ハイパーメディアのJaccard係数に着目した定義文拡張による語義曖昧性解消

村田 亘　大沢 英一

自然言語処理の基本タスクの 1 つに語義曖昧性解消がある．これは，文中の多義語の語義を識別するタスクである．語義曖昧性解消は，一般に教師あり機械学習手法を用いて解決されているが，訓練データの作成コストが高く，対象単語が数百程度に限定されてしまうという問題がある．そこで，本研究では機械学習を用いず，知識ベースの語義曖昧性解消手法を提案する．具体的には，Wikipedia をコーパスとして記事 (語義) の本文 (定義文) の拡張を行い，Lesk アルゴリズムを適用する．Wikipedia はハイパーテキストをもつことから，記事をノード，リンクをエッジとする大規模ネットワークとして捉えられる．そこで，ネットワーク特徴量や共起指標を用いて，ある記事 (語義) と類似性の高い記事の選定を，2 つの手法で行った．さらに，それらの手法で選定された記事本文からキーワードを抽出する．抽出されたキーワードを用いて Lesk アルゴリズムを適用させたところ，75%の正解率であった．

Word Sense Disambiguation is one of the basic tasks of Natural Language Processing. This is the task which distinguishes multi-sense word in a sentence. Word Sense Disambiguation is solved by method using Supervised Machine Learning generally. But there are problems that cost for making training data is high and words of objects are limited only several hundred. Then, the present study tries Word Sense Disambiguation based knowledge not using Machine Learning. Specifically, the text (word sense) of articles (definition statement) is expanded using Wikipedia as a corpus, and Lesk algorithm is applied. Wikipedia can be regarded as a large scale-network that articles are nodes and links are edges because Wikipedia has link structure. Therefore the present study selects articles which have strong connectivity with other articles (word sense) using co-occurrence index and network feature quantity in two ways. Moreover keyword is extracted from the text selected by those ways. correct answer rate is about 75% applying Lesk algorithm using extracted keyword.

1　はじめに

近年，インターネットやスマートフォンの普及に伴い，多くの人々が SNS やブログなどを用いて情報を発信することができる．このことから，膨大な量の言語データがインターネット上に存在し，これらを分析することで，情報検索，情報抽出，テキストマイニング，文書要約，翻訳などの応用技術の研究が行われている．そういった研究は，自然言語処理の研究分野に属しており，「言葉がわかる」計算機システムの構築を目指している [11].

自然言語処理の基本タスクの 1 つに，語義曖昧性解消 (Word Sense Disambiguation) がある．これは，文中の多義語の語義を識別するタスクである．語義曖昧性解消は一般に教師あり機械学習手法を用いた研究が多くなされており，現在の教師あり機械学習の枠組みで処理すれば，90%程度の正解率に達すると言われている [11]．しかし，訓練データの作成コストが高く，対象単語が多くても数百程度に限定されていしまうという問題がある [11]．現実のアプリケーションではすべての単語を対象にする必要があるため．教師あり機械学習手法で解決することは難しい．そこ

Word Sense Disambiguation by Definition Statement Extension Focusing on Jaccard Coefficient in Hypermedia

Wataru Murata, 公立はこだて未来大学大学院システム情報科学研究科, Graduate School of Systems Information Science, Future University Hakodate.

Osawa Ei-ichi, 公立はこだて未来大学システム情報科学部複雑系知能学科, Department of Complex and Intelligent Systems, School of Systems Information Science, Future University Hakodate.

で，すべての単語に語義を付与する語義曖昧性解消は all-words WSD として，主に教師なし機械学習を用いて研究がされており，現在は 60〜70%の正解率である [11]．教師なし機械学習の手法には，辞書の定義文から語義の分散表現を求め，テストデータの文脈の分散表現との比較を行うものがある [2]

一方で，知識ベースの研究もなされており，古くからある知識ベースの手法に Lesk アルゴリズム [3] がある．これは，辞書の各語義の定義文に含まれる単語と，対象の文の単語との重複が一番多い語義に決定する手法である．しかし，Lesk アルゴリズムには，重複する単語が存在しない場合に，語義を決定できないという問題がある．これに対して Bashile らは，WordNet や Wikipedia などを含む多言語意味ネットワークである BabelNet をコーパスとし，定義文の拡張を行った [1]．Bashile らの研究では，拡張を行う語義に直接関係する単語を用いている．しかし，直接は関係がないが，拡張を行う語義と関係の強い単語は存在する可能性がある．そこで，本研究では BabelNet に含まれている Wikipedia を用いて，関係の強い単語の抽出を行う．

Wikipedia はウィキメディア財団が運営しているインターネット百科事典であり，現在では 298 言語が存在する．2017 年 7 月の段階では，日本語版 Wikipedia には約 106 万記事が登録されており，日に日に記事は増加している．また，幅広い分野において膨大な概念が網羅されているだけでなく，知識抽出のためのコーパスとして興味深い特徴を数多く持っている [7]．それらの特徴を利用し，情報検索や情報抽出，語義曖昧性解消などの様々な研究がされてきた．本研究では，Wikipedia の特徴の 1 つであるハイパーテキストを利用することで，語義の定義文の拡張のためのキーワード抽出を行い，それらのキーワードが語義曖昧性解消のキーワードとして有用であるか検証を行うことを目的とする．

2 Wikipedia の特徴

本研究では，Wikipedia の特徴を利用することでキーワード抽出を行う．以下に，本研究で着目した特徴を述べる．

まず，語義ごとに記事が存在する点に着目した．国語辞典などの一般的な辞典では，語義の説明を簡潔にしていることから，その語義がどのような場面，どのような単語と共に使われるのかをコンピュータに理解させるには情報量が少ない．しかし，Wikipedia の場合，一つ一つの記事が詳細に記述されているので，そういった問題は解決できると考えられる．また，記事タイトルの競合を防ぐため，片仮名でよく用いられる語を漢字で表したり，末尾に「(企業)」「(生物)」「(病気)」などを付け加えて記事タイトルを決められる場合がある．

次に，記事本文には他のハイパーテキストが多く存在する点である．記事をノード，他の記事へのリンクをエッジとするネットワークを構築すると，Wikipedia は大規模ネットワークであり，特に複雑ネットワークであることがわかっている [6]．Wikipedia の場合，次数分布がべき則に従っているスケールフリー・ネットワークである．Wikipedia をネットワークとして捉えることで，ネットワーク科学における特徴量を用いて解析できる．

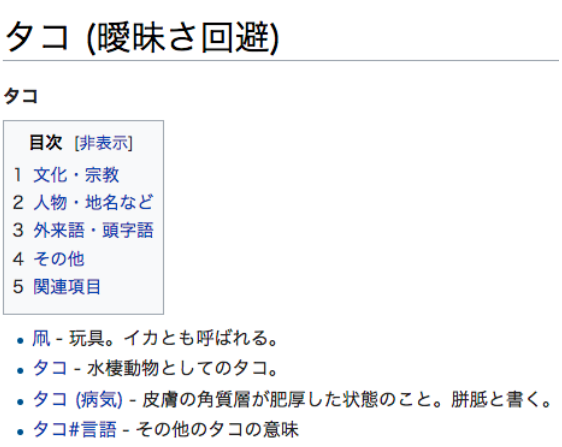
Wikipedia には曖昧さ回避ページというページがある．曖昧さ回避ページの例を図 1 に示す．これは「タコ」の曖昧さ回避ページで，「凧」「タコ」「胼胝」などの語義が載っている．図からわかるように，曖昧さ回避ページとは，同じ名前で異なる記事が存在する場合の一覧を表示しているページで，言い換えると多義語の語義のリストである．したがって，曖昧さ回避ページを用いることで多義語の抽出が可能である．

3 関連研究

本章では，本論文の内容に関連する研究について述べる．

3.1 BabelNet を用いた語義の定義文拡張と語義曖昧性解消

Bashile らは，辞書の語義の定義文に含まれる単語と，対象の文に含まれる単語の重複が存在しない場合に，Lesk アルゴリズムを適用できないという問題に対して，BabelNet を用いて定義文の拡張を行った [1]．BabelNet とは，WordNet や Wikipedia などの


タコ (曖昧さ回避)

タコ

目次 [非表示]
1 文化・宗教
2 人物・地名など
3 外来語・頭字語
4 その他
5 関連項目

- 凧 - 玩具。イカとも呼ばれる。
- タコ - 水棲動物としてのタコ。
- タコ (病気) - 皮膚の角質層が肥厚した状態のこと。胼胝と書く。
- タコ#言語 - その他のタコの意味

図 1　曖昧さ回避ページの例

コーパスが含まれた 271 言語に対応している多言語意味ネットワークである．

まず，BabelNet で多義語 w_i を検索し，j 番目の語義 s_{ij} を抽出する．s_{ij} の定義文を g_{ij} とすると，s_{ij} に関係する単語の定義文を重み付けし，g_{ij} に拡張する．拡張されたものを g^*_{ij} とする．

似た文脈に出現する単語は似た意味を持つという仮説をもとに，単語が持つ意味的な情報をベクトルで表現するベクトル空間モデルである，DSM(Distributional Semantic Model) を用いて，g^*_{ij} と対象の文脈の cos 類似度を算出し，一番高いものを語義に決定する．

ここで，単語 w_i の語義分布を計算する．WordNet で注釈がつけられている文書コレクションである SemCor を用いて，s_{ij} がどの程度用いられているのかの割合を算出する．これと，cos 類似度を線形結合し，一番高いものを語義に決定する．

実験では，語義分布を用いた場合と，用いない場合で比較を行った．対象言語は英語とイタリア語で行った．実験結果は，英語，イタリア語共に提案手法が MFS(Most Frequent Sense) よりも F 値が高く，特に語義分布を用いた場合が高かった．考察では，イタリア語は英語より F 値が低いことに対し，イタリア語の BabelNet に含まれるコーパスの質の低さを指摘している．

Bashile らの手法では，語義と直接関係している単語の語義を定義文に追加している．しかし，直接関係はないが語義と関係が強い単語も存在すると考えられる．本研究ではそういった単語を，BabelNet に含まれる Wikipedia のハイパーリンク構造を解析することで抽出することで，多くの定義文の拡張ができると考えている．

3.2　単語の分散表現を用いた語義曖昧性解消

Chen らは，語義の定義文を用いて語義を分散表現で表し，テストデータの分散表現と比較することで語義を識別する手法を提案した [2]．

語義の分散表現 (語義ベクトル) の求め方について述べる．まず，辞書の定義文に含まれる対象単語以外の内容語 (名詞・動詞・形容詞・副詞) を抽出し，すべての内容語と見出し語の分散表現を word2vec で求めた分散表現に変換し，ベクトルで表現する．そして，それぞれの内容語と見出し語の cos 類似度が，閾値以上の内容語を語義候補とし，その語義候補の平均ベクトルを，語義ベクトルとする．

次に，テストデータから文脈ベクトルを求める．テストデータの内容語を抽出し，語義ベクトルと同様に内容語の分散表現を word2vec を用いてベクトルで表現し，それらの平均を文脈ベクトルとする．

語義曖昧性解消には，語義ベクトルと文脈ベクトルを用いる．まず，各語義の語義ベクトルと文脈ベクトルの cos 類似度を算出する．そして，最も cos 類似度が高い語義に決定する．

本研究では以上の手法 (教師なし機械学習手法) と，本手法 (知識ベース手法) の比較を行う．

4　前提知識

本章では，本論文の内容で用いる技術や知識について，その概要を述べる．

4.1　クラスター係数

クラスター係数とは，あるノードの隣接ノードどうしが隣接ノード，つまり三角形の割合を示す指標である．ネットワークにおけるノード数を N，ノード v_i に隣接しているノードの数 (次数) を k_i とすると，v_i のクラスター係数 C_i と平均クラスター係数 C は以

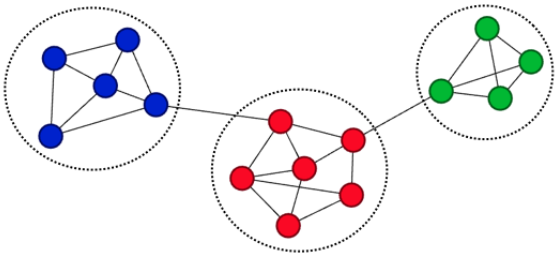


図 2　コミュニティの例

下の式で表される [9].

$$C_i = \frac{v_i\text{を含む三角形の数}}{k_i(k_i-1)/2}$$

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i$$

$0 \leq C_i \leq 1$，$0 \leq C \leq 1$ となり，C が高いとネットワーク全体が密であるといえる．

4.2　コミュニティ抽出

図 2 のように，同じ集団内ではエッジが密で異なる集団間にはエッジがあまりないネットワークはよく見られる [9]．点線内が 1 つの集団に対応し，これをコミュニティと呼ぶ．ここでは，コミュニティ抽出法である Newman 法と CNM 法について概要を述べる．

コミュニティ抽出の分割の良さを表す指標として，モジュラリティ Q が用いられ，Q が高いほど良い分割である．モジュラリティ Q は以下の式で表される．

$$Q = \sum_i (e_{ii} - a_i^2)$$

e_{ii} はネットワーク全体におけるコミュニティ内のエッジの割合，a_i はネットワークにおけるノード i のエッジの割合である．したがって，コミュニティ内のノード間のエッジが多く，コミュニティ外のノード間のエッジが少ない分割ほど Q は高くなる．

次に，Newman 法 [4] のアルゴリズムの概要について述べる．この際，N はノード数である．

(i)　ネットワークを N 個のコミュニティに分ける．すなわち，各ノードが 1 コミュニティを成す．

(ii)　全てのの 2 つのコミュニティの組み合わせに対して，Q を求める．

(iii)　一番大きい Q となる 2 つのコミュニティを結合する．

(iv)　(ii), (iii) をコミュニティの数が 1 個になるまで繰り返し行う．

(v)　コミュニティの数が N 個から 1 個になるまでの最も高い Q 値の分割を最終的なコミュニティとする．

大規模ネットワークのコミュニティ抽出を行う際，Newman 法では Q の計算量が膨大になってしまう問題がある．そこで，Clauset ら [5] は Q ではなく，コミュニティが結合したときの Q の増加量 ΔQ を用いることで，計算量を抑えた CNM 法を提案した．

本研究では Wikipedia の記事をノード，他の記事へのリンクをエッジとしたネットワークに対し，CNM 法を適用し，語義 (記事) と関係の強い記事の抽出を試みる．

4.3　Jaccard 係数

Jaccard 係数は，2 つの集合間の類似性を表す指標である．X と Y の Jaccard 係数は以下の式で表される．

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

本研究では，ノード v_i とノード v_j に隣接しているノードをそれぞれの集合とし，$Jaccard(v_i, v_j)$ を算出する．つまり，v_i と v_j が互いに共通のノードと隣接しているほど Jaccard 係数は高くなる．

また，単語 X と単語 Y の Jaccard 係数は，$|X \cup Y|$ が X または Y を含む文の数，$|X \cap Y|$ が同一文中に X と Y を含む文の数で算出される．

4.4　Wikipedia のナビゲーション情報の抽出

千田らは，Wikipedia においてユーザが求めている有用な情報にたどりつくため，Wikipedia のリンク構造からコミュニティの同定を行うことでナビゲーション情報の抽出手法を提案した [8]．コミュニティとはネットワーク構造の中でリンクが密な部分集合のことである．本研究では，千田らの手法を参考にし，ある記事 (語義) において，類似性の高い記事の抽出を行う．これらの記事を解析することで，語義曖昧性

解消に有用なキーワードを抽出できると考えている．

まず，ソース記事 s に対し距離 2 のサブネットワークを構築し，Jaccard 係数とクラスター係数を用いて類似性が高く，密なネットワークを抽出した．距離とはノード v_i とノード v_j の最短距離のことである．具体的な手法は 5 章で述べる．

4.5 Small World 構造を利用した文書からのキーワード抽出

松尾らは，単語間の共起を用いて Small World 構造を持つ共起ネットワークを構築し，キーワード抽出を行う手法を提案した [12]．共起ネットワークのノードには，文書の中で規定回数 f_0 回以上出現した単語をノードに追加する．次に，同一文中の共起が大きい単語どうしにリンクを張る．共起指標には Jaccard 係数を用いており，上位から規定値 k_0 に平均次数が達するまでリンクを張る．

次に，構築されたネットワークの 1 つのノードに対する contribution を，平均経路長 L の定義を非連結の部分に拡張した extend path length を利用して算出した．extend path length の定義を以下に示す．

$$d'(i,j) = \begin{cases} d(i,j) & if\ (i,j)\ are\ connected \\ w_{sum} & otherwise \end{cases}$$

w_{sum} は非連結のサブネットワークの幅の和である．ノード v を取り除いた全てのノードの組についての extend path length の平均 L'_{G_v} と，ノード v 以外の全てのノードの組についての extend path length の平均 L'_v を算出する．

L'_v の計算ではノード v はネットワークに接続されているが，平均には含めない．L'_{G_v} はノード v と v を含むリンクが除外される．ノード v の contribution である CB_v は以下の式で定義する．

$$CB_v = L'_{G_v} - L'_v$$

実験では，7 著者 20 論文に対してキーワード抽出を行ったところ，make のような一般的に使用される単語がキーワードとして抽出されてしまった．そこで，一般的に使われる語を取り除く目的で，CB_v と $idf(v)$ の積でキーワード抽出を行うと，良い性能が得られた．

本研究では，Wikipedia の文書に対して松尾らの手法を適用し，キーワード抽出を試みる．

5 提案手法

提案手法の手順の概要について述べる．

(i) ソース記事 s に対して被リンク先に距離 2 の無向ネットワークを構築

(ii) Jaccard 係数とクラスター係数を用いた記事の選定

(iii) 選定された記事 (Selected Articles) から，共起ネットワーク (Co-occurrence Network) の構築

(iv) キーワード抽出 (Keyword Extraction)

(v) 語義曖昧性解消 (Word Sense Disambiguation)

手順 (i)，(ii) は千田ら [8]，手順 (iii)，(iv) は松尾ら [12] の提案手法を参考にしている．

また，本論文でのネットワークは，「Wikipedia の記事をノード，他の記事へのリンクをエッジとする無向ネットワーク」とする．

5.1 手順 (i)：ネットワークの構築

ソース記事 s(語義) から被リンク先に対して距離 2 のネットワークを構築する．被リンク先とは，ソース記事へリンクを張っている記事のことである．ただし，被リンク先が存在しない場合はソース記事 s のみを用いる．図 3(a) は構築されたネットワークの例である．矢印はある記事が他の記事へリンクを張っていることを表し，被リンク先に距離 2 のネットワークは点線内のネットワークとなる．

5.2 手順 (ii)：Jaccard 係数とクラスター係数を用いた記事の選定

手順 (i) で構築されたネットワークからエッジを除去し，図 3(b) のようにソース記事 s と類似性の高い記事を選定する．ここで，2 つの方法を検討する．1 つ目 (「記事選定手法 A」と表記) は千田ら [8] の手法の記事の選定方法を変更した方法である．2 つ目 (「記事選定手法 B」と表記) は千田ら [8] の手法をそのまま適用し，記事の選定を行う方法である．

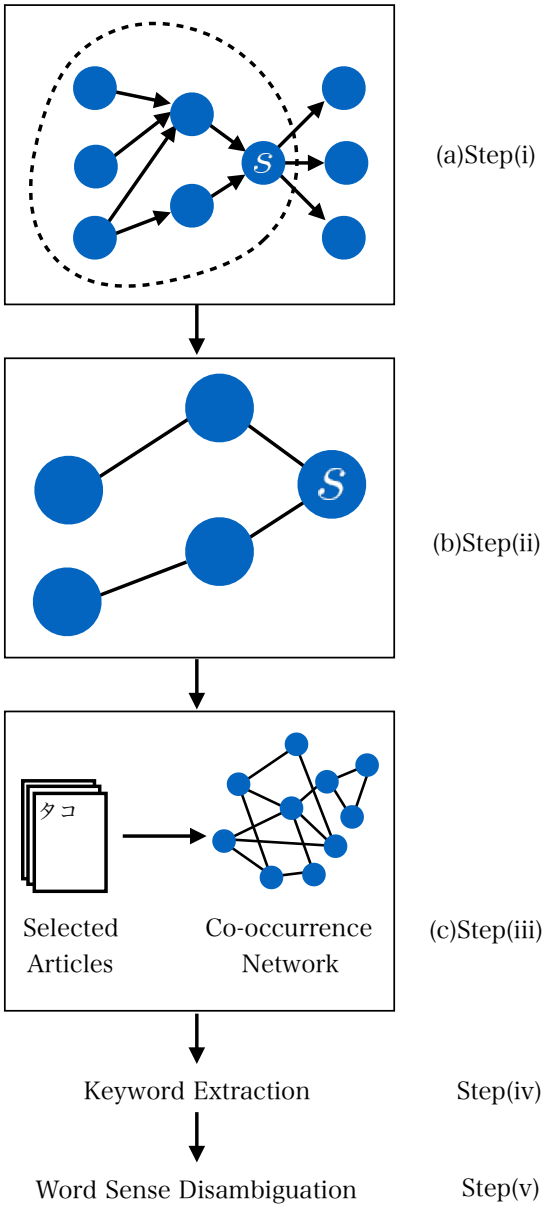


図 3　提案手法の手順

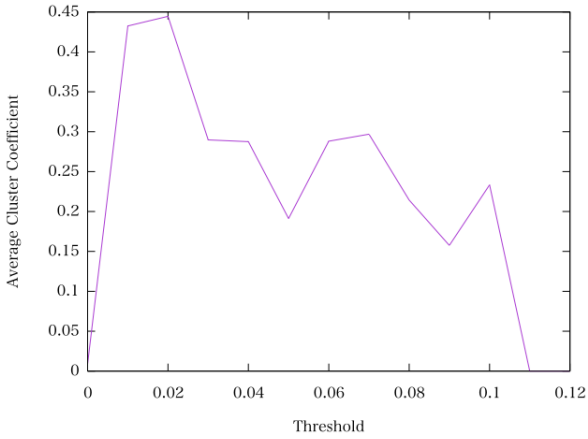


図 4　平均クラスター係数の推移の例 (記事「錠剤」)

具体的には，記事選定手法 A では Jaccard 係数と平均クラスター係数を利用し，記事選定手法 B では Jaccard 係数とソース記事 s のクラスター係数，CNM 法を利用して記事の選定を行う．

5.2.1　記事選定手法 A

まず，手順 (i) で構築されたネットワークの全てのエッジに対して Jaccard 係数を算出し，閾値を 0.01 に設定する．次に，Jaccard 係数が閾値未満であればエッジを除去し平均クラスター係数を算出する．ただし，エッジの本数が 0 本となったノードは除去する．さらに閾値を 0.01 上げ，エッジの除去，平均クラスター係数の算出を繰り返す．そして，平均クラスター係数が最後に極大をとる Jaccard 係数を最終的な閾値に決定し，残った記事を選定された記事とする．

図 4 は記事「錠剤」に対して被リンク先に距離 2 のネットワークを構築し，Jaccard 係数の閾値を上げたときの平均クラスター係数の推移である．この場合，最後に極大をとっている 0.1 を最終的な閾値に決定する．また，閾値を設けることで 4277 個あったノードが，9 個に減った．最後の極大にする理由は，Jaccard 係数が高いため類似性が高い記事が残り，平均クラスター係数が高いためネットワークが密であるからである．

5.2.2　記事選定手法 B

記事選定手法 A は平均クラスター係数を用いているため，ソース記事 s が除去され，選定されない問題がある．ソース記事 s との関係性の強い記事が重要な記事であると考えられるため，千田ら [8] の手法を用いて記事の選定を行う．

まず，全てのエッジに対して Jaccard 係数を算出し，閾値を 0.01 に設定する．次に，Jaccard 係数が閾値未満であればエッジを除去しソース記事 s のクラスター係数を算出する．ただし，エッジの本数が 0

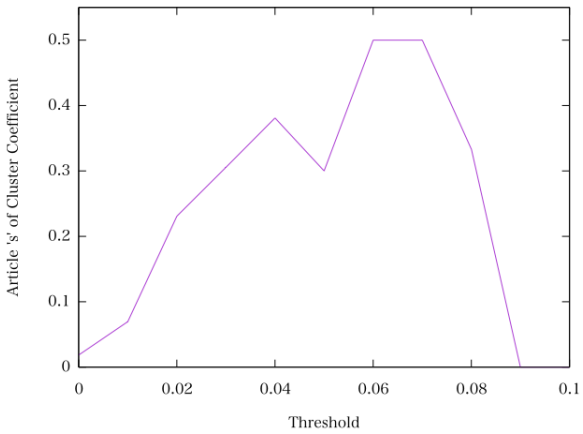


図 5　クラスター係数の推移の例 (記事「錠剤」)

表 1　キーワード抽出の例 (記事「錠剤」)

	CB_v	$CB_v \cdot idf(v)$
1	錠	錠
2	剤	作用
3	錠剤	剤
4	作用	トローチ
5	服用	錠剤
6	トローチ	ベンザ
7	製剤	服用
8	医薬品	製剤
9	カプセル	医薬品
10	ベンザ	カプセル

本となったノードは除去する．さらに閾値を 0.01 上げ，エッジの除去，ソース記事 s のクラスター係数の算出を繰り返す．そして，ソース記事 s のクラスター係数が最後に極大をとる Jaccard 係数を最終的な閾値に決定し，ネットワークを構築する．次に，CNM 法を適用し，コミュニティ抽出を行い，ソース記事 s が含まれるコミュニティに含まれる記事を選定する．

図 5 は記事「錠剤」に対して被リンク先に距離 2 のネットワークを構築し，Jaccard 係数の閾値を上げたときのソース記事 s のクラスター係数の推移である．この場合，0.06 を最終的な閾値に決定する．また，この場合はノードが 18 個に減った．さらに，CNM 法を適用させ，ソース記事 s が含まれるコミュニティには 6 個のノードが存在した．

5.3　手順 (iii)：共起ネットワークの構築

選定された記事本文を全て形態素解析し，同一文中に現れる名詞どうしの Jaccard 係数を算出する．本研究では出現回数の上位 200 件の名詞をノードとした．上位 200 件にした理由は，選定された記事本文の名詞の数が，少なくとも 200 件以上存在していることを確認したためである．

次に，2 つのノードに対応する名詞の共起が多いものにリンクを張る．上で算出した Jaccard 係数が上位のものから平均次数 $k = 4.0$ に達するまでリンクを張る．

5.4　手順 (iv)：キーワード抽出

構築された共起ネットワークから，キーワード抽出を行う．松尾ら [12] の提案手法でノード v の contribution を算出する．本研究では CB_v と $CB_v \cdot idf(v)$ をそれぞれ用いて contribution とし，キーワード抽出を行う．ただし，$idf(v)$ を求める際の記事集合は，選定された記事全てとする．

表 1 は，記事「錠剤」に対して提案手法を適用させたときの，抽出されたキーワードの contribution 上位 10 件である．

5.5　手順 (v)：語義曖昧性解消

抽出されたキーワードの contribution 上位 t 件を用いて Lesk アルゴリズムを適用する．ただし，contribution の値が 0 のものは除外する．除外した結果，キーワードが t 件存在しない場合は，除外後の全ての名詞を用いる．

Lesk アルゴリズムは，曖昧性を解消する語が含まれる文章の名詞集合と，キーワード集合の重複が多い語義を選ぶ．

6　実験

本手法が語義曖昧性解消に有用であるかの実験を行った．詳細を以下に示す．

6.1 実験方法

抽出されたキーワードの contribution 上位 t 件を用いて，Lesk アルゴリズムを適用し，語義曖昧性解消を行う．まず，提案手法の記事選定手法 A(「手法 A」と表記) と，記事選定手法 B(「手法 B」と表記) で選定された記事の本文から contribution を算出する．contribution は CB_v と $CB_v \cdot idf_v$ の 2 種類を用い，上位 50 件 ($t = 50$) または上位 100 件 ($t = 100$) を正解率で評価する．正解率は以下の式で算出する．

$$\text{正解率} = \frac{|\ \text{正解データ} \cap \text{提案手法の結果データ}\ |}{|\ \text{正解データ}\ |}$$

6.2 テストデータ

実験に用いる多義語は，曖昧さ回避ページから抽出した．実験は，多義語を含むパラグラフに対して語義曖昧性解消を行う．文書はアメーバブログから各語義に対して 10 件ずつ抽出し，それぞれパラグラフを抽出した．ブログを対象にした理由は，専門的な知識を含む内容から一般的な内容まで存在するからである．

本実験では，多くの分野で本手法が有効であるか検証するため，goo 国語辞書に収録されているデジタル大辞泉を用いた．デジタル大辞泉は約 28 万 4,400 項目を収録しており,「文学」「宗教・思想」「日本史」「世界史」「地理」「社会」「美術・音楽」「演劇・映画」「物理・化学」「生物」「地学」「医学」「生活」「IT 用語」「数学」のカテゴリに分けられている．これらの 15 カテゴリと曖昧さ回避ページを用いて多義語を選択した結果，表 2 のようになり，多義語の総数は 40 個，語義の総数は 91 個となった．

6.3 比較手法

本実験では，Lesk アルゴリズムと単語の分散表現を用いた手法を用いて，本手法との比較を行う．詳細を以下に示す．

6.3.1 Lesk アルゴリズム

定義文の拡張を行わず，語義の定義文の名詞を全て用いて Lesk アルゴリズム (「Lesk」と表記) を適用し，本手法の定義文拡張が有用であるかを評価する．Lesk アルゴリズムを適用する際，Wikipedia の記事本文と，デジタル大辞泉の定義文を用いた．

表 2　カテゴリと本実験で用いる語義の数

カテゴリ	個数
文学	6
宗教・思想	5
日本史	5
世界史	5
地理	6
社会	5
美術・音楽	6
演劇・映画	6
物理・化学	6
生物	5
地学	5
医学	5
生活	11
IT 用語	10
数学	5

6.3.2 単語の分散表現

3.2 章で述べた手法 (「w2v」と表記) を適用するための，単語の分散表現の作成方法について述べる．

日本語版 Wikipedia の 2016 年 8 月のダンプファイルをコーパスとし，word2vec を用いて分散表現を求めた．学習モデルは Continuous Bag-of-Words(C-BoW)，次元数は 200，ネガティブサンプル数を 5 とした．

文脈ベクトルを求める際に使う辞書の定義文には，Wikipedia の記事本文と，デジタル大辞泉の定義文を用いた．また，語義ベクトルを求める際の閾値は，最も正解率の高かった 0.0 とした．

6.4 実験結果

Lesk と w2v の実験結果を表 3，本手法の実験結果を表 4 に示す．Lesk は Wikipedia の記事を定義文とした場合が goo 国語辞典を定義文とするよりも平均正解率が 10%高く，w2v はどちらでも平均正解率の差は見られなかった．

本手法は手法 A，手法 B 共に CB_v と $CB_v \cdot idf(v)$ のどちらの方法で抽出されたキーワードでも，上位

表 3　Lesk と w2v の平均正解率 (%)

	平均正解率 (%)
Lesk(goo)	55.7
Lesk(Wikipedia)	65.0
w2v(goo)	40.9
w2v(Wikipedia)	40.8

表 4　本手法の平均正解率 (%)

	平均正解率 (%)	
	手法 A	手法 B
CB_v (上位 50 件)	74.6	73.2
CB_v (上位 100 件)	71.8	69.1
$CB_v \cdot idf(v)$ (上位 50 件)	72.0	70.1
$CB_v \cdot idf(v)$ (上位 100 件)	71.6	69.0

50 件の平均正解率が高かった．その中でも CB_v が最も高い平均正解率であった．

6.5 考察

表 3，表 4 より，手法 A の CB_v 上位 50 件が一番良い平均正解率であった．手法 A，手法 B 共に CB_v と $CB_v \cdot idf(v)$ のどちらであっても，上位 50 件の方が平均正解率が高い．これは，contribution が上位のものほど，キーワードとして正しいものが抽出されていることがわかる．

CB_v が $CB_v \cdot idf(v)$ よりも平均正解率が高い理由は，選定された記事を記事集合として $idf(v)$ を算出しているからであると考えられる．選定された記事どうしの Jaccard 係数高いため，類似度が高い．類似度が高い記事集合で idf を算出すると，重要なキーワードの idf が低い値となり，contribution を下げていると考えられる．

次に，必ず記事 (語義) が存在する手法 B が手法 A より平均正解率が低いのは，記事が極端に多いことや，Jaccard 係数が低いことが原因だと考えられる．表 5 に，手法 A と手法 B で選定された記事数の平均を示す．

手法 A の方が，選定された記事数が手法 B よりも少ない．選定された記事数が多いほうが，定義文の拡張ができると考えられるが，実際に選定された記事の中には記事 (語義) と関係が弱い記事も含まれていると考えられる．なぜなら，図 4(閾値 : 0.1) と図 5(閾値 : 0.06) のように，手法 A で選定した場合のほうが Jaccard 係数が高いからである．したがって，手法 A では，記事 (語義) との類似度が高い記事が選定された．

表 5　選定された記事数の平均

	手法 A	手法 B
記事数の平均	16.7	58.0

表 6　多義語「移植」の Lesk(Wikipedia) の名詞の数と正解率 (%)

語義 (Wikipedia 記事名)	名詞数	正解率 (%)
移植 (医療)	817	100
移植 (ソフトウェア)	459	80
移植 (生物)	231	10

Lesk は，その記事 (語義) の本文の全ての名詞を用いて，Lesk アルゴリズムを適用している．それらの名詞の中には，語義の特徴として重要な名詞を含んでいることは間違いないが，語義と関係のない名詞も多く含んでいると考えられる．

また，名詞の数が多ければ正解率が高く，少なければ正解率が低くなる可能性がある．ここでは，Wikipedia の本文を定義文とした場合について述べる．名詞の数は，少ないもので 100 程度，多いもので 3000 程度あった．表 6 は，多義語「移植」の各記事 (語義) に含まれる名詞の数と，正解率である．このように，名詞の数が多ければ多いほど，様々な名詞を含むため，正解率が高くなっていることがわかる．これは，本実験で行った語義曖昧性解消の全てに見られた．このため，名詞の数が多ければ良いというわけではなく，語義の特徴を掴む名詞のみを使用する必要がある．

本手法で正解率が低かった場合は，一般的な内容のブログのパラグラフであった．本手法では，特に専門

用語が多く抽出されていたため，一般的な内容の正解率が低くなったと考えられる．したがって，専門用語だけでなく，語義と共に使われることの多い一般的に用いられる用語の抽出が正解率の向上につながるだろう．しかし，Wikipedia はその語義の説明を詳細にしているため，一般的に用いられる用語の抽出は困難であると推測する．そのため，新たなキーワード抽出手法を模索する必要がある．

6.6 おわりに

本論文では，日本語版 Wikipedia のハイパーリンク構造を利用することでネットワークとして捉え，語義と関係性の高い記事を 2 つの手法で選定し，それらの記事本文からキーワード抽出を行うことで，語義の定義文の拡張を行った．実験より，本手法の最も正解率が高かった手法は，定義文の拡張を行っていない Wikipedia を用いた Lesk アルゴリズム [3] よりも正解率が 10% 高く，単語の分散表現を用いた関連研究 [2] よりも 35% 高かった．しかし，実験では一部の多義語を対象にしていたため，他の多義語に対して実験を行い，本手法の有用性を検証する必要がある．

今後は，実験データとして SemEval-2010 に対して語義曖昧性解消を行う予定である．また，本手法では，語義曖昧性解消をする際に，contribution の値を用いていないため，1 位であっても 50 位であっても，同じ重要度で扱っている．contribution の値が高いものほど，重要なキーワードであると考えられるため，キーワードに重みを付けることで，正解率が向上すると考えている．

さらに，本論文の実験では，語義が一般的に用いられる場合に正解率が低かったので，そういった用語の抽出方法を検討したい．

参 考 文 献

[1] Bashile, P., Caputo, A., and Semeraro, G.: An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model, *Proceedings of COLING 2014: Technical Papers*, 2014, pp. 1591–1600.

[2] Chen, X., Liu, Z., and Sun, M.: A Unified Model for Word Sense Representation and Disambiguation, *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1025–1035.

[3] Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, *the 5th annual international conference on Systems documentation*, 1986, pp. 24–26.

[4] Newman, M. E. J.: Fast algorithm for detecting community structure in networks, *Physical Review E*, Vol. 69, No. 066133(2004).

[5] Newman, M. E. J. and Girvan, M.: Finding and evaluating community structure in networks, *Physical Review E*, Vol. 69, No. 026113(2004).

[6] Zlati, V., Boievi, M., H.tefani, and M.Domazet：: Collaborative web-based encyclopedias as complex networks, *APS Physics*, 2006.

[7] 中山浩太郎: Wikipedia マイニングによる大規模 Web オントロジの実現, 人工知能学会全国大会, 2008.

[8] 千田俊輔, 大沢英一: リンク構造解析による Wikipedia のナビゲーション情報の抽出, 合同エージェントワークショップ&シンポジウム *JAWS2010*, 2010.

[9] 増田直紀, 今野紀雄: 複雑ネットワーク基礎から応用まで, 近代科学社, 2010.

[10] 大橋勝揮, 小林暁雄, 増山繁: Wikipedia を用いた段階的語釈文拡張手法による語義曖昧性解消, 言語処理学会 第 *20* 回年次大会, 2014, pp. 314–317.

[11] 新納浩幸: 自然言語処理の現状と展望 語義曖昧性解消, 情報処理学会誌, Vol. 57, No. 1(2016), pp. 20–21.

[12] 松尾豊, 大澤幸生, 石塚満: Small World 構造に基づく文書からのキーワード抽出, 情報処理学会誌, Vol. 43, No. 6(2002), pp. 1825–1833.