]を，文節の係り受け解析器としてCabocha[25]を用いて得られる形態素情報，構文情報を利用した．表5.2にコーパスの一部分を掲載する．

実験では，RWCコーパスのうち1割に当たる300記事をテストデータ，同じく300記事を調整用データ，残りの2400記事を訓練データとした．まず調整用データに対してWSDを行い，NBモデルの式(4.17)のβと式(4.28)のβ'の値を変動させ，最適な値を求めた．この結果，$\beta = 0.9$，$\beta' = 0.4$となった．

本研究で用いた分類器は以下の3つである．

- SVM(Support Vector Machine)

  4.1.1項で述べたSVMを用いた手法.

- NB(Naive Bayes)

  4.1.2項で述べた，辞書定義文から取り出した上位概念を用いたNaive Bayesモデルによる分類器.

- BL(Baseline)

  最も出現頻度が高い語義を選択する分類器．但し，最頻出語義が複数ある場合にはその全てを答えとして選択している．

表5.10にそれぞれの分類器でWSDを行った結果を示す．表中の評価値を以下に示す.

- 精度$(P) = \dfrac{\text{正解数}}{\text{分類器が出力した語義数}}$

図 5.2: RWC コーパス

```
<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>
```

表 5.10: 単体分類器の比較

	精度	再現率	F 値	適用率	正解数
BL	76.71	76.61	76.66	98.71	10188
NB	78.67	**79.09**	78.88	**100**	**10518**
SVM	**84.77**	78.46	**81.50**	92.56	10433

表 5.11: 低頻度語を対象にした実験結果

頻度	0	1	2	3	4
評価単語数	226	159	112	108	88
平均 F 値 (NB)	0.4963	0.6714	0.7482	0.7315	0.6203
平均 F 値 (BL)	×	0.6906	0.7189	0.6893	0.6477

表 5.12: 中頻度語を対象にした実験結果

頻度	5	6	7	8	9
評価単語数	87	70	54	54	60
平均 F 値 (NB)	0.7808	0.7381	0.7531	0.7855	0.8083
平均 F 値 (BL)	0.7222	0.6498	0.6564	0.7083	0.7628
頻度	10	11	12	13	14
評価単語数	52	63	43	22	51
平均 F 値 (NB)	0.7949	0.7124	0.8256	0.8034	0.7729
平均 F 値 (BL)	0.7196	0.6658	0.7178	0.9091	0.7667

- 再現率 $(R) = \dfrac{\text{正解数}}{\text{テストデータに含まれる全単語数}}$
- F 値 $= \dfrac{2PR}{P+R}$
- 適用率 $= \dfrac{\text{分類器が正解，不正解に関わらず語義を出力した単語数}}{\text{テストデータに含まれる全単語数}}$
- 正解数 = 分類器が正解の語義を出力した単語数

表 5.10 から，SVM は，頻度 5 未満の低頻度語については語義の判別を行わないため，再現率は低いものの，精度と F 値は高かった．本研究の提案手法である NB モデルは，SVM には及ばなかったものの，ベースラインモデルと比べて F 値で約 3%の上昇が見られた．正解数でも単体の分類器の中では最も良い値を示した．

表 5.11 は低頻度語におけるベースラインモデルと NB モデルの比較を行った結果であり，表 5.12 は中頻度語におけるベースラインモデルと NB モデルの比較を行った結果であ

表 5.13: 混合モデルの頻度別比較

頻度	0	1〜4	5〜
SVM+NB (頻度)	NB	NB	SVM
SVM+NB (正解含有率)	NB	NB	NB or SVM
SVM+NB (スタッキング)	NB	NB	スタッキング
SVM+BL	×	BL	SVM

る．この結果を見ると，全体的にはNBモデルがベースラインモデルよりも良い結果を示したが，思っていたほどの差は得られなかった．むしろ，表5.10で見られる両モデルの差は，中頻度で現れていることが分かった．低頻度語でNBモデルが思ったほど精度が良くなかった理由として，表3.2を見れば分かるように，上位概念を用いても低頻度の上位概念が多く存在し，低頻度語を思ったほど減らすことが出来なかったことが考えられる．

5.2.2 混合分類器の評価

次に，分類器の組み合わせ手法の実験結果を示す．

SVMとNBの頻度による混合モデルは，低頻度である頻度5未満の単語にはNBモデルを，それ以外の高頻度語はSVMモデルを用いた．正解含有率による混合モデルは，低頻度である頻度5未満の単語にはNBモデルを，それ以外の高頻度語に関しては，調整用データから求めた単語の正解含有率の高い方の分類器を用いた．ただし，調整用データにおいて頻度が10以下の単語に関しては，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比較する．詳しくは4.2節で述べた．

スタッキングによる手法は，交差検定を用いる手法と用いない手法を行ったが，交差検定を用いない，本来のスタッキングの手法に即した手法では，まず訓練データ2400記事により訓練された各分類器に，訓練データに調整用データを加えた2700記事の語義を判定させ，その判定結果を素性に加えた2700記事のデータをSVMで訓練し，テストデータの語義を判定させることにより評価した．次に，交差検定を用いる手法では，各単語ごとに訓練データと調整用データを合わせた2700記事を5分割し，それらの1つをテストデータ，4つを訓練データとしたSVM分類器を5回作成し，それぞれのテストデータの語義判別結果を求めて1次分類器の素性とした．NB分類器はSVMと違い単語ごとに分類器を作成するわけではないので，2700記事の通常の9-fold交差検定を行い，素性に加えた．それから先は交差検定を用いない手法と同様である．また，スコアを用いるスタッキングも，交差検定を用いる手法と同様な手法で行った．

その他に，混合分類器の比較のために，SVMモデルとベースラインモデルを組み合わせる．SVMはそのモデルの性質上，単語ごとに分類器を訓練する．よって，種類の多い低頻度語の学習は大量の分類器が必要となりコストがかかる上，精度も明らかに落ちる．

表 5.14: SVM と NB の混合分類器の比較

	精度	再現率	F 値	適用率	正解数
頻度	83.35	83.73	83.54	**100**	11134
正解含有率	83.55	83.92	83.73	**100**	11160
スタッキング (シンプル)	83.70	84.08	83.89	99.76	11154
スタッキング (交差検定)	**84.23**	**84.61**	**84.42**	99.76	**11225**
スタッキング (スコア)	84.02	84.41	84.21	99.05	11118

そこで，本研究では訓練データにおける頻度が 5 未満の単語に関しては，SVM モデルにおける学習を行っていない．混合分類器において，SVM モデルに NB モデルを組み合わせることによる効果を示すために，SVM モデルで低頻度語の語義選択を行う代わりに，低頻度語に対してベースラインモデルを用いた混合モデルと比較する．その理由は，SVM などの教師あり学習のモデルでは，訓練データにおける頻度が低い単語に対しては語義と素性の共起情報が少ないため，語義の頻度自体が高い語義が選択されやすく，ベースラインモデルと近い語義選択を行うと考えられるからである．SVM とベースラインモデルの具体的な組み合わせ方は，SVM と NB の頻度による混合モデルとほぼ同じである．但し，BL モデルは最も出現頻度が高い語義を選択するモデルであるから，頻度が 0 である単語に対しては分類を行わない点が異なる．

表 5.13 は，以上の分類器の組み合わせ手法について，訓練データにおける単語の頻度別にどの分類器を選択するかをまとめたものである．

表 5.14 はこれまで述べた様々な手法で SVM モデルと NB モデルを組み合わせた結果をまとめたものである．表 5.14 の結果から最も F 値が高かったのは交差検定を行ったスタッキングのモデルであったことが分かる．スタッキングの手法では若干適用率が落ちているが，それは素性に分類器の出力を加える過程で，訓練データの頻度が低いために分類器が作られなかった場合などで，結果的にスタッキングの 2 次分類器の学習データ量が少なくなったものと思われる．

次に，分類器の組み合わせ手法の比較を行う．表 5.15 は分類器の混合モデルを比較した結果をまとめたものである．各モデルは以下の通りである．

- SVM+BL

  訓練データにおける頻度が 5 未満の低頻度語を BL，それ以外の高頻度語を SVM で WSD を行う．

- SVM+NB

  表 5.14 で示した SVM と NB を組み合わせた手法のうち，最も F 値が高かった手法 (スタッキング (交差検定)) で WSD を行う．

- $\mathrm{NB}_{s,c}$

表 5.15: 混合モデル全体の比較

	精度	再現率	F 値	適用率	正解数
SVM+BL	83.72	82.92	83.32	98.73	11194
SVM+NB	**84.23**	**84.61**	**84.42**	99.76	**11225**
$NB_{s,c}$	80.09	80.46	80.27	**100**	10699

表 5.16: 改良抽出法による上位概念を用いた NB モデルの比較

T_l	T_h	T_d	精度	再現率	F 値	適用率	正解数
5	10	1	78.58	78.99	78.78	**100**	10504
		2	78.60	79.01	78.80	**100**	10507
		ALL	78.55	78.97	78.76	**100**	10501
10	10	1	78.67	79.09	78.88	**100**	10517
		2	78.67	79.09	78.88	**100**	10517
		ALL	78.65	79.07	78.86	**100**	10515
改良前のモデル			**78.67**	**79.09**	**78.88**	**100**	**10518**

4.3 節で提案した，語義と素性の共起情報と，上位概念と素性の共起情報を同時に用いるモデルで WSD を行う．

表 5.15 の結果を見ると，SVM と NB を組み合わせた手法が最もよく，SVM と BL を組み合わせた手法は，表 5.14 の頻度による組み合わせ手法と比べても若干ではあるが F 値が劣る．また，語義と素性の共起情報と，上位概念と素性の共起情報を同時に用いるモデルは，SVM と BL を組み合わせた手法よりも全ての値が下回り，良い結果は得られなかった．原因としてはモデル自体の妥当性の低さも考えられるが，Naive Bayes モデル自体が SVM のモデルに比べてかなり精度が低く，Naive Bayes モデルに改良を加えたこの手法では SVM の精度を上回るのは難しいのではないかと思われる．

5.2.3 改良抽出法による上位概念の評価

最後に，3.4 節で述べた改良抽出法による上位概念を用いた Naive Bayes モデルの比較を行った実験結果を示す．

表 5.16 は閾値を変化させて実験した結果をまとめたものである．表中の T_l の値は，改良抽出法を適用する対象となる低頻度の上位概念の範囲を表しており，T_l の値未満の上位概念について改良抽出法を適用する．T_h の値は，上位概念の置き換えを行う際に，低頻度の上位概念を置き換える高頻度の上位概念の頻度の上限を表しており，T_h の値以上の上位概念が見つかればそれに置き換える．T_d の値は，改良抽出法を適用する際に，上位

概念を何文字まで削ることにより上位概念の置き換えを行うかを表しており，T_d 文字まで削ることを許す．つまり，T_d が 1 の場合は，1 文字削った文字列が高頻度の上位概念にマッチしなければ，そこでその上位概念に関しては置換せずに終了するが，T_d が 2 の場合は，1 文字削った文字列が高頻度の上位概念にマッチしなければ，2 文字削った場合も試し，T_d が ALL の場合は，元の上位概念が最後の 1 文字となるまで続けて行う．

この結果，全ての結果が改良前のモデルとほとんど変化がなく，しかも精度が悪いという結果であった．元のモデルの結果と比較したところ，元のモデルで不正解であった語義を正解している場合も 50〜100 例ほどあるのだが，それと同等，もしくは多い数で，元のモデルで正解であった語義を間違っている場合があり，結果として全体の精度が落ちていた．結果として，本研究で行った上位概念の改良の手法では学習モデルの精度を向上させることはできなかった．今後の課題として，改良の手法を洗練することと，最適な閾値を決めることが必要であると思われる．

第6章　おわりに

本論文では，WSD におけるデータの過疎性の問題に対処するために，国語辞典から語義の上位概念を抽出する手法を述べた．さらに，高頻度語に有効な教師あり学習による分類器と低頻度語に有効な上位概念を用いた分類器を組み合わせる様々な手法を実験的に比較した．

最後に，今後の課題を 4 つ挙げる．

1. 他の国語辞書への上位概念抽出パターンの応用
   本論文では，上位概念を抽出するための国語辞書として，岩波国語辞典を用いた．上位概念を抽出するパターンは岩波国語辞典の定義文を見て作成したが，これを他の一般の国語辞書に用いることが出来れば，どの辞書でも上位概念を抽出でき，その語義を読解支援システムなどに用いることが出来るようになる．今回用いた上位概念抽出パターンが岩波国語辞典に特化したものなのか，それとも他の国語辞書に対してもいくらか適用できるのかを検討したい．
2. 分類器の組み合わせ手法の検討
   本論文では，SVM+NB との組み合わせを様々な手法で行い実験を行った．結果はスタッキングを用いた手法が最も精度，再現率ともに良かったが，単純な頻度による組み合わせと比較してそれほど向上したわけではなく，さらに向上する余地があると思われる．実際に各単体分類器の結果から，Naive Bayes モデルの正解と SVM モデルの正解を必ず選択する理想的な組み合わせが実現したら，本論文の最良の手法の F 値よりも 3%ほど向上する．それに少しでも近づく組み合わせ手法を検討する余地がある．
3. 教師無し学習との比較
   本研究は辞書の上位概念を用いることにより，低頻度語の曖昧性解消の精度と再現率を向上させた．この手法とは別に，語義タグ付きコーパスを必要としない教師無し学習での手法を用いることで，低頻度語の曖昧性解消を行う手法もある．よって，これらの手法と本手法の比較を行うべきである．
4. 上位概念の抽出法の洗練
   本研究では，上位概念の抽出に関してはパターンを増やすことにより，ほとんどの定義文に対して上位概念を抽出することに成功したが，上位概念にしても低頻度のままであり，訓練例を増やすことが出来ない語義が多かった．このため，上位概念をただ取り出すだけではなく，学習に役立つ上位概念になるように抽出法の改良を試みたが結果的に上手く行かなかった．よって，上位概念の抽出法の洗練をさらに検討する必

要がある.

謝辞

本研究を行うにあたり，終始暖かいご指導を賜りました白井清昭助教授に深く感謝いたします．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，中村誠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．

参考文献

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

[2] Seong-Bae Park, Byoung-Tak Zhang and Yung Taek Kim. Word Sense Disambiguation by Learning Decision Trees from Unlabeled Data. Proceedings of Applied Intelligence 19, pp.27-38, 2003.

[3] 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.

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

[5] 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.

[6] Adam Kilgarriff and J. Rosenzweig. Framework and results for English Senseval. Computers and the Humanities. 34(1):15-48,2000.

[7] Ted Pedersen. A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. In the Proceedings of the NAACL-00, 2000.

[8] Veronique Hoste, I. Hendrickx, W. Daelemans and A. van den Bosch. Parameter optimization for machine-learning of word sense disambiguation. Natural Language Engineering,8(3), 2002.

[9] Radu Florian, Silviu Cucerzan, C Schafer and D. Yarowsky. Combining Classifiers for Word Sense Disambiguation. Journal of Natural Language Engineering. Vol. 8 No.4, 2002.

[10] Radu Florian and D. Yarowsky. Modeling Consensus: Classifier Combination for Word Sense Disambiguation. In Proceedings of EMNLP'02, pp25-32, 2002.

[11] Dan Klein, K. Toutanova, H. Tolga Ilhan, S. D. Kamvar and C. D. Manning. Combining Heterogeneous Classifiers for Word-Sense Disambiguation. In Workshop on Word Sense Disambiguation at ACL 40, pages 74-80, 2002.

[12] Eibe Frank, M. Hall and Bernhard Pfahringer. Locally Weighted Naive Bayes. Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2003.

[13] Xiaojie Wang, Yuji Matsumoto. Trajectory Based Word Sense Disambiguation. Proceedings of COLING pp.903-909, 2004.

[14] J. Ross Quinlan. C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[15] Ronald L. Rivest. Learning decision lists. Machine Learning, Vol. 2, pp. 229.246, 1987.

[16] Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. Computational Linguistics, 22(1):39-71, 1996.

[17] Mitchell, T. Machine Learning. Mc-Graw Hill, 1997.

[18] David H. Wolpert. Stacked Generalization. Neural Networks, v.5 n.2, pp.241-259, 1992.

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

[20] 日本電子化辞書研究所. EDR 電子化辞書仕様説明書第２版
Technical Report TR-405,1995

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

[22] 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.

[23] 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.

[24] 松本裕治、北内啓、山下建雄、平野喜隆、松田寛、高岡一馬、浅原正幸、茶筌 version 2.3.3 使用説明書,2003

[25] 日本語係り受け解析システム「南瓜」 マルチメディア言語学情報 [18], 月刊 言語, Vol.32, No.6, pp.74-75, June 2003.

[26] 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.

[27] 新納浩幸. 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.

[28] 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.

[29] 八木恒和. EDR 概念辞書とコーパスを用いた語義曖昧性解消に関する研究. 修士論文, 2004.

[30] 玉垣隆幸. 辞書の語義立てに基づく語義曖昧性解消に関する研究. 修士論文, 2004.

[31] 白井清昭, 八木恒和. 辞書定義文を用いた低頻度語のための語義曖昧性解消モデルの学習. 情報処理学会自然言語処理研究会 (NL-158-20), pp.127-132, 2003.

[32] 白井清昭, 八木恒和. コーパスと辞書定義文中の上位概念を用いた頑健な語義曖昧性解消. 言語処理学会第 10 回年次大会, pp.745-748, 2004.

[33] Kiyoaki Shirai and Tsunekazu Yagi. Learning a Robust Word Sense Disambiguation Model using Hypernyms in Definition Sentences. 20th International Conference on Computational Linguistics, pp. 917-923, 2004.

付 録A　上位概念抽出パターン

品詞別の上位概念抽出パターンの一覧を示す．パターンの詳しい見方は 3.2 節を参照のこと．

なお，抽出パターンの適用順序はここで挙げた抽出パターンの順序と一致する．ただし，接尾語の抽出パターンを先に適用し，マッチしなかったときにそれぞれの品詞のパターンを適用する．

A.1　名詞の抽出パターン

名詞から上位概念を抽出するパターンの一覧を以下に示す．

```
n(N のこと)
名詞
[o(連体詞),o(形名),o(名詞)] <の|に関する|にわたる|に対する|に当たる><助詞><格助詞> <こと|もの><名詞><*>
o:1

n(V ということ)
名詞
[o(動基 3)] <という><助詞><格助詞> <こと|もの><名詞><*>
o:1 +こと

n(A ということ)
名詞
[a(N である),a] <という><助詞><格助詞> <こと|もの><名詞><*>
o:1

n(連体詞+こと)
名詞
<*><連体詞><*> <こと|もの><名詞><*>
h:1 + b:2

n(V ている+形名)
名詞
[o(動基 2)] <て><助詞><接続助詞> <いる><動詞><非自立> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 + b:4

n(V ていない+形名)
名詞
[o(動基 2)] <て><助詞><接続助詞> <いる><動詞><非自立> <ない><助動詞><*> <こと|もの|さま|物|様
```

子|状態|所|ところ|程度><名詞><*>
o:1 + b:5

n(V|A|N ない+形名)
名詞
[o(動基 2),o(形容詞),n] <ない|ぬ><助動詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 +否定+ b:3

n(V た+形名)
名詞
[o(動基 3)] <た|だ><助動詞><特殊型> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 +完了+ b:3

n(V べき+形名)
名詞
[o(動基 3)] <べき><助動詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 + b:3

n(V べきである+形名)
名詞
[o(動基 3)] <べき><助動詞><*> <だ><助動詞><*> <ある|ない><助動詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 + b:5

n(V ほどである+形名)
名詞
[o(動基 3)] <ほど><助動詞><*> <だ><助動詞><*> <ある|ない><助動詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 + b:5

n(N である+形名)
名詞
[o(形名),o(名詞)] <だ><助動詞><*> <ある|ない><助動詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 +コピュラ+ b:4

n(N ではない+形名)
名詞
[o(形名),o(名詞)] <だ><助動詞><*> <は|も><助詞><*> <ない><助動詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 +コピュラ+否定+ b:5

n(N にしたもの)
名詞
[o(名詞)] <に|を|と><助詞><*> <し><動詞><サ変・スル> <た><助動詞><特殊型> <もの><名詞><*>
o:1

n(V+形名)
名詞
[o(動基 3)] <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
o:1 + b:2

```
n(A+形名)
名詞
[o(形容詞)] <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
oh:1 + b:2

n(C+形名)
名詞
<*><名詞><*> <の|な><助詞><*> <こと|もの|さま|物|様子|状態|所|ところ|程度><名詞><*>
h:1 +コピュラ+ b:3

n(V ような感じ)
名詞
[o(動基)] <よう><名詞><非自立> <な><助詞><連体化> <感じ><名詞><*>
o:1 + h:2 + h:3 + h:4

n(A 感じ)
名詞
<*><形容詞><*> <感じ><名詞><*>
h:1 + h:2

n(A さ)
名詞
<*><形容詞><*> <さ|み><名詞><接尾>
h:1 + b:2

n(V 方)
名詞
<*><動詞><*> <方><名詞><接尾>
h:1 + b:2

n(N の方)
名詞
[o(名詞)] <の><助詞><*> <方><名詞><接尾>
o:1 +の方

n(N 的)
名詞
[o(名詞)] <的><名詞><接尾>
o:1 + h:2

n(複合名詞)
名詞
<*><名詞><*>* <物|法><名詞><*>
h:1 + b:2

n(V)
名詞
[o(動基 3)]
o:1 +こと

n(A)
名詞
[a]
```

```
o:1 +さま

n
名詞
<*><名詞><*>
b:1
```

A.2 動詞の抽出パターン

動詞から上位概念を抽出するパターンの一覧を以下に示す.

```
v(V 状態になる)
動詞
[o(動基 2)] <状態|もの|よう><名詞><*> <に><助詞><*> <する|なる><動詞><*>
o:1

v(V た状態になる)
動詞
[o(動基2)] <た|だ><助動詞><特殊型> <状態|もの|よう><名詞><*> <に><助詞><*> <する|なる><動詞><*>
o:1

v(N 状態になる)
動詞
[o(名詞)] <状態><名詞><*> <に><助詞><格助詞> <なる|する><動詞><*>
o:1 +する

v(N のようになる)
動詞
[o(名詞)] <の|な><助詞><*> <よう><名詞><非自立> <に><助詞><*> <なる|する><動詞><*>
o:1 + h:2 + h:3 + h:4 + h:5

v(A 状態になる)
動詞
<*><形容詞><*> <状態><名詞><*> <に><助詞><格助詞> <なる><動詞><*>
h:1 + h:2 + h:3 + h:4

v(V 状態にある)
動詞
[o(動基 2)] <状態><名詞><*> <に><助詞><格助詞> <ある><動詞><*>
o:1

v(A 状態にある)
動詞
<*><形容詞><*> <状態><名詞><*> <に><助詞><格助詞> <ある><動詞><*>
h:1 + h:2 + h:3 + h:4

v(V 状態である)
動詞
[o(動基 2)] <状態><名詞><*> <で><助動詞><特殊型> <ある><助動詞><*>
o:1

v(V ようとする)
```

```
動詞
[o(動基 2)] <う|よう><助動詞><*> <と><助詞><格助詞> <する><動詞><*>
o:1

v(V たりする)
動詞
[o(動基 3)] <たり|だり><助詞><接続助詞> <する><動詞><*>
o:1

v(V なくなる)
動詞
[o(動基)] <ない><助動詞><形容詞型> <なる|する><動詞><*>
o:1 +否定

v(A なくなる)
動詞
<*>形容詞><*> <ない><助動詞><形容詞型> <なる|する><動詞><*>
b:1 +否定

v(N でなくなる)
動詞
[o(名詞)] <で><助動詞><特殊型> <ない><助動詞><形容詞型> <なる><動詞><*>
o:1 +否定

v(N になる)
動詞
<*><名詞><*> <に|と|が><助詞><*> <なる><動詞><*>
h:1 +になる

v(V になる)
動詞
<*><動詞><*> <に><助詞><*> <なる><動詞><*>
h:1 +になる

v(N なる)
動詞
[o(名詞)] <なる><動詞><*>
oh:1 +なる

v(A なる)
動詞
[o(形容詞)] <なる><動詞><*>
oh:1 +なる

v(V てある)
動詞
[o(動基 3)] <て><助詞><*> <ある|いる><動詞><*>
oh:1 + b:2 + h:3

v(N にある)
動詞
[o(名詞)] <に><助詞><*> <ある><動詞><*>
o:1 + b:2 + h:3
```

```
v(V ことができる)
動詞
[o(動基)] <こと><名詞><非自立> <が><助詞><格助詞> <できる><動詞><*>
o:1 +可能

v(N できる)
動詞
[o(形名),o(名詞)] <できる><動詞><*>
o:1 +可能

v(V て+非自立動詞１)
動詞
[o(動基 2)] <て><助詞><接続助詞> <いる|いく|おく|しまう|ゆく><動詞><*>
o:1

v(V で+非自立動詞１)
動詞
<*><動詞><*> <で><助詞><接続助詞> <いる|いく|おく|しまう|ゆく><動詞><*>
b:1

v(V+非自立動詞２)
動詞
[o(動基 2)] <はじめる|始める|続ける|過ぎる|合う|かける|終わる|終える><動詞><*>
o:1

v(V れる)
動詞
[o(動基)] <れる|られる><動詞><*>
o:1 +受身

v(V ようにさせる)
動詞
[o(動基 2)] <よう><名詞><非自立> <に><助詞><格助詞> <する><動詞><*> <せる><動詞><*>
o:1 +使役

v(V せる)
動詞
[o(動基)] <せる><動詞><*>
o:1 +使役

v(N|A|ADV する)
動詞
[o(名詞),a,o(ADV)] <する><動詞><*>
oh:1 +する

v(V 連用+する)
動詞
<*><動詞><*> <する><動詞><*>
b:1

v(N サ変接続+をする)
動詞
```

```
<*><名詞><サ変接続> <を><助詞><格助詞> <する><動詞><*>
h:1 +する

v(V をする)
動詞
<*><動詞><*> <を><助詞><格助詞> <する><動詞><*>
b:1

v(N をする)
動詞
[o(形名),o(名詞)] <を><助詞><格助詞> <する><動詞><*>
o:1 +する

v(N とする)
動詞
[o(形名),o(名詞)] <と><助詞><格助詞> <する><動詞><*>
o:1 +とする

v(N する)
動詞
[o(名詞)] <する><動詞><*>
o:1 +する

v(V させる)
動詞
<*><動詞><*> <させる><動詞><*>
h:1 + h:2

v(N がある)
動詞
[o(名詞)] <が|の><助詞><格助詞> <ある|ない|する><*><*>
o:1 +が+ b:3

v(N にする)
動詞
[o(形名),n] <に><助詞><*> <する><動詞><*>
o:1 +にする

v(V ない)
動詞
[o(動基 3)] <ない><助動詞><*>
o:1 +否定

v(N めく)
動詞
<*><名詞><*> <めく><動詞><*>
h:1 + b:2

v(N だ)
動詞
[o(形名),o(名詞)] <だ><助動詞><*>
o:1 +コピュラ
```

v(N である)
動詞
[o(形名),o(名詞)] <で><助詞><*> <ある><助動詞><*>
o:1 +コピュラ

v(複合動詞)
動詞
<*><動詞><*> <*><動詞><*>
b:1

v
動詞
<*><動詞><*>
b:1

A.3 形容詞の抽出パターン

形容詞から上位概念を抽出するパターンの一覧を以下に示す.
a(V|A 様子だ)
形容詞
[o(動基 3),o(形容詞)] <様子|感じ|気持|状態|有様|態度><名詞><*> <だ><助動詞><*>
o:1 + b:2 +コピュラ+さま

a(V|A な様子だ)
形容詞
[o(動基 3),o(形容詞)] <た|な><助動詞><*> <様子|感じ|気持|状態|有様|態度><名詞><*> <だ><助動詞><*>
o:1 + h:2 + b:3 +コピュラ+さま

a(N な様子だ)
形容詞
<*><名詞><*> <な|の><助詞><*> <様子|感じ|気持|状態|有様|態度><名詞><*> <だ><助動詞><*>
b:1 + h:2 + b:3 +コピュラ+さま

a(V|A ような様子だ)
形容詞
[o(動基3),o(形容詞)] <よう><名詞><非自立> <な><助詞><連体化> <様子|感じ|気持|状態|有様|態度><名詞><*> <だ><助動詞><*>
o:1 + b:2 + b:3 +コピュラ+さま

a(V そうだ)
形容詞
<*><動詞><*> <そう><名詞><接尾> <だ><助動詞><特殊型>
h:1 + h:2 +コピュラ+さま

a(N だ)
形容詞
[o(形名),o(名詞)] <だ><助動詞><*>
o:1 +コピュラ+さま

a(N である)

形容詞
[o(形名),o(名詞)] <で><助詞><*> <ある><助動詞><*>
o:1 +コピュラ+さま

a(N がある)
形容詞
<*><名詞><*> <が><助詞><*> <ある><動詞><*>
b:1 + h:2 + h:3 +さま

a(N がいい)
形容詞
<*><名詞><*> <が><助詞><*> <いい|よい><形容詞><*>
b:1 + h:2 + h:3 +さま

a(N らしい)
形容詞
<*><名詞><*> <らしい><助動詞><*>
b:1 + h:2 +さま

a(V にくい)
形容詞
[o(動基 3)] <にくい><形容詞><*>
o:1 + h:2 +さま

a(N な感じがする)
形容詞
[o(名詞)] <な><助詞><*> <*><名詞><*> <が|と><助詞><*> <する><動詞><*>
o:1 + b:2 + b:3 + b:4 + b:5 +さま

a(N がする)
形容詞
<*><名詞><*> <が|と><助詞><*> <する><動詞><*>
b:1 + b:2 + b:3 +さま

a(V たい)
形容詞
[o(動基 3)] <たい><形容詞><*>
o:1 + b:2

a(N としている)
形容詞
<*><名詞><*> <と><助詞><*> <する><動詞><*> <て><助詞><接続助詞> <いる><動詞><*>
b:1 +コピュラ+さま

a(N している)
形容詞
<*><名詞><*> <する><動詞><*> <て><助詞><接続助詞> <いる><動詞><*>
b:1 +コピュラ+さま

a(V)
形容詞
[o(動基 3)]
o:1 +さま

a(V+さま)
形容詞
[o(動基 3)] <さま><名詞><非自立>
o:1 + h:2

a
形容詞
<*><形容詞><*>
b:1

A.4 接尾語の抽出パターン

接尾語から上位概念を抽出するパターンの一覧を以下に示す.
s(「N」の略)
接尾語
[o(名詞)] <」><*><*> <の><助詞><格助詞> <略><名詞><*>
o:1

s(N の一つ)
接尾語
[o(形名),o(名詞)] <の><助詞><格助詞> <一つ|一種|総称|敬称|類|名|称|俗称|名称><名詞><*>
o:1

s(N の単位)
接尾語
[o(名詞)] <の><助詞><格助詞> <単位><名詞><*>
o:1 +単位

s(A さを表す単位)
接尾語
<*><形容詞><*> <さ><名詞><接尾> <を><助詞><格助詞> <表す|あらわす><動詞><*> <単位><名詞><*>
h:1 + b:2 +単位

s(N を表す単位)
接尾語
[o(名詞)] <を><助詞><格助詞> <表す|あらわす><動詞><*> <単位><名詞><*>
o:1 +単位

s(N をはかる単位)
接尾語
<*><名詞><*> <を><助詞><格助詞> <はかる><動詞><*> <単位><名詞><*>
b:1 +単位

s(N 単位)
接尾語
<*><名詞><*> <単位><名詞><*>
b:1 +単位

s(N を V て言う語)
接尾語

[o(形名),o(名詞)] <を|に><助詞><格助詞> <*><動詞><*> <て><助詞><*> <言う|いう><動詞><*> <語><名詞><*>
o:1

s(「N」を V て言う語)
接尾語
[o(形名),o(名詞)] <」><*><*> <を|に><助詞><格助詞> <*><動詞><*> <て><助詞><*> <言う|いう><動詞><*> <語><名詞><*>
o:1

s(N の意を表す語)
接尾語
<*><名詞><*> <の|に当たる><助詞><格助詞> <意><名詞><*> <を><助詞><格助詞> <表す><動詞><*> <語><名詞><*>
b:1

s(N を表す語)
接尾語
[o(形名),o(名詞)] <を><助詞><格助詞> <表す|示す><動詞><*> <語><名詞><*>
o:1

s(N を言う語)
接尾語
[o(形名),o(名詞)] <を|に><助詞><*> <言う><動詞><*> <語><名詞><*>
o:1

s(N|V の尊敬語)
接尾語
[o(形名),o(名詞)] <の><助詞><格助詞> <尊敬><名詞><*> <語><名詞><*>
o:1

s(V 語)
接尾語
<*><動詞><*> <語><名詞><*>
h:1 + h:2

s(V たばかり)
接尾語
[o(動基 3)] <た><助動詞><*> <ばかり|だけ><助詞><*>
oh:1 + h:2 + h:3

s(N ばかり)
接尾語
<*><名詞><*> <ばかり|だけ><助詞><*>
h:1 + h:2

s(N など)
接尾語
<*><名詞><*> <など><*><*>
h:1

s(「N」)
接尾語

```
<「><*><*> <*><名詞><*> <」><*><*>
b:2
```

A.5 その他のパターン

その他における上位概念を抽出するパターンを以下に示す．これらは，品詞のパタンマッチには使わないが，他のパタンマッチ規則の部品として使う．

```
o(名詞)
その他
[s(「N」),s(N など),s(N ばかり),s(V たばかり),n(連体詞+こと),n(N のこと),n(V|A ということ),n(N にしたもの),n(A さ),n(V 方),n(N の方),n(N 的),n(複合名詞),n,o(ADV),o(V)]
oh:1

o(形名)
その他
[n(V ている+形名),n(V ていない+形名),n(V|A|N ない+形名),n(V た+形名),n(N である+形名),n(V+形名),n(A+形名),n(C+形名)]
oh:1

o(形容詞)
その他
[a(V そうだ),a(N だ),a(N である),a(N がある),a(N がいい),a(N らしい),a(V にくい),a(N がする),a(V たい),a(N としている),a(N している),a(V),a]
o:1

o(連体詞)
その他
<*><連体詞><*>
b:1

o(ADV)
その他
<*><副詞><*>
b:1

o(V)
その他
<*><動詞><自立>
b:1

# 「動詞基本パタン 2」 + アルファ（動詞全て）
o(動基 3)
その他
[v(V+非自立動詞２),v(V で+非自立動詞１),v(V て+非自立動詞１),v(N でなくなる),v(V なくなる),v(V たりする),v(V ようとする),v(V 状態である),v(V 状態にある),v(A 状態にある),v(A 状態になる),v(N 状態になる),v(V 状態になる),v(V た状態になる),v(N のようになる),o(動基 2)]
o:1

# 「動詞基本パタン」 + 受身形，使役形，可能形，〜なる，〜ある
o(動基 2)
```

その他
[v(V てある),v(N にある),v(V せる),v(V ようにさせる),v(V れる),v(N がある),v(V れる),v(A なる),v(N になる),v(N できる),v(V ことができる),o(動基)]
o:1

o(動基)
その他
[v(N である),v(N だ),v(V ない),v(N めく),v(N にする),v(N する),v(N とする),v(N をする),v(V をする),v(N サ変接続+をする),v(V 連用+する),v(N|A|ADV する),v(複合動詞),v]
o:1

付 録B　複数の定義文のタイプ分類パターン

第 2 文以降の文に以下のような分類のパターンを適用し，複数の定義文の上位概念を決定する．この並び順に上から適用される．また，以下のパターンが当てはまらない定義文は同語義定義文と分類される．詳しくは 3.3 節を参照のこと．

B.1　非定義文分類パターン

- 第 1 文から抽出した上位概念が次のいずれかに当てはまるとき,

  その語義の第 2 文以降は全て非定義文.

  「動物」,「植物」,「昆虫」,「魚」,「高木」,「低木」,「鳥」,「神」,「つる草」,「器官」,「調味料」,「淡水魚」,「油」,「海魚」

- 第 1 文の定義文の文末の単語の表記が次のいずれかに当てはまるとき,

  その語義の第 2 文以降は全て非定義文.

  「一つ」,「一種」,「名称」,「総称」,「単位」,「俗称」,「称」,「敬称」,「名」,「類」

- 第 2 文以降の定義文の文頭の単語の表記が「例」であるとき,

  その語義の次の文以降は全て非定義文.

- 第 2 文以降の定義文の文末の単語の表記が次のいずれかに当てはまるとき,

  その語義の次の文以降は全て非定義文.

  「使う」,「つかう」,「言う」,「いう」,「用」,「用いる」

- 第 2 文以降から抽出した上位概念が次のいずれかに当てはまるとき,

  その語義の次の文以降は全て非定義文.

  「読む+こと」,「多い+さま」,「通ずる+こと」,「ある+こと」

B.2　別語義定義文分類パターン

- 第 2 文以降の定義文の文頭の単語の表記が次のいずれかに当てはまるとき,
  その定義文は別語義定義文.
  「また」,「転じ」,「もと」,「比ゆ」