Vol.2014-NL-218 No.3
2014/9/1

周辺語義モデルによる日本語の教師無し語義曖昧性解消

佐々木悠人[†1]　古宮嘉那子[†2]　森田一[†3]　小谷善行[†1]

本研究では，多義語の周辺に現れる語義の分布を利用する周辺語義モデルを提案し，日本語に対する教師無しの All-word の語義曖昧性解消を行った．システムには EDR 電子化辞書による概念体系辞書を組み込み，EDR の日本語コーパスを用いて実験を行った．ランダムベースラインおよびトピックモデル（LDAWN）を用いた実験結果と比較したところ，周辺語義モデルは語義の頻度分布のエントロピーによる難易度が高い単語に対して特に優れた結果を示した．

Unsupervised Japanese Word Sense Disambiguation Using Surrounding Word Sense Model

YUTO SASAKI[†1]　KANAKO KOMIYA[†2]
HAJIME MORITA[†3]　YOSHIYKI KOTANI[†1]

This paper proposes surrounding word sense model, that uses distribution of word senses that appear nearby the ambiguous words, for unsupervised all-word word sense disambiguation in Japanese. Concept dictionary of EDR electronic dictionary was embedded in the system and Japanese Corpus of EDR was used for the experiments. The experiments showed that the surrounding word sense model outperformed the system with random baseline and the system that uses topic model (LDAWN) especially when the entropy of the word sense distribution of the ambitious words is high.

1. はじめに

語義曖昧性解消（Word Sense Disambiguation，WSD）とは，複数の語義を持つ単語（多義語）が文章中に出現した際に, どの語義を表しているのかを判断するタスクである．語義タグ付きコーパスのような教師データを必要としない教師無し WSD に関する研究は盛んに行われており，単語列 w が与えられた条件下での語義列 s の条件付き確率 p(s|w)をもとに多義語の語義を推定する手法もそのひとつである．本研究は，このような手法の一種として，周辺語義モデルを提案し，日本語の All-word の教師無し WSD を行う．

周辺語義モデルは，多義語の語義ごとに周辺に現れる語義の分布が異なることを仮定し，各語義が周辺の語義に関する確率分布を持つ．この確率分布の事前分布をディリクレ分布とし,そのパラメータを語義ごとに設定することで，各語義の周辺語義分布に事前に差をつけ，辞書中の語義との対応付けを行う．語義ごとのパラメータは，タグ無しコーパス中で実際に各語義の周辺に現れた語義をカウントして得た周辺語義頻度をもとに計算する．

本稿では，EDR の日本語コーパスと概念体系辞書を用いた実験から，日本語の教師なし WSD において，周辺語義モデルが有効であることを示す．

†1 東京農工大学
Tokyo University of Agriculture and Technology
†2 茨城大学
Ibaraki University
†3 京都大学
Kyoto University

2. 関連研究

WSD とは, 文章中の多義語の語義を推定するタスクであり，その手法は大きく教師有り学習と教師無し学習の二つに分けられる．教師有り学習では，語義タグ付きコーパスなど人手で用意された教師データを利用して，SVM（Support Vector Machine）などの機械学習手法により学習を行う．教師有り学習では高い精度で多義語の語義を推定することが可能だが，学習データの作成には高い人的コストがかかるため，あらゆる多義語に対応できる量のデータを用意することは不可能である．そのため，幅広い多義語に対応するためには，教師無し学習手法の精度向上が必要となる．

教師無し WSD に関する研究は多く，様々な手法が考えられている．Pedersen らは，WSD の対象語の語義と周辺単語の語義との間の意味的類似性を計算し，適切な語義を選択する手法を提案している[1]．確率的なモデルを利用したものとしては，Boyd-Graber らの研究[2]や Guo らの研究[3]がある．Boyd-Graber らは，トピックが持つ単語の確率分布を概念体系 WordNet 上での単語生成過程である WORDNET-WALK に置き換えた Latent Dirichlet Allocation with WORDNET（LDAWN）というモデルを考案し，トピックモデルを教師無しの英語 WSD へ応用した．Guo らも同様にトピックモデルと WordNet[4]の組み合わせだが，概念構造は利用せず，辞書の定義文から事前学習を行う手法

で，WSD に関して Boyd-Graber らと同程度の精度を上げている．

WSD では一般に，システムは内部に辞書を持っており，辞書中で定義されている語義に従って単語の語義推定を行う．一方，辞書の語義を利用せずに，文脈情報から多義語をクラスタリングし，分類された各クラスを語義と見なす手法もある．そういった手法は語義推定（Word Sense Induction，WSI）と呼ばれ，WSD とは区別されることが多い．Agirre らは，多義語の周辺単語の共起情報を基にグラフを作成し，クラスタリングを行うことで語義を判断するグラフベースの手法を報告している[5]．確率的なモデルで WSI を行った研究としては，Brody らの研究[6]がある．Brody らは，周辺の文脈から複数種類の素性を抽出し，それらを組み合わせる手法で成果を上げている．

本研究では，教師無し WSD に対する一つのアプローチとして周辺語義モデルを提案し，日本語の All-word の教師無し WSD を行う．入力は日本語のコーパス集合とし，文書全体の全多義語に対して概念体系辞書に基づき語義を推定する．

3. 周辺語義モデルによる WSD システム

WSD は，多義語に対して，文脈を考慮して最も適切だと思われる語義をシステム内部の辞書から選択する．本稿のシステムの入力から出力までは以下のようになる．

まず，入力として語義タグのついていないコーパスの集合を受け取る．これに対して形態素解析を行い，文章の単語への分割，品詞タグ付け，動詞の原形化，自立語の判定を行う．ただし，コーパスにすでにこれらの情報が付与されている場合，この処理は必要ない．こうして得られた単語列のうち，名詞と動詞の自立語を対象語とし，これらの基本形と品詞を取得する．本研究で提案するモデルでは周辺の単語の情報も利用するため，これらの情報に加えて周辺に現れた対象語という情報も取得する．なお，名詞・動詞以外の品詞の単語や非自立語については，本システムでは WSD の対象として扱わない．また，単語の語義はシステムが内部に持つ辞書に定義されたものの中から選択するため，辞書に載っていない単語や語義も扱わない．

入力として対象語が与えられると，システムは与えられたすべての対象語に対して，概念辞書と周辺語義モデルによりその語義を推定する．最終的に，すべての対象語に対して，各対象語が取りうる語義を辞書から一つ選択して割り当てた結果が出力として得られる．例として，「当たり障りのない内容になった事例は多い」という文章に対する形態素解析の結果を表 1 に，形態素解析結果の中の対象語とそれに対するシステムの出力の例を表 2 に示す．なお，表 2 における出力語義は，EDR 電子化辞書[7]における語義 ID である．

表 1　原文「当たり障りのない内容になった事例は多い」の形態素解析結果

表層形	基本形	品詞
当たり障り	当たり障り	名詞
の	の	助詞
ない	ない	形容詞
内容	内容	名詞
に	に	助詞
なっ	なる	動詞ー自立
た	た	助動詞
事例	事例	名詞
は	は	助詞
多い	多い	形容詞

表 2　原文「当たり障りのない内容になった事例は多い」における
WSD の対象語とシステムの出力の例

対象語	出力：語義
当たり障り	0e31d7
内容	3bc701
なる	3ceae3
事例	0f7497

3.1　周辺語義モデル

周辺語義モデルは，多義語の語義を判断する情報として周辺に現れる語義の分布を利用する．このモデルでは，多義語の語義によって周辺に現れる語義に違いが生じることを仮定している．たとえば，「可能性」という単語は，EDR の単語辞書によると，次の三つの意味がある：

(1). 物事をうまくやりこなすことのできる力
(2). 実現できる見込み
(3). 起こりうる確実性の度合い

たとえば，この例において，事前分布としては(3)の意味が最も高くなるが，周辺に「人間」や「研究」といった単語が現れると(1)の意味を取る確率が高くなり，最終的に(1)の意味だと判断されやすい．実際には周辺語義の分布の違いを教師無しで厳密に学習することは困難な問題だが，周辺語義モデルではこういった状況を考慮したモデルを近似的に作成した．

なお，本稿では，対象語の前後 N 個ずつの形態素（ただし，記号は含めない）をサイズ 2N のローカルウィンドウ

と定義し，ローカルウィンドウ内に含まれる，名詞もしくは動詞の自立語の語義を周辺語義とする．例として，EDR 電子化辞書のコーパス中の一文「両者とも、人間の可能性というものを聴く者に考えさせた演奏だった」という文について考える. コーパスに付与されている形態素情報では，この文は次のような形態素に分かれる：

両者/とも/、/人間/の/可能性/と/い/う/もの/を/聴/く/者/に/考え/させ/た/演奏/だっ/た

形態素の分かれ目にスラッシュを入れている．この例の場合，ローカルウィンドウのサイズを N=10 とすると（つまり，前後 5 形態素ずつ)，「可能性」という単語の周囲の形態素は，

両者，とも，人間，の，と，い，う，もの，を

の九個となる．この中で，名詞もしくは動詞の自立語は「両者」「人間」の二語なので，「可能性」の周辺語義としてはこれら二語の語義を考えることになる．

3.2 周辺語義モデルによる多義語の語義推定

周辺語義モデルでは，全文書の全単語列 w が観測された状態での，各単語に対応する語義列 s の確率 $P(\boldsymbol{s}|\boldsymbol{w})$ に従って確率的に語義列を選択し，多義語の語義とする方法を取る．

周辺語義モデルのために，まず，周辺語義を表す確率変数 $\boldsymbol{c}$ を導入する．3.1 節の文を例にとると，対象語 w_i = 可能性,「可能性」「両者」「人間」の語義をそれぞれ $s_{\text{可能性}}$，$s_{\text{両者}}$，$s_{\text{人間}}$ とすると，w_i の周辺語義 c_i は，$c_i = (s_{\text{両者}}, s_{\text{人間}})$ となる．このとき，

$$\boldsymbol{P}\left(\boldsymbol{c_i}\middle|\boldsymbol{s_{\text{可能性}}}\right) = \boldsymbol{P}\left(\boldsymbol{s_{\text{両者}}}\middle|\boldsymbol{s_{\text{可能性}}}\right)\boldsymbol{P}\left(\boldsymbol{s_{\text{人間}}}\middle|\boldsymbol{s_{\text{可能性}}}\right) \quad (1)$$

と定義する．これは，各語義が周辺語義に関する確率分布を持っており，周辺語義 c_i の確率は含まれる語義の確率の積になることを意味している．

さらに，多義語は取りうる語義に関する確率分布を持つことも仮定する．これは，多義語の語義に関する事前分布に相当する．

以上の仮定に基づき，周辺語義モデルでは，単語列 $\boldsymbol{w}$ が観測された条件下での対応する語義列 $\boldsymbol{s}$ の条件付き確率を次のように計算する：

$$\boldsymbol{P}(\boldsymbol{s},\boldsymbol{c}|\boldsymbol{w}) = \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{N}} \boldsymbol{P}(\boldsymbol{s_i}|\boldsymbol{w_i})\boldsymbol{P}(\boldsymbol{c_i}|\boldsymbol{s_i},\boldsymbol{w}) \quad (2)$$

N は全対象語数である(2)式右辺の前半部分は各単語が持つ語義に関する確率分布，後半部分は各語義が持つ周辺語義に関する確率分布であり，それぞれの事前分布にはディリクレ分布を設定する．

パラメータまで考慮した最終的な式は次のようになる：

$$\boldsymbol{P}(\boldsymbol{s},\boldsymbol{c},\boldsymbol{\theta},\boldsymbol{\phi}|\boldsymbol{w}) = \prod_{\boldsymbol{k}=\boldsymbol{1}}^{\boldsymbol{W}} \boldsymbol{P}(\boldsymbol{\theta_k}|\boldsymbol{\gamma_k}) \prod_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{S}} \boldsymbol{P}(\boldsymbol{\phi_j}|\boldsymbol{\tau_j}) \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{N}} \boldsymbol{P}(\boldsymbol{s_i}|\boldsymbol{\theta}_{\boldsymbol{w_i}})\boldsymbol{P}(\boldsymbol{c_i}|\boldsymbol{\phi}_{\boldsymbol{s_i}},\boldsymbol{w}) \quad (3)$$

W は総単語数，S は総語義数であり，$\boldsymbol{\theta_k}$，$\boldsymbol{\phi_j}$ はそれぞれ単語 k が持つ語義に関する確率分布，語義 j が持つ周辺語義に関する確率分布であり，多項分布のパラメータである．γ，τ はディリクレ分布のパラメータである．(3)式が基本的な形であるが，今回は各語義が持つ周辺語義に関する確率分布 $\boldsymbol{\phi}$ を，概念辞書の WORDNET-WALK（後述）による生成過程に置き換える．ただし，この WORDNET-WALK は，単語ではなく語義の生成となる．ハイパーパラメータには遷移確率パラメータ $S\alpha$ を設定する．この置き換えにより，周辺語義に似た概念が集まりやすくなることが期待できるが，一方でパラメータ数が増大するという問題もある．

このモデルは，このままではある種のクラスタリングは行うが，語義 $\boldsymbol{s}$ が辞書中のどの語義に対応するのかを定めることができない．そのため，WORDNET-WALK における遷移確率パラメータ $S\alpha$ を語義ごとに設定することで周辺語義の分布に事前に差をつける方法を試みる．ハイパーパラメータの設定，及び具体的な語義 $\boldsymbol{s}$ の選択方法を以下に述べる．

3.3 周辺語義モデルにおける概念構造

周辺語義モデルでは，WORDNET-WALK を利用して周辺語義に関する確率を得る. WORDNET-WALK とは，WordNet のような概念同士の上位下位関係が定義されている概念体系において，ルート概念から確率的に下位概念への遷移を繰り返していき，末端まで辿りついたらその概念が示す単語を出力する，という単語生成過程である． 図 1 に WORDNET-WALK による単語の生成確率の簡単な例を示す. 丸いノードは概念，三角のノードはリーフ概念 (X, Y) を語義として持つ単語であり，数字は遷移確率を表しているとすると，単語 A, B, C, D の生成確率はそれぞれ 0.03, 0.27, 0.28, 0.42 である．Boyd-Graber らの研究（LDAWN）では，トピックごとの単語の確率分布をこのようなルート概念からの遷移による確率分布に置き換えることで，単語の語義ごとの確率を扱う．たとえば，単語 A と単語 C が同じ単語の場合，この単語は語義 X より語義 Y を取りやすい．

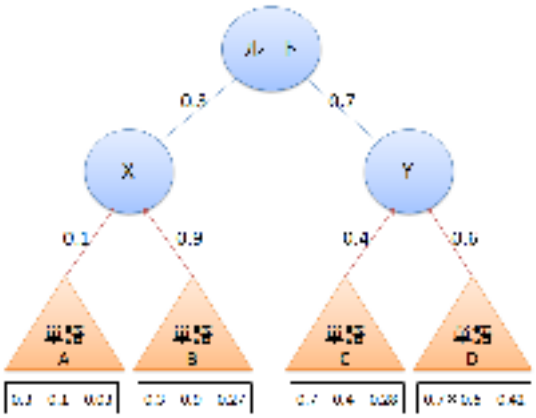

図 1　WORDNET-WALK による単語の生成確率の例

周辺語義モデルでは，周辺の語義に関する確率を利用するので，概念と単語間のリンクの確率は考慮しない．しかし，単純に概念と単語間の確率を無視して各概念への遷移確率を計算すると，上位の概念の確率は必ず下位の概念よりも高くなる．また，出現しうるすべての概念についての総和が 1 にならない．これは，下位の概念に遷移しないという確率を考慮していないために起こる問題である．

このため，周辺語義モデルでは生成される単語をリーフ概念ととらえて図 2 のような概念構造を持たせ，ある概念の出現する確率を，その概念から直接単語が生成される確率とする．このような概念構造において，概念 B の出現確率を概念 B に直接リンクするリーフ概念 B（図 2 三角のノード）の出現確率と置換えて考える．こうすることで，ルート概念からリーフ概念までの遷移確率として各概念の出現する確率を扱うことが出来る．

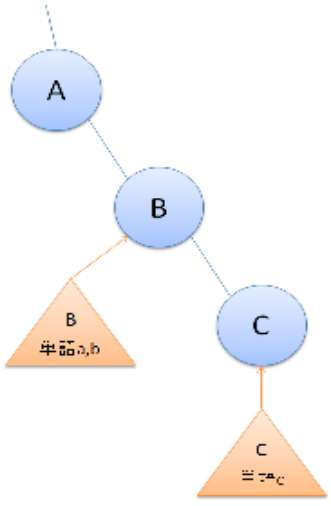

図 2　周辺語義モデルにおける概念構造

3.4　事前学習

周辺語義モデルでは，遷移確率パラメータ$S\alpha$により各語義の周辺語義の分布に事前に差をつける．語義タグ付きコーパスがある場合は，コーパスの語義情報を使って周辺に出現しやすい語義を学習することができるが，本研究では語義タグ付きコーパスを利用しない教師無し学習を目的としており，語義タグ付きコーパスは使用しない．そこで今回は，語義タグの付いていないコーパスから語義ごとの周辺語義の分布の違いを事前学習する方法として，コーパス中に出現した周辺単語の取りうる語義をすべてカウントする方法を試みる．さらに，意味的に類似した語義同士では周辺語義の分布も似ていると考え，概念体系上の近隣概念をまとめる「概念抽象化」により複数の類似概念をまとめる処理を行う．

3.4.1 周辺語義頻度の取得

各語義の周辺分布の差を事前に学習し，遷移確率パラメータαに反映させる方法として，語義タグのついていないコーパス中で，各語義の周辺に実際に現れた単語の語義をカウントし，その値をもとにパラメータαを計算する方法を用いる．ただし，単語の正しい語義は分からないので，取りうる語義すべてについてカウントを行う．つまり，コーパス中で多義語 A の周囲に多義語 B が現れた場合，多義語 A の取りうるすべての語義について，周辺に多義語 B の全語義が現れたとみなしてカウントを行う．これでは多義語 A のすべての語義に同様のカウントが行われるので，A の各語義の周辺語義の分布に違いは現れない．ここで，別の多義語もしくは単義語 C があったとし，C の語義の中に A と共通の語義があるとする．単語 C の語義についても同様に，コーパス中で周辺に現れた単語の全語義がカウントされる．すると，A と C とで共通の語義には，A と C の二つの単語の周辺語義のカウントが与えられることになる．この語義を持つ単語が他にもあったとすると，その単語の分のカウントも与えられ，最終的に多義語 A の他の語義とカウントの結果に差が生じる．こうして各語義の周辺語義のカウントを求め，遷移確率パラメータαの計算に利用する．この方法は，上記の A と C のような，共通の語義sを持つ単語の周辺に共通して現れやすい単語の語義は，sの周辺に現れやすい語義を含んでいることを期待している．

簡単な例として，コーパス中の「跡地を何に利用すれば事業成功の可能性が高いかを診断してくれる」という文章を考える．形態素解析の結果から対象語を抽出し，さらに各対象語についてウィンドウサイズ 10 のローカルウィンドウ内に含まれる周辺の対象語情報を得ると表 3 のようになる．「診断する」については，前後 5 形態素内に対象語が存在しなかった．ここで，「可能性」について見てみる．

「可能性」の語義は 2.5 節で述べたように，次の三つである：

(1). 物事をうまくやりこなすことのできる力
(2). 実現できる見込み
(3). 起こりうる確実性の度合い

今回の例では「可能性」の周辺の対象語は「事業」と「成功」であり，「可能性」の語義(1)〜(3)の周辺語義として，「事業」と「成功」が取りうるすべての語義がカウントされる．さらにコーパス中の他の文章で，「成功の見込みがない」というような文があったとすると，「見込み」は「可能性」の(2)の語義を持っているので，(2)の語義に「成功」の語義がカウントされる．このようにして，(1)〜(3)の語義が，それぞれの周辺に現れやすい語義を多く含むカウント値を得られることを期待する．

表 3　対象語とローカルウィンドウ内の対象語

対象語	周辺の対象語
跡地	何，利用する
何	跡地，利用する，事業
利用する	跡地，何，事業，成功
事業	何，利用する，成功，可能性
成功	利用する，事業，可能性
可能性	事業，成功
診断する	

なお，カウントの方法として，多義語の影響を単義語よりも減らす，対象語の近くの周辺後の影響を強くするなどの工夫も考えられるが，ここでは単純に，出現した語義すべてについて 1 ずつ出現回数を数えていく方法を採用する．

3.4.2 遷移確率パラメータの設定

3.4.1 節で得られた周辺語義頻度のカウントをもとに，各概念への遷移確率が周辺語義頻度に比例するような遷移確率パラメータを設定する．

実際に単語トークンの出現頻度から遷移確率を計算する方法として，各概念sの確率$P(s)$を求め，概念s_iから概念s_jへの遷移確率$P(s_j|s_i)$を，

$$\boldsymbol{P(s_j|s_i) = \frac{P(s_i, s_j)}{P(s_i)} = \frac{P(s_j)}{P(s_i)}} \tag{4}$$

として求める方法がある[8]．この遷移確率をそのまま概念s_iから概念s_jへの遷移確率パラメータα_{s_i,s_j}の値とする．

単語の出現頻度から概念の確率$P(s)$を求める方法として，ここでは Resnik の手法[9]がある．Resnik の手法では，ある概念の出現頻度は，その概念が含む概念（その概念自身と，その概念のすべての下位概念）に属する単語の出現頻度の総和であるとし，概念sの頻度$freq(s)$を，

$$\boldsymbol{freq(s) = \sum_{w \in words(s)} count(w)} \tag{5}$$

と計算する．ここで，$words(s)$は概念sとその下位概念に属する全単語であり，$count(w)$は単語wのコーパス中での出現頻度である．概念の確率$P(s)$は，単語トークンの出現数の総和をNとすると，

$$\boldsymbol{P(s) = \frac{freq(s)}{N}} \tag{6}$$

と求めることができる．$freq(s)$は概念s以下に属する全単語の頻度の総和なので，この手法では$P(s)$は概念sが含む単語の出現頻度に比例する．しかし，Resnik の手法では，ある概念から下位概念への遷移確率の和が 1 にならない．

そのため， Resnik の手法を，各単語を語義への経路によって別のものと考えて頻度を算出する．この場合の各概念の頻度を式で表すと，概念s_iから概念s_jへ到達する経路の数を$path(s_i, s_j)$，概念sが含むリーフ概念の集合を$L(s)$としたとき，

$$\boldsymbol{freq(s_i) = \sum_{s \in L(s_i)} path(s_i, s) \sum_{w \in words(s)} count(w)} \tag{7}$$

となる．遷移確率は下位概念の頻度を上位概念の頻度で割ることで計算する．

図 3 に簡単な概念体系の例を示す．図中の各ノード A〜F が概念であり，(a)のような表記は単語 a が取りうる語義ということを表している．つまり，この例では単語 a は多義語であり，語義として C か D を取りうる．a が二回，b が一回出現したとする．この場合，概念 A の頻度は，ABD の経路の a が二回，AC の a が二回，ABE の b が一回，ACE の b が一回の計六回と数え，A から B への遷移確率$\alpha_{A,B}$は，

$$\boldsymbol{\alpha_{A,B} = \frac{freq(B)}{freq(A)} = \frac{3}{6} = \frac{1}{2}} \tag{8}$$

と求まる．この方法では，ある概念の頻度はその直下の概念の頻度の総和になるので，下位概念への遷移確率の総和は常に 1 となる．

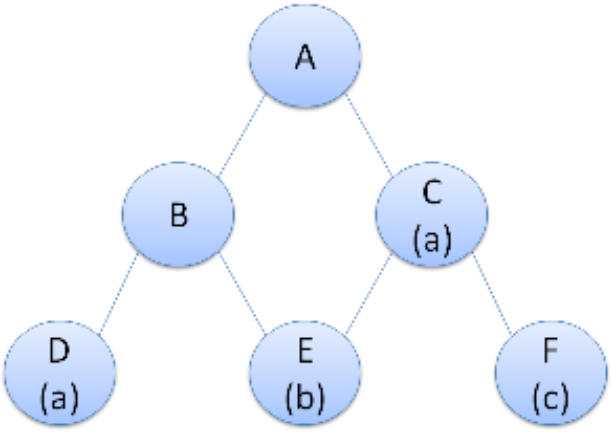

図 3　簡単な概念体系の例

ルート概念s_{root}から任意のリーフ概念s_lへ到達する，任意の経路$path_{s_l}$の遷移確率$p(path_{s_l})$は，経路$path_{s_l}$中で通過する概念を$c_1 c_2 \dots c_n$とすると，

$$\begin{aligned} &\boldsymbol{P(path_{s_l})} \\ &= \frac{\boldsymbol{freq(c_1)}}{\boldsymbol{freq(s_{root})}} \frac{\boldsymbol{freq(c_2)}}{\boldsymbol{freq(c_1)}} \cdots\cdots \frac{\boldsymbol{freq(c_n)}}{\boldsymbol{freq(c_{n-1})}} \frac{\boldsymbol{freq(s_l)}}{\boldsymbol{freq(c_n)}} \\ &= \frac{\boldsymbol{freq(s_l)}}{\boldsymbol{freq(s_{root})}} \end{aligned} \quad (9)$$

である．よって，$freq(s_l)$にs_lの周辺語義頻度のカウントを設定すると，ルートからs_lまでの各経路の確率がカウントに比例する値となる．

この際，学習用のコーパスで周辺に出現しなかった語義の確率が 0 となる問題が起こる．この問題に対しては，頻度補正を行うのではなく，語義sの周辺語義頻度のカウントから計算した遷移確率パラメータをα_b^s，均等な遷移確率パラメータをα_aとし，

$$\boldsymbol{S_a \alpha_a + S_b \alpha_b^s} \quad (10)$$

を遷移確率パラメータとする．S_a，S_bはそれぞれ定数である．このように設定することで，S_bの値によって，事前学習した周辺語義によるカウントの影響を調整できる．

均等な遷移確率パラメータをα_aは，以下のように求める．まず，すべてのリーフ概念の頻度$freq(s_l) = 1$とすると，

$$\begin{aligned} &\boldsymbol{p(path_{s_l})} = \frac{\boldsymbol{freq(s_l)}}{\boldsymbol{freq(s_{root})}} \\ &= \frac{\boldsymbol{1}}{\sum_{s \in L(s_{root})} \boldsymbol{path(s_{root}, s) freq(s)}} \\ &= \frac{\boldsymbol{1}}{\sum_{s \in \boldsymbol{L}(s_{root})} \boldsymbol{path(s_{root}, s)}} \end{aligned} \quad (11)$$

となり，s_lによらない定数となる．よって，すべてのリーフ概念の頻度を 1 にすると，ルート概念から各リーフ概念への各経路の確率が等しくなるような遷移確率パラメータαを設定することができる．ただしこの場合，ルート概念からの経路数が多いリーフ概念ほど確率が高くなる．すべてのリーフ概念の確率を等しくしたい場合，リーフ概念s_lの頻度を 1 ではなく，

$$\boldsymbol{freq(s_l)} = \frac{\boldsymbol{1}}{\boldsymbol{path(s_{root}, s_l)}} \quad (12)$$

に設定する．

3.4.3 均等クラス確率法による概念抽象化

概念抽象化とは，概念体系において，下位の概念をより上位の抽象的な概念にマッピングする操作である．概念体系において，深い階層にある概念は概念粒度が小さく，「子猫」や「ペルシャ猫」，「東京農工大学」や「東京大学」など具体的な概念となっている．「子猫」と「ペルシャ猫」や「東京農工大学」と「東京大学」の周辺の語義の分布に大きな違いがあるとは考えられず，これらの概念はより抽象的な「猫」や「大学」といった概念としてまとめて扱った方が都合が良い．また，このように概念を抽象的な上位概念にまとめあげることで，コーパスから周辺単語の語義をカウントする際に，意味的に近い単語同士のカウントが共有され，周辺語義の分布の差がより顕著に出ることが期待できる．

平川ら[10]は「均等深度法」「均等サイズ法」「均等クラス確率法」の三つの抽象化手法について EDR 電子化辞書で比較した結果，「均等クラス確率法」，「均等サイズ法」，「均等深度法」の順に良い結果が得られたことを報告している．そのため，本研究では，均等クラス確率法で概念抽象化を行う．

均等クラス確率法は，ルート概念から深さ優先で探索し，コーパスから計算した概念の確率（クラス確率）がある一定値未満となる概念について，その概念と下位のすべての概念を上位の概念にマッピングする手法である． 図 4 に，クラス確率 0.30 の均等クラス確率法による概念抽象化の例を示す．ノード中に書かれた数字が概念の確率を表す．

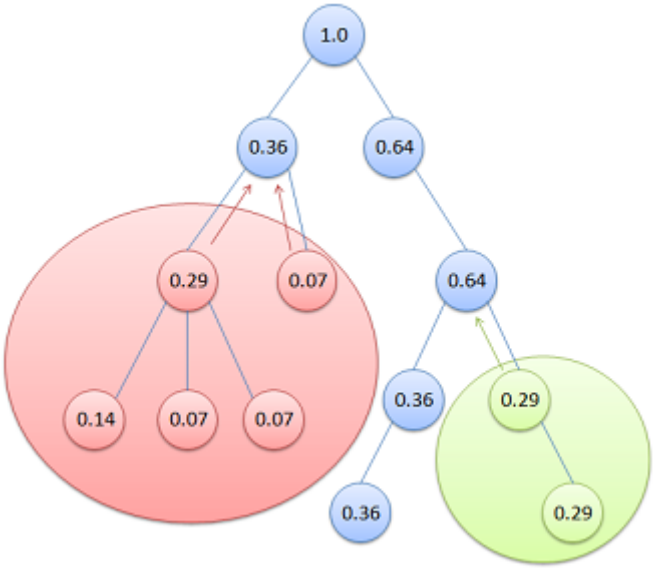

図 4　クラス確率 0.30 の均等クラス確率法による概念抽象化の例

概念確率の計算には，Ribas[11]，McCarty[12]の手法を採用する．Ribas は単語の出現頻度をその単語が取りうる語義数で割り，その値を各語義に割り振る手法を採用している．単語wの取りうる語義を$senses(w)$，概念sとその下位概念の集合を$U(s)$，$count(w)$を単語wのコーパス中での出現頻度とすると，Ribas の手法における概念sの頻度$freq(s)$は，

$$freq(s) = \sum_{w} \frac{|senses(w) \in U(s)|}{|senses(w)|} count(w) \tag{13}$$

と表される．これは，単語wの出現頻度$count(w)$に対し，wの語義のうち，概念sとその下位の概念に含まれる語義の割合で重み付けをしていることになる．クラス確率$P(s)$は，単語トークンの出現数の総和をNとすると，

$$P(s) = \frac{freq(s)}{N} \tag{14}$$

と求める．概念構造において，コーパス中に単語 a が二回，単語 b が一回出現したとする．この場合，Ribas の手法による各概念の頻度と確率は図 5 のようになる．単語 a が語義 C と D を持つ多義語となっているので，単語 a の出現頻度は C と D に 1/2 ずつ割り当てられている．図 5 ではリーフ概念が省略されているが，実際には C，D，E，F の直下にそれぞれリーフ概念が存在している．

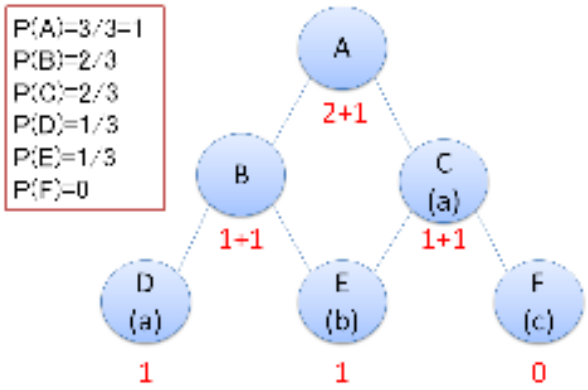

図 5　Ribas の手法による各概念の頻度と確率

この場合の概念確率の計算に対してもゼロ頻度問題を考慮する必要があるため，グッド・チューリング推定法により頻度の補正を行う．グッド･チューリング推定法では，コーパス中にr回出現した語義（リーフ概念）の補正後の頻度r^*に次の値を使う[13]：

$$r^* = (r+1)\frac{N_{r+1}}{N_r} \tag{15}$$

N_rはコーパス中にr回出現した語義の数である．今回の場合，多義語の出現はその回数を全語義に均等に割り振っているため，頻度rは整数にならない．そこで，N_rを求める際には，各語義の頻度の小数点以下第一位を四捨五入して考える．たとえば，$r = 2.4$の語義はN_2にカウントされる．

上記の(15)式の補正方法では，$N_r = 0$となる場合に補正を行うことができない．そこで，rが小さい場合のみ(15)式で補正を行い，rが大きい場合は出現頻度rをそのまま使うことにする．ここで，rが大きい場合とは，$N_r = 0$か，$N_{r+1} > N_r$となるr以上の値とする．後者の条件は，rが大きい場合の補正後の頻度を，補正前より大きくしすぎないための条件である．

概念抽象化を行うと，多義語の複数の語義が同一の上位概念に抽象化されてしまう可能性がある．WSD の実験では，最終的な出力として抽象化された概念から元の概念に戻した語義を出力する必要がある．しかし，複数の語義が同一の上位概念に抽象化されていると，元の語義を一意に定めることができない．そのような場合，元の語義の候補間で概念確率を比較し，最も概念確率の高い語義を出力した．

3.5　ギブスサンプリングによる語義の推定

周辺語義モデルでは，ギブスサンプリング[14]で語義sを推定する．3.2 節で述べたように，モデルの条件付き確率は，

$$P(\boldsymbol{s},\boldsymbol{c},\boldsymbol{\theta},\boldsymbol{\phi}|\boldsymbol{w}) = \prod_{k=1}^{W} P(\boldsymbol{\theta}_k|\boldsymbol{\gamma}) \prod_{j=1}^{S} P(\boldsymbol{\phi}_j|\boldsymbol{\tau}_j) \prod_{i=1}^{N} P(s_i|\boldsymbol{\theta}_{w_i}) P(c_i|\boldsymbol{\phi}_{s_i}, \boldsymbol{w}) \quad (16)$$

である．(16)式から，サンプリングに必要となる，単語w_iに関する変量以外を定数と見なした場合の条件付き分布を求める．実際は語義が持つ確率分布$\boldsymbol{\phi}$は WORDNET-WALK による語義生成に置き換え，下位概念への遷移に関する複数の多項分布となるが，ここでは数式の簡単化のため$\boldsymbol{\phi}$のまま計算を行う．まず，$\boldsymbol{\theta}$と$\boldsymbol{\phi}$を積分消去すると，

$$P(\boldsymbol{s},\boldsymbol{c}|\boldsymbol{w}) = \int_{\theta} \prod_{k=1}^{W} P(\boldsymbol{\theta}_k|\boldsymbol{\gamma}) \prod_{i=1}^{N} P(s_i|\boldsymbol{\theta}_{w_i})\, d\boldsymbol{\theta} \cdot \int_{\phi} \prod_{j=1}^{S} P(\boldsymbol{\phi}_j|\boldsymbol{\tau}_j) \prod_{i=1}^{N} P(c_i|\boldsymbol{\phi}_{s_i}, \boldsymbol{w})\, d\boldsymbol{\phi} \quad (17)$$

(17)式の前半部分について，各単語kに関するθ_kは独立なので，

$$\int_{\theta} \prod_{k=1}^{W} P(\boldsymbol{\theta}_k|\boldsymbol{\gamma}) \prod_{i=1}^{N} P(s_i|\boldsymbol{\theta}_{w_i})\, d\boldsymbol{\theta} = \prod_{k=1}^{W} \int_{\theta_k} P(\boldsymbol{\theta}_k|\boldsymbol{\gamma}) \prod_{\{i|w_i=k\}} P(s_i|\boldsymbol{\theta}_k)\, d\boldsymbol{\theta}_k \quad (18)$$

ここで，単語kの取りうる語義の集合を$sen(k)$とおく．$P(\theta_k|\gamma)$はディリクレ分布，$P(s_i|\theta_k)$は$n=1$の多項分布なので，(18)式は次のようになる：

$$\prod_{k=1}^{W} \int_{\theta_k} \frac{\Gamma\left(\sum_{sen\in sen(k)} \boldsymbol{\gamma}_{sen}\right)}{\prod_{sen\in sen(k)} \Gamma(\boldsymbol{\gamma}_{sen})} \cdot \prod_{sen\in sen(k)} \boldsymbol{\theta}_{k,sen}^{\gamma_{sen}-1} \boldsymbol{\theta}_{k,sen}^{n_{k,sen}}\, d\boldsymbol{\theta}_k \quad (19)$$

$n_{k,sen}$は単語kに語義senが割り当てられた回数であり，Γはガンマ関数である．$\boldsymbol{\gamma'}_{sen} = \boldsymbol{n}_{k,sen} + \boldsymbol{\gamma}_{sen}$と置くと、

$$\int_{\theta_k} \boldsymbol{f}(\boldsymbol{\theta}_k|\boldsymbol{\gamma'})\, d\boldsymbol{\theta}_k = \int_{\theta_k} \frac{\Gamma\left(\sum_{sen\in sen(k)} \left(\boldsymbol{n}_{k,sen} + \boldsymbol{\gamma}_{sen}\right)\right)}{\prod_{sen\in sen(k)} \Gamma\left(\boldsymbol{n}_{k,sen} + \boldsymbol{\gamma}_{sen}\right)} \cdot \prod_{sen\in sen(k)} \boldsymbol{\theta}_{k,sen}^{n_{k,sen}+\gamma_{sen}-1}\, d\boldsymbol{\theta}_k = \mathbf{1} \quad (20)$$

(fはディリクレ分布を表す関数)が成り立つので，(19)式は最終的に次の形になる：

$$\prod_{k=1}^{W} \frac{\Gamma\left(\sum_{sen\in sen(k)} \boldsymbol{\gamma}_{sen}\right)}{\prod_{sen\in sen(k)} \Gamma(\boldsymbol{\gamma}_{sen})} \cdot \frac{\prod_{sen\in sen(k)} \Gamma(\boldsymbol{n}_{k,sen} + \boldsymbol{\gamma}_{sen})}{\Gamma\left(\sum_{sen\in sen(k)} (\boldsymbol{n}_{k,sen} + \boldsymbol{\gamma}_{sen})\right)} \quad (21)$$

(17)式の後半部分も同様に計算を行う．ただし，周辺語義$\boldsymbol{c}$について，これは語義列$\boldsymbol{s}$に対応して定めるものであるが，今回の導出では確率変数と見なす．そして，サンプリングを行う際に割り当てられている語義列に対応するように周辺語義$\boldsymbol{c}$を決定的に選択することにする．計算は(21)式の導出と同様に行う．語義jの周辺語義として語義senが出現した回数を$n_{j,sen}$とおくと，

$$\begin{aligned} &\int_{\phi} \prod_{j=1}^{S} P(\boldsymbol{\phi}_j|\boldsymbol{\tau}_j) \prod_{i=1}^{N} P(c_i|\boldsymbol{\phi}_{s_i}, \boldsymbol{w})\, d\boldsymbol{\phi} \\ &= \prod_{j=1}^{S} \int_{\phi_j} P(\boldsymbol{\phi}_j|\boldsymbol{\tau}_j) \prod_{\{i|s_i=j\}} P(c_i|\boldsymbol{\phi}_j, \boldsymbol{w}) d\boldsymbol{\phi}_j \\ &= \prod_{j=1}^{S} \int_{\phi_j} \frac{\Gamma\left(\sum_{sen} \boldsymbol{\tau}_{j,sen}\right)}{\prod_{sen} \Gamma(\boldsymbol{\tau}_{j,sen})} \prod_{sen} \boldsymbol{\phi}_{j,sen}^{\tau_j-1} \cdot \boldsymbol{\phi}_{j,sen}^{n_{j,sen}} d\boldsymbol{\phi}_j \end{aligned} \quad (22)$$

最終的に，次の形を得る：

$$\prod_{j=1}^{S} \frac{\Gamma\left(\sum_{sen} \boldsymbol{\tau}_{j,sen}\right)}{\prod_{sen} \Gamma(\boldsymbol{\tau}_{j,sen})} \cdot \frac{\prod_{sen} \Gamma(\boldsymbol{n}_{j,sen} + \boldsymbol{\tau}_{j,sen})}{\Gamma\left(\sum_{sen} (\boldsymbol{n}_{j,sen} + \boldsymbol{\tau}_{j,sen})\right)} \quad (23)$$

単語w_iに関する変量s_iとc_i以外の変量を定数と見なした場合の条件付き分布$P(s_i, c_i|\boldsymbol{s}_{-i}, \boldsymbol{c}_{-i}, \boldsymbol{w})$は，

$$\boldsymbol{P(s_i, c_i|s_{-i}, c_{-i}, w)} = \frac{\boldsymbol{P(s, c|w)}}{\boldsymbol{P(s_{-i}, c_{-i}|w)}} \propto \boldsymbol{P(s, c|w)} \quad (24)$$

となり，(17)式に比例する．(21)式と(23)式の結果と，ガンマ関数の性質$\Gamma(x+1) = x\Gamma(x)$から，条件付き分布$P(s_i, c_i|\boldsymbol{s_{-i}}, \boldsymbol{c_{-i}}, \boldsymbol{w})$は次のようになる：

$$\boldsymbol{P(s_i = x, c_i = y|s_{-i}, c_{-i}, w)} \propto \left(\boldsymbol{n^{-i}_{w_i,x}} + \boldsymbol{\gamma}\right) \cdot \prod_{j=1}^{|y|} \frac{\left(\boldsymbol{n^{-i}_{x,y_j}} + \boldsymbol{m_y(j, y_j)} + \boldsymbol{\tau_{x,y_j}}\right)}{\left(\sum_{sen}\left(\boldsymbol{n^{-i}_{x,sen}} + \boldsymbol{\tau_{x,sen}}\right) + (\boldsymbol{j-1})\right)} \quad (25)$$

x, yはそれぞれ語義s_i, 周辺語義c_iが実際にとる値であり，それぞれ何らかの語義，周辺語義（周囲の語義を並べたベクトル）となっている．$n^{-i}_{w_i,x}$は現在サンプリング対象としているi番目の変量を除き，単語w_iに割り当てられている語義xの数，n^{-i}_{x,y_j}は同様にi番目を除き，語義xの周囲に語義y_jが出現した数，$m_y(j, y_j)$は周辺語義yの中で，j番目より前に語義y_jが出現した数であり，yの中に同じ語義が複数出現しない場合は無視できる．実際のサンプリング時には，yに周辺の語義に対応する語義列を決定的にあてはめる近似処理を行った後，各s_iの確率を計算し，単語w_iに対応する語義s_iを決定する．

なお，語義が持つ分布を WORDNET-WALK に置き換えた場合，(25)式の後半部分を置き換えることになる．周辺語義y中の語義y_jのルート概念からの経路を$r_{j,0}, r_{j,1}, \ldots, r_{j,l}$としたとき，すべての語義のすべてのルート概念からの経路の組み合わせについて次の値を計算し，足し合わせると式(26)のようになる（図 6）．$T^{-i}_{x,r_{j,p},r_{j,p+1}}$は$i$番目の変量を除き，語義$x$の周辺語義が概念$r_{j,p}$から$r_{j,p+1}$へのリンクを通過した数であり，$m_y(j, r_{j,p}, r_{j,p+1})$は$y$の$j$番目の経路より前に概念$r_{j,p}$から$r_{j,p+1}$へのリンクが通過された回数である．

語義s_iを割り当てた後，T_{s_i}の値を更新する必要があり，そのためには周辺語義の経路が必要になる．すべての経路の組み合わせから(26)式に従って確率的に選択しても良いが，今回は各経路に確率に比例する値を与えることにした．語義s_iの周辺語義c_i中の各語義$c_{i,j}$について，そのルート概念からの経路を$path_1, path_2, \ldots, path_n$とすると，経路$path_k$上の各リンクの通過回数$T_{s_i,path_k}$には次の値を加算する：

$$\frac{\boldsymbol{P(path_k|s_i)}}{\sum_{l=1}^{n} \boldsymbol{P(path_l|s_i)}} \quad (27)$$

ただし，$path_k$が通過する概念を$r_1, r_2, \ldots, r_l$とすると，確率$P(path_k|s_i)$は

$$\boldsymbol{P(path_k|s_i)} = \prod_{p=1}^{l-1} \frac{T^{-i}_{s_i,r_p,r_{p+1}} + S_a\alpha_{a,r_p,r_{p+1}} + S_b\alpha^{s_i}_{b,r_p,r_{p+1}}}{\sum_r (T^{-i}_{s_i,r_p,r} + S_b\alpha^{s_i}_{b,r_p,r}) + S_a} \quad (28)$$

とする．ルートからの経路が複数存在する概念は複数の性質を多重継承している概念であり，この場合は語義$c_{i,j}$の一回の出現を各性質に割り振っていると見なせる．

4. 実験

4.1 データ

本研究では，EDR 電子化辞書（平成 14 年の Ver2.0）のうち，日本語の単語辞書，概念辞書，EDR コーパスの三つを使用して実験を行った．

4.1.1 システムの辞書の作成

システム内部の辞書の作成には，概念辞書と単語辞書を使用する．概念辞書は，「概念見出し辞書」と「概念体系辞書」の二つの辞書から成り，見出し辞書は概念識別子（概念を識別する 16 進数の整数）と概念見出し（概念の意味内容に近い単語），概念説明を対応付けている．概念体系辞書は概念同士の関係，特に上位下位概念関係によって体系化した辞書であり，二つの概念の上位下位関係を記述したレコードから成る．単語辞書は，単語見出しや品詞などの情報と，その単語が持つ語義（概念識別子）が記述されている．

$$\prod_{j=1}^{|y|} \prod_{p=1}^{l-1} \frac{\boldsymbol{T^{-i}_{x,r_{j,p},r_{j,p+1}}} + \boldsymbol{m_y(j, r_{j,p}, r_{j,p+1})} + \boldsymbol{S_a\alpha_{a,r_{j,p},r_{j,p+1}}} + \boldsymbol{S_b\alpha^{x}_{b,r_{j,p},r_{j,p+1}}}}{\sum_r \left(\boldsymbol{T^{-i}_{x,r_{j,p},r}} + \boldsymbol{m_y(j, r_{j,p}, r)} + \boldsymbol{S_b\alpha^{x}_{b,r_{j,p},r}}\right) + \boldsymbol{S_a}} \quad (26)$$

図 6　式(26)

辞書に登録する単語は，すべての名詞と動詞のうち，概念体系上でルート概念から辿ることができる単語とした．また，サ変動詞については末尾の「する」を除いた名詞形も登録した．その結果，登録された単語数（表層数）は 263757 個，概念構造の単語のリーフ概念数は 406710 個，周辺語義モデルで利用した語義のリーフ概念数（図 2）は 199430 個となった．

未使用概念や未分類概念を削除すると，最終的にリーフ概念を除いた概念数は 203565 個となった．未使用概念は，そのほとんどが英語の単語辞書からのみリンクされている概念である．また，未分類概念に含まれる日本語は「静電容量」のみだった．また，概念抽象化の概念確率の閾値には5.0×10^{-5}を設定し，抽象化後の総概念数は 13846（うち，リーフ概念 6905）となった．

4.1.2 EDR 日本語コーパス

WSD の実験には EDR 日本語コーパスを使用する．EDR のコーパスは，複数の出典元文書から文章単位で情報を抜き出し，形態素，構文情報，意味情報などを付与している．コーパスの各形態素には語義タグが付与されているため，システムの出力と比較して正解判定を行うことができる．

出典元文書は，「日本経済新聞」，「朝日新聞」，「アエラ」，「平凡社百科辞典」，「岩波情報科学辞典」，「雑誌」，「用例集」の七つである．それぞれの出典に含まれる文章数と総形態素数を表 4 に示す．

表 4　出典別の文章数と総形態素数

出典	文章数	総形態素数
日本経済新聞	5018	121301
朝日新聞	91400	2272555
アエラ	49589	1183897
平凡社百科辞典	10072	284059
岩波情報科学辞典	13578	357607
雑誌	21199	528452
用例集	16946	368285

今回は実験対象として日本経済新聞の全文章を選択する．日本経済新聞の全文章中の多義語の種類数は 4822，名詞と動詞のトークン数はそれぞれ 12149，6199 だった．また，多義語の平均語義数は名詞 4.2，動詞 5.5 だった．日本経済新聞以外の六つの出典の文章は，単語の出現頻度に比例する遷移確率の計算のためのコーパスとして使用する．

4.1.3 形態素情報の補足処理

EDR コーパスの各文章には形態素情報が付与されており，語義タグは各形態素に対して付与されている．そのため，語義タグを使用して正解判定を行うためには，EDR の形態素に合った入力を行う必要がある．しかし，EDR コーパスの形態素情報には，動詞の原形や自立語かどうかといった情報が含まれていない．システムの辞書は単語辞書から作られるため，単語は全て基本形で登録されており，動詞は原形でなければ扱うことができない．そこで，動詞の原形や自立語の情報を得るため，形態素解析器 MeCab[15] の解析結果と比較をして，対応が取れた形態素について原形化や自立語の判定を行った．

4.2 評価方法

システムは全ての対象語（名詞と動詞の自立語で，単語辞書に載っている単語）に対して語義を割り当てるので，正解判定は対象語の中の全ての多義語に対して行う．正解かどうかの判定はコーパスの語義タグ情報を使用し，システムに割り当てられた語義が語義タグと一致した場合に正解とする．周辺語義モデルでは概念抽象化を行うが，正解判定の際には抽象化する前の概念に戻してから正解判定を行う．ただし，語義タグがついていない場合と，語義タグがついているが単語辞書に載っていない語義タグを指しているものは正解判定から除外した．

さらに，SENSEVAL2 日本語タスク[16]における，語義の頻度分布のエントロピーを考慮した難易度設定に基づき，語義判別の難易度を設定する．コーパス中で五回以上出現した多義語に対して，表 5 の条件により三段階の難易度を設定した．

表 5　語義判別の難易度設定基準

難易度	エントロピーの範囲
Easy	$E(w) < 0.5$
Normal	$0.5 \leq E(w) < 1$
Hard	$1 \leq E(w)$

実験対象である日本経済新聞のコーパスにおける，難易度別の多義語の種類数とトークン数を表 6 に，多義語の平均語義数を表 7 に示す．なお，All は多義語全体に関するデータである．

表 6　日本経済新聞のコーパスにおける多義語の難易度別の種類数とトークン数

難易度	種類数	名詞トークン数	動詞トークン数
All	4822	12149	6199
Easy	399	3630	1723
Normal	337	2929	1541
Hard	105	1028	1196

表 7　日本経済新聞のコーパスにおける多義語の難易度別の平均語義数

難易度	名詞の平均語義数	動詞の平均語義数
All	4.2	5.5
Easy	3.9	4.0
Normal	4.4	5.3
Hard	8.6	10.3

本研究の手法はランダムなサンプリングによって語義を推定する手法であるため，毎回の実行で得られる結果が異なる．そのため，実験ではシステムを複数回実行し，各実行で得られた正解率の平均を取った．

5. 結果

周辺語義モデルを用いて, EDR コーパス中の日本経済新聞の全文章に対して WSD の実験を行った結果を示す．

周辺語義頻度の取得の際，周囲に現れた語義を 1 ずつカウントする方法を採用したが，最終的に周囲に一度しか出現しなかった語義はノイズとして削除した．周辺語義頻度の取得, 及び実験の際のローカルウィンドウのサイズは 10 に設定した．

実験は，遷移確率パラメータ$S_a = \{1.0,\ 5.0,\ 10.0\}$, $S_b = \{10.0,\ 15.0,\ 20.0\}$の計九通りのパラメータ設定について行った．$S_a$は均等な遷移確率パラメータ$\alpha_a$にかかる定数, S_bは周辺語義頻度から計算した遷移確率パラメータα_b^sにかかる定数である．均等な遷移確率パラメータα_aには，各リーフ概念への経路の確率を等しくするもの，また，各単語が持つ語義に関する確率分布のハイパーパラメータγはすべての実験で$\gamma = 0.1$と設定した．イテレーション回数は 2000 回(1800 回分のイテレーションで割り当てられた語義の中から，間をあけて 100 サンプル取り，最も多く割り当てられた語義を出力する.）とし，実行回数は 3 回とする．

全多義語，Easy，Normal，Hard の各場合について，S_bごとにマクロ平均が最も高くなった結果を抽出した表を表 8 ～表 11 に示す．

全体の結果（表 8）を見ると，各S_bに共通して$S_a = 1.0$とするのが良く，S_bは小さいほどマクロ平均が高い．名詞のマイクロ平均には大きな差がないが，動詞のマイクロ平均は$S_b = 20.0$の場合は低い．Easy の結果（表 9）では，マイクロ平均に差は出るが，マクロ平均では大きな差となっていない．Normal の結果（表 10）では，$S_b = 20.0$の場合の名詞のマイクロ平均がやや高いが，全体的には似たような結果となっている．Hard の結果（表 11）では，$S_b = 20.0$の場合の動詞と$S_b = 15.0$の場合の動詞のマイクロ平均が低く，マクロ平均は名詞，動詞ともに高かったと$S_b = 10.0$の場合が最も高い．

表 8　マクロ平均が高かった結果同士の比較（対象 All）

パラメータ S_a, S_b	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
$S_a = 1.0$ $S_b = 10.0$	**0.3891**	**0.4117**	**0.3449**	**0.4258**
$S_a = 1.0$ $S_b = 15.0$	**0.3920**	**0.4110**	**0.3546**	**0.4243**
$S_a = 1.0$ $S_b = 20.0$	**0.3778**	**0.4104**	**0.3140**	**0.4226**

表 9　マクロ平均が高かった結果同士の比較（対象 Easy）

パラメータ S_a, S_b	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
$S_a = 1.0$ $S_b = 10.0$	**0.4687**	**0.4887**	**0.4266**	**0.4478**
$S_a = 5.0$ $S_b = 15.0$	**0.4455**	**0.5126**	**0.3041**	**0.4442**
$S_a = 1.0$ $S_b = 20.0$	**0.4391**	**0.4819**	**0.3490**	**0.4454**

表 10　マクロ平均が高かった結果同士の比較（対象 Normal）

パラメータ S_a, S_b	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
$S_a = 1.0$ $S_b = 10.0$	**0.3344**	**0.3478**	**0.3089**	**0.3638**
$S_a = 5.0$ $S_b = 15.0$	**0.3248**	**0.3400**	**0.2957**	**0.3673**
$S_a = 1.0$ $S_b = 20.0$	**0.3377**	**0.3535**	**0.3076**	**0.3646**

表 11　マクロ平均が高かった結果同士の比較（対象 Hard）

パラメータ S_a, S_b	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
$S_a = 1.0$ $S_b = 10.0$	**0.1992**	**0.2130**	**0.1873**	**0.2106**
$S_a = 1.0$ $S_b = 15.0$	**0.1968**	**0.2027**	**0.1917**	**0.2037**
$S_a = 1.0$ $S_b = 20.0$	**0.1794**	**0.2166**	**0.1474**	**0.2075**

比較用のベースラインとして，文章中のすべての多義語に対して，取りうる語義の中からランダムに選択したものを付与するランダムベースラインを設定する．ランダムベースラインの難易度ごとの正解率を表 12 に示す．なお，この値は 1000 回の平均を取ったものである．

表 12　ランダムベースライン（1000 回の平均）

難易度	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
All	0.3097	0.3317	0.2666	0.3663
Easy	0.3301	0.3471	0.2944	0.3691
Normal	0.2935	0.3135	0.2555	0.3209
Hard	0.1347	0.1569	0.1157	0.1603

また，この結果を，既存手法である LDAWN を用いた日本語 WSD の結果と比較する．なお，実験結果[17]から，比較手法の LDAWN では一文章を一文書として扱う方法を採用し，遷移確率パラメータは各経路均等，$S = 10.0$としている．
このときの難易度別の正解率を表 13 に示す．([17]では 2 回実行した平均であるが，ここでは 3 回実行した平均を示す.）

表 13　トピックモデルで全体のマクロ平均とマイクロ平均が最高値となった手法の結果
（一文章を一文書，遷移確率パラメータ各経路均等，$S = 10.0$）

難易度	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
All	0.3612	0.3771	0.3302	0.4251
Easy	0.4206	0.4045	0.4546	0.4465
Normal	0.3066	0.3247	0.2723	0.3483
Hard	0.1352	0.1741	0.1017	0.1780

6. 考察

6.1　周辺語義モデルとランダムベースラインの比較

まず，周辺語義モデルによる各実験結果とランダムベースラインを比較する．周辺語義モデルの実験結果では，マイクロ平均が極端に悪いことがあり，そういった場合はランダムベースラインに劣る結果となることがあった．しかし，全体のマイクロ平均や名詞のマイクロ平均，マクロ平均は，ほぼすべての場合においてランダムベースラインより優れた結果となっている．

次に，全多義語に対するマクロ平均とマイクロ平均が最も良かった実験設定の結果をそれぞれ選択し，ランダムベースラインとの間でχ^2検定を行う．全多義語に対するマクロ平均が最も良かったのは$S_a = 1.0$，$S_b = 10.0$の場合であり，このときの難易度別の正解率は表 14 のようになっている．この結果とランダムベースラインとの間で難易度ごとにχ^2検定を行うと，Normal，Hard も含めてすべての難易度について有意水準 1%で有意差が認められた．全多義語に対するマイクロ平均が最も良かった場合について見てみると，マイクロ平均が最も良かったのは$S_a = 5.0$, $S_b = 20.0$の場合であり，このときの正解率は表 15 のようになっている．この結果とランダムベースラインとの間でχ^2検定を行うと，マクロ平均のとき同様，すべての難易度について有意水準 1%で有意差が認められた．以上の検定結果から，提案手法である周辺語義モデルは，ランダムベースラインより有意に優れており，WSD モデルとしての可能性は示せたと考える．

表 14　周辺語義モデルで全体のマクロ平均が最高値となった手法の結果
（$S_a = 1.0$，$S_b = 10.0$）

難易度	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
All	0.3891	0.4117	0.3449	0.4258
Easy	0.4687	0.4887	0.4266	0.4478
Normal	0.3344	0.3478	0.3089	0.3638
Hard	0.1992	0.2130	0.1873	0.2106

表 15　周辺語義モデルで全体のマイクロ平均が最高値となった手法の結果
（$S_a = 5.0$，$S_b = 20.0$）

難易度	マイクロ平均（全体）	マイクロ平均（名詞）	マイクロ平均（動詞）	マクロ平均
All	0.3960	0.4088	0.3710	0.4209
Easy	0.4890	0.4851	0.4974	0.4368
Normal	0.3285	0.3471	0.2931	0.3601
Hard	0.2395	0.2147	0.2609	0.2044

6.2　トピックモデルと周辺語義モデルの比較

トピックモデルの実験結果と，周辺語義モデルの実験結果について比較する．

トピックモデルのマクロ平均とマイクロ平均が最大のときの結果（表 13）と，周辺語義モデルのマクロ平均，マイクロ平均が最大のときの結果（表 14，表 15）を比べると，全体的にマイクロ平均，マクロ平均ともに周辺語義モ

デルの方が高い値を示している．特に Normal や Hard で差があり，周辺語義モデルはコーパス中で複数の語義を取る多義語に対して強い傾向が見られる．ただし，コーパスの構造上，入力が文書単位でないなどトピックモデルにやや不利だと思われる条件なので，トピックモデルより周辺語義モデルが優れているとは言い切れない．

トピックモデルの特徴として，Easy の対象語に対するマクロ平均がやや高いという点が挙げられる．Easy の多義語は，コーパス中でほとんど一つの語義が割り当てられている単語である．トピックモデルでは，特にパラメータSが小さいと周辺のトピックよりもトピックからの出現確率が重視される傾向にあるため，一つの語義が割り当てられやすい．この特徴が Easy における高いマクロ平均に寄与しているものと考える．一方で，そのような特徴は Hard のように様々な語義を取る多義語に対しては不利であり，実際 Hard のマクロ平均は低い．

6.3 周辺語義モデルによる語義の判別例

周辺語義モデルによってある程度語義を判別できた例として，「可能性」と「洗う」の二つの多義語の例を挙げる．

「可能性」は Hard の対象語であり，コーパス中の出現数は 18 回である．実行ごとに結果は変動するものの，「可能性」については安定して 70%弱の正解率が得られた．18 回の出現のうちのいくつかの場合について，「可能性」の語義，実際にコーパス中に出現した際の周辺の単語，システムの正誤結果を表 16 に示す．

表 16 「可能性」のコーパス中の周辺単語の例とシステムの正誤

語義	周辺の単語	正誤
物事をうまくやりこなすことのできる力	両者，人間	○
	研究，コンビナート，今後	○
実現できる見込み	毎日，違う，直面する，人々	×
	破る，音楽，広げる	×
起こりうる確実性の度合い	事態，生ずる，出る	○
	円高，進む，出る	○
	読む，否定する	○

結果を見ると，「物事をうまくやりこなすことのできる力」と「起こりうる確実性の度合い」は周辺の単語（語義）から正しく区別できていることが分かる．しかし，「実現できる見込み」についてはほとんど答えられなかった．周辺の単語が「毎日，違う，直面する，人々」に対応する実際の文章は，「都市は社会変化をつくり出すマシンであり，毎日違った可能性に直面しながら，人々はそこに生まれる出会いを求めているのではなかろうか」である．これに対して，システムは「物事をうまくやりこなすことのできる力」という答えを出力した．これは，「人々」という単語が「物事をうまくやりこなすことのできる力」の周囲に出現しやすかったためだと考えられる．また，「破る，音楽，広げる」に対応する文章は，「古来からの常識を破り，音楽の可能性を広げる意欲的な演奏活動をしている」であり，システムはこれに対しても「物事をうまくやりこなすことのできる力」と判断した．正しく答える手がかりは「広げる」にあると考えられるが，「音楽」という単語とイテレーション中での割り当て頻度から，上記の誤った答えを出力したものと考える．

もう一つの例として，「洗う」を挙げる．「洗う」は Normal の対象語であり，コーパス中の出現数は 5 回である．大体の実行において 80%（4 個）正解が取れるが，実行によっては 60%や 100%となることもある．「洗う」の語義，コーパス中での周辺単語，システムの正誤結果を表 17 に示す．

表 17 「洗う」のコーパス中の周辺単語の例とシステムの正誤

語義	周辺の単語	正誤
（心を）清らかにする	見る，心	○
	島民，涙，石	○
水で汚れを洗い落とす	今夜，体，否	×
	手足，顔，私，風呂	○
	体，水，抜く	○

語義は他に三つあるが，コーパス中に出現したのはこの二つの語義だけである．結果を見ると「清らかにする」と「水で汚れを洗い落とす」がほぼ区別できている．区別できなかったのは周辺単語が「今夜，体，否」の場合で，「清らかにする」を結果として出力してしまうことが多かった．「体」は他にも「水で汚れを洗い落とす」の周辺単語として出現しているが，「今夜」や「否」の影響により正しく取れなかったと考えられる．周辺単語が「島民，涙，石」の場合の文章は，「ときの町長，越森幸夫彫刻にしがみつき，島民の涙，石を洗う」であった．判断が難しい文ではあるが，「涙」という単語（語義）が「水で汚れを洗い落とす」の周囲に出現しにくいことを考慮できているのではないかと考える．

7. おわりに

本研究では，多義語の周辺に現れる語義の分布を利用する周辺語義モデルを提案し，これを用いて，日本語に対する教師無し WSD を行った．システムには EDR 電子化辞書による概念体系辞書を組み込み，実験は EDR の日本語コーパスを用いて行った．

実験では，EDR の日本語コーパスのうち，出典が日本経済新聞となっているものを実験用コーパスとし，それ以外をすべて事前学習用のコーパスとして利用した．システムはコーパス中のすべての対象語(名詞または動詞の自立語)に対して語義を一つ定め，その正解率をマイクロ平均とマクロ平均で評価する．また，各多義語に対して，コーパス中で使用されている語義のエントロピーによって難易度を三段階（Easy，Normal，Hard）設定し，より詳細な評価を行った．ベースラインとしては，語義をランダムに割り当てるランダムベースラインを設定した．

実験では，遷移確率パラメータの定数$S_a = \{1.0,\ 5.0,\ 10.0\}$，定数$S_b = \{10.0,\ 15.0,\ 20.0\}$と変化させ，計九通りの実験を行った．その結果，全多義語に対するマクロ平均が最大となったのは，$S_a = 1.0$，$S_b = 10.0$の場合であり，全多義語，Easy，Normal，Hard の各対象語に対して，マクロ平均はそれぞれ 42.58%，44.78%，36.38%，21.06%となった．また，全多義語に対するマイクロ平均が最大となったのは，$S_a = 5.0$，$S_b = 20.0$の場合であり，全多義語，Easy，Normal，Hard の各対象語に対して，マイクロ平均はそれぞれ 39.60%，48.90%，32.85%，23.95%となった．ランダムベースラインとの比較では，全多義語，Easy，Normal，Hard のすべての結果について有意水準 1%で有意に優れていた．以上の結果から，周辺語義モデルは，ランダムベースラインより優れていると言える．また，トピックモデルの実験結果とも比較したところ，コーパスの構造上，トピックモデルは多少不利な設定ではあるが, 全体的にマイクロ平均，マクロ平均ともに周辺語義モデルの方が高い値を示した．特に Normal や Hard で差があり，周辺語義モデルはコーパス中で複数の語義を取る多義語に対して強い傾向が見られた．

謝辞

本研究は，文部科学省科学研究費補助金[若手 B（No：24700138）]の助成により行われた．ここに，謹んで御礼申し上げる．

参考文献

1) Ted Pedersen，Satanjeev Banerjee，Siddharth Patwardhan：Maximizing Semantic Relatedness to Perform Word Sense Disambiguation，Research Report UMSI，(2005).
2) Jordan Boyd-Graber，David M. Blei，Xiaojin Zhu：A Topic Model for Word Sense Disambiguation，Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning，pp.1024-1033，(2007).
3) Weiwei Guo，Mona Diab：Semantic Topic Models: Combining Word Distributional Statistics and Dictionary Definitions，Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing，pp.552-561，(2011).
4) WordNet，http://wordnet.princeton.edu/
5) Eneko Agirre，David Martínez，Oier López de Lacalle，Aitor Soroa：Two graph-based algorithms for state-of-the-art WSD，Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing，pp.585-593，(2006).
6) Samuel Brody，Mirella Lapata：Bayesian Word Sense Induction，Proceedings of the 12th Conference of the European Chapter of the ACL，pp.103-111，(2009)
7) NiCT：EDR 電子化辞書，http://www2.nict.go.jp/out-promotion/techtransfer/EDR/J_index.html
8) Jay J. Jiang，David W. Conrath：Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy，Proceedings of International Conference Research on Computational Linguistics，pp.19-33，(1997)
9) Philip Resnik：Using Information Content to Evaluate Semantic Similarity in a Taxonomy，International Joint Conferences on Artificial Intelligence，pp.448-453，(1995)
10) 平川秀樹，木村和広：概念体系を用いた概念抽象化手法と語義判定におけるその有効性の評価，情報処理学会論文誌 Vol.44 No.2，pp.421-432，(2003).
11) Francesc Ribas：On Learning more Appropriate Selectional Restrictions，Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics，pp.112-118，(1995)
12) Diana McCarthy：Estimation of a Probability Distribution over a Hierarchical Classification，The Tenth White House Papers COGS – CSRP，(1997)
13) Good, I. J., The population frequencies of species and the estimation of population parameters, Biometrika 40,pp. 237-264, (1953).
14) Liu, Jun S.:The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem, Journal of the American Statistical Association, Vol.89, No.427, pp. 958–966 (1994).
15) MeCab, http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
16) 黒橋禎夫，白井清昭：SENSEVAL2 日本語タスク，電子情報通信学会言語とコミュニケーション研究会，pp.1-8，(2001)
17) 佐々木 悠人，古宮 嘉那子，小谷 善行：トピックモデルと概念辞書による日本語の語義曖昧性解消，第 5 回コーパス日本語学ワークショップ予稿集， pp. 71-80，(2014)