ワードネットを用いた日本語テキストの語義曖昧性解消

2013

修士(工学)

情報・知能工学専攻

栗林孝行
125303

豊橋技術科学大学

2014年　1月　8日

情報・知能工学専攻	学籍番号	M125303
申請者氏名	栗林孝行	

指導教員氏名	井佐原均

論　文　要　旨(修士)

論文題目	ワードネットを用いた日本語テキストの語義曖昧性解消

多くの語は2つ以上の意味を持っている。人間はそれまで培ってきた文法的知識や語彙的知識等によってそれぞれどの意味で用いられているか判別できるが、コンピュータにとっては難しい場合が多い。

この「文中の語がどの意味で使われているか」をコンピュータに特定させることは語義曖昧性解消(Word Sense Disambiguation, WSD)と呼ばれ、機械翻訳や情報検索等に用いられる基礎的な技術となっている。

WSD はいわば入力文中の語と辞書の語義とを結び付けるもので、本研究では日本語ワードネット(JWN)を辞書として用いた。その中で概念は、それを表すのに用いられる同義語(synonym)の集合である synset という単位で扱われる。また各 synset は他の概念と何らかの関係性をもってリンクし、ネットワークを成している。

本研究では、まず JWN 1.1 の異表記への対応を行った。既存の日本語辞書から異表記情報を抽出して同じ語の異表記と考えられるものを異表記セットにまとめ synonym を各異表記セットに振り分けた。この過程で新たな表記が追加され、コーパスのカバー率についても6 ~ 20 ポイントの向上を見ている。

ワードネットは synset をノード、synset 間リンクをエッジと見なすことができるため、グラフ理論で捉えることが可能である。そこで、本研究ではグラフ理論に基づいている Personalized PageRank を JWN のネットワーク上で実行する手法を採用した。実際には、本研究では Personalized PageRank を実装した ukb_wsd というツールを利用した。これは各入力文ごとに、全ての内容語の原形と品詞情報から WSD を行うものである。

また、本研究では WSD の実験時の入力と検証のため、また語義頻度情報を得るために手動で JWN 1.1 の語義データを付与した既存のコーパスを用いた他、異表記対応を行った JWN を用いて自動で語義データを付与した新たなコーパスを作成した。前者はエッセー、小説、新聞文が、後者は JWN の定義文・例文が元になっている。

実験はエッセーコーパスを入力とし、ukb_wsd を用いて行った。その結果を検証すると、ukb_wsd 単体では充分正解率が得られなかったが、以下の変更を適用すると改善した。

1. コーパスの英語版のアノテーションデータを日本語版に対応させたものも併せて用いる
2. コーパスから得た品詞情報に拘らない
3. 各コーパスから得た語義頻度情報によって最初から語義の候補を絞り込む

最も改善が見られたのが上記全て、特に 3. については小説から得た頻度情報を適用したものだった。これは入力コーパスのドメインが小説に近いものである点が要因として考えられる。

DATE: 2014/01/08

Department of Computer Science and Engineering	ID	125303
Name	Takayuki Kuribayashi	

Supervisor	Hitoshi Isahara

Abstract

Title	Japanese Word Sense Disambiguation Using Wordnets

Most words have more than 1 meaning. It's kind of difficult for computers to distinguish the meaning which is used for a word in a sentense while human can do it. The technique to do it is called "Word Sense Disambiguation" (WSD) and it's a fundamental one used for machine translation, information retrieval and so on.

In other words, WSD is a technique to tie a word in sentence with a sense in a dictionary. In this study we make use of the Japanese Wordnet (JWN) for the dictionary. Concepts are called "synsets" that are represented by sets of "synonym" and they form a network in JWN.

At first, I expanded JWN 1.1 so that it can cover more orthographic variants. The coverage of coepora improved after the expantion.

Then we decided to adopt Personalized PageRank which is based on Graph theory for WSD. The reason to choose it is you can put synsets as nodes and put links between synsets to edges. In actual experiments we run ukb_wsd in which Personalized Pagerank is implemented.

For datasets, we utilized Japanese corpora tagged with JWN 1.1 senses by hand as well as corpola newly auto-tagged with expanded JWN senses in this study. hand-tagged ones are derived from novels, an essay and newspaper articles. auto-tagged ones are derived from JWN definitions and example sentences.

we conducted experiments with ukb_wsd and the essay corpus, and we found original ukb_wsd didn't perform well. Therefore we applied following modification.

1. Combine Japanese annotationdata and English annotation data.
2. Don't be constrained by POS information in the corpus
3. Apply Most Frequent Sense (MFS) that is obtained from corpora

After applying all of them the accuracies improved and the results showed us that MFS obtained from novel corpola was the most efficient.

目次

1. はじめに

a）ドライバーでねじを緩めた
b）ドライバーを雇う
c）ドライバーで飛ばす

以上の 3 文について、人間はそれまで培ってきた文法的知識や語彙的知識等によってそれぞれの「ドライバー」がどのような意味で用いられているか判別できるが、コンピュータにとっては難しい場合が多い。

この「文中の語がどの意味で使われているか」をコンピュータに特定させることは語義曖昧性解消（Word Sense Disambiguation, WSD）と呼ばれ、自然言語処理分野において機械翻訳や情報検索等の精度向上に資する基礎的な技術となっていおり、その重要性は Senseval シリーズのような WSD の技術コンテストが開催されることからも明らかである。

WSD のための技術はいくつかあるが、 前もっての正解データの蓄積が必要であったり、計算量が膨大であったりする場合が多い。また、 WSD は文中の語と辞書の語義を結びつけるものであるが、その辞書が利用・ WSD 結果の頒布等に制限がある場合もある。そこで本研究では大規模かつオープンな構造付き概念辞書である日本語ワードネットのバージョン 1.1 と、 Personalized PageRank を実装した軽量な ツールである ukb_wsd を利用して日本語文に対して WSD を行った。

ukb_wsd への入力や結果の検証、また語義等の頻度情報を得るために新聞記事、エッセー、小説のコーパスを利用したが、より多くのデータ獲得のために、本研究で作成した異表記対応版 JWN を用いて、 JWN の定義文・例文に語義情報を付与したコーパスの作成も行った。

WSD 実験においてはオリジナルの ukb_wsd では充分な正解率が得られなかったため、英語版のアノテーション結果を導入する、予め対象語に付与されている品詞情報に依存しないようにする、コーパスから得た MFS（Most Frequent Sense, 最頻出語義）によって語義候補を最初から絞り込むという改良を行った結果、最も高い場合の正解率は 0.4514 となった。

本稿の構成として、第 2 章で日本語ワードネットの全般的な説明と、本研究で行った異表記対応について述べる。第 3 章で WSD の先行手法について述べ、第 4 章でそれを踏まえ本研究で用いる手法とツールについて述べる。第 5 章で本研究で用いるデータセットの紹介を行い、またその一部である定義文・例文コーパスの本研究での構築について述べる。第 6 章では実験、第 7 章では結果の考察、第 8 章でまとめと今後の課題について述べる。

2. 日本語ワードネット

2.1. 日本語ワードネットの概要

筆者も開発に参加している日本語ワードネット(JWN)は 独立行政法人 情報通信研究機構(NICT) から公開されている[1]、リンク構造付きのオープンな概念辞書である。これは米プリンストン大学による Princeton WordNet[2] (PWN) のバージョン 3.0 をもとに半自動的に構築されている。

その中で概念は、それを表すのに用いられる語、いわゆる同義語（synonym）の集合である ”synset”いう単位で存在し、各概念は他の概念と何らかの関係性（例えば hypernym-hyponym: 上位下位関係）をもってリンクしネットワークを構成している。

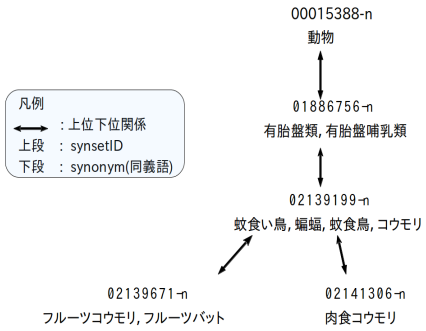


図 1: synset 間ネットワーク概略図

JWN は 2009 年に主に日本語 synonym とリンク関係を主体とした バージョン 0.9 が公開され、バージョン 1.0 からは Princeton WordNet 3.0 の synset についての全ての定義文と例文の日本語訳も追加された。現状の最新バージョンは 1.0 のエラーデータを修正したバージョン 1.1 である。

ライセンスは BSD スタイルのもので、 PWN と同様であり、利用・複製・改変・再配布が自由となっている。

バージョン 1.1 の規模は以下のとおりである。

- **概念 (synset) 数:　57,238**
- **語数 :　93,834**
- **語義 (synsetと単語のペア) 数: 158,058**
- **定義文数:　135,692**
- **例文数:　48,276**

また、動物の「蝙蝠」を指す synset を例にスクリーンショットを図 2 に示す。JWN では、PWN 3.0 の synset と synset 間リンク構造をそのまま利用しているため、PWN 3.0 の情報も同時に示している。

スクリーンショットの一行目はその synset の ID であり、8桁の数字と品詞を表すアルファ

1 http://nlpwww.nict.go.jp/wn-ja/
2 http://wordnet.princeton.edu/

ベット一文字で構成される。 品詞については、

a : 形容詞、形容動詞
n : 名詞
v : 動詞
r : 副詞

という対応になっている。

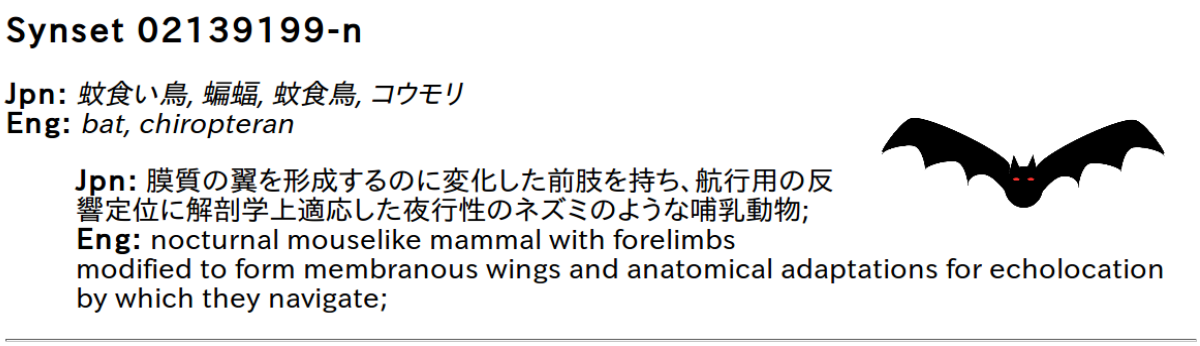


図 2 : JWN 1.1 スクリーンショット

図 2 の二行目が JWN の synonym（同義語）、三行目が PWN 3.0 の synonym であり、インデントされた五行目以降が定義文、”Hype” や ”Hypo” 等は他 synset とのリンク情報である。
また、右上の絵はクリップアート、最下行は他の言語資源との対応情報である。
これらの詳細は http://nlpwww.nict.go.jp/wn-ja/jpn/detail.html に掲載されている。

WSD をはじめ言語処理に JWN を用いる利点としては以下が挙げられる。

I. オープンなライセンスのもとで公開されているため、誰でも入手できる
   - JWN を用いて作成したデータも自由に公開できる
II. Princeton WordNet をもとに多数の言語のワードネットが構築されている
III. synset 間ネットワークが利用できる

JWN の構築は、PWN の synset と、他言語のワードネットにおいて同一のものを指す synset (equivalent synset) の synonym を対日辞書で翻訳し、スコアリングや人手チェックによって信頼性が高いとされた語を日本語の synonym として登録している。

2.2. 異表記対応

現在の公開バージョン 1.1 において問題であったのが、日本語における表記バリエーションの豊富さに充分対応できていなかった点である。そこで本研究において異表記対応を行った手元版 JWN を作成した。

また、この異表記対応結果は次版の JWN に反映される予定である。

2.2.1. 異表記対応の概要

日本語には字種や送り仮名等の違いで、同じものを指す語で、同じ読みであっても書き表し方が何種類もある場合が多いが、ときに言語処理においては悪影響が現れる。 JWN において解決すべき主な問題は以下の 3 つであった。

I. コーパスアノテーション時のカバー率が若干低下する
あるコーパスに出現した「あやうい」や「防空ごう」等は JWN に概念自体は存在するものの、当該表記がカバーできていなかった

II. synset によって登録されている synonym の表記にばらつきがある
例えば、「吸い込む」と「吸込む」は同じ語の表記バリエーションと考えられるが、synsetID=01539063-v には両者とも登録されているものの、 synsetID= 02765464-v には「吸込む」が登録されていなかった。
01539063-v : {吸い込む, 吸込む,...}
02765464-v : {吸い込む,....}

III. synonym や 語義 (synonym-synset ペア) の数が実質より多い
上記 Ⅱ. の「吸い込む」も「吸込む」も全く別個の語として扱っていたため、見かけ上数が大きくなっていた

そこで、既存の辞書の情報を利用し、本来ある表記の異表記と思われる表記どうしを 1 つの”異表記セット”としてまとめた。その過程で Ⅰ. のように JWN 1.1 でカバーされていなかった表記も取り込み、Ⅱ. のような場合も synset に異表記セットを適用することでばらつきを無くした。また、Ⅲ. についても語義数を”synset-異表記セットのペア”としたことで実態を反映することが可能になった。

尚、JWN のオープンなライセンスを維持するため、ここで利用した辞書もオープンライセンス

なもののみを選択した。具体的には、JUMAN 辞書[3] 、JMdict[4] 、IPA 辞書[5] の 3 つである。

2.2.2. 読み情報を兼ねる仮名表記

各異表記セットには、必ずカタカナ表記とひらがな表記が付与される。これは仮名表記が語として文章中に出現するのみならず、読みを表す文字列として扱われる場合も多いため、その表記の読み情報を参照したい場合に有用なためである。

また、表記が同じでも読みによって意味が変わる場合もあることから、読み情報としての仮名表記を追加することには意義がある。例えば、「面」は「顔」を意味する場合は「メン」、「ツラ」あるいは「オモテ」と読めるが、それ以外はほぼ「メン」としか読まない。

2.2.3. 表示表記

仮名表記の他に、各異表記セットについて”表示表記”を定めた。これは JWN の検索結果表示時や文章生成に表示する表記としていずれか一つに決定する必要があり、また異表記セットの ID に利用したいためであった。現状では表示表記を定める理由は以上の２つ以外には無く、 JWN では ”用いるべきスタンダードな表記”として表示表記を扱う意図は無い。

表示表記は、現状ではその異表記セットにある表記群の中から以下の優先順位で選択される。ただし、今後表記頻度情報をもとに出現頻度の最も高いものを表示表記とする予定である。

1. JUMAN 辞書の表示表記と一致する
2. 漢字の割合が多い
3. 新字体の割合が多い
4. 文字数が
   (a) より多い (元がカタカナ表記)
   (b) より少ない (元がカタカナ以外)

2.2.4. 異表記対応作業の手順

手順 1

JMdict の意味と読みが同一の表記群と、 JUMAN 辞書の代表表記が同一の表記群をマージした異表記セットを作成する。

---

3 http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

4 http://www.edrdg.org/jmdict/j_jmdict.html

5 http://sourceforge.jp/projects/ipadic/

手順 2

JMdict の読み情報をカタカナ・ひらがな表記として各異表記セットに付与する。

ここまでの手順で、前出 II. の「吸い込む」と「吸込む」が一つの異表記セットに組み込まれ、また、「みみずく」の異表記としての「角鴟」と「木兎」のように、 JUMAN 辞書や JMdict から新たな表記が JWN に加えられた。以下にそれらの異表記セットを示す。太字体の部分が異表記セット ID、下線部が本手順で追加された漢字表記である。

ただし、作成した異表記セットのうち、 JWN 1.1 の synonym が含まれていないセットはこの段階で破棄される。

[吸い込む 0（スイコム，吸込む，吸いこむ，すいこむ)]

[みみずく 0（ミミズク，木兎，角鴟，木菟)]

手順 3

手順2 までででカバーしきれなかった JWN 1.1 の synonym については、 辞書として IPA 辞書を用いた MeCab (http://mecab.sourceforge.net/) で形態素分割し、各形態素の IPA 辞書における読みをカタカナ・ひらがなで付与する。「情報機関」という語の場合、「情報/機関」の2形態素に分割され、それぞれの読みを合わせた「ジョウホウキカン」「じょうほうきかん」が付与される。

[情報機関　0 （ジョウホウキカン，じょうほうきかん)]

手順 4

手順3 で読み情報を付与できなかった場合、”読みなし”を示す記号「YOMI」を付与する。

[吸熱　0 （YOMI，YOMI)]

手順4 まで完了した段階で、一度人手によるチェックを行い、全ての異表記セットに読み情報が付与された状態にした。

手順 5

各 synset について、 バージョン 1.1 で付与されていた synonym を含む異表記セットを付与する。2.2.1. 節で例示した synsetID=02765464-v の場合、元々｛呑みこむ，呑込む，吸引，吸い込む，呑み込む，吸収｝が synonym であったため、以下のように異表記セットが付与される。

[吸い込む 0 （ スイコム， 吸込む， 吸いこむ， すいこむ)]
[吸収　　0 （ キュウシュウ， きゅうしゅう)]
[吸引　　0 （ キュウイン， きゅういん)]
[呑みこむ 0 （ノミコム，のみこむ)]
[飲み込む 0 （ノミコム， 飲込む， 呑み込む， 呑込む， のみ込む， のみこむ)]

ただしこの手順では以下のことが問題となる。

I. たとえ元々の synonym が仮名表記であろうと一致する表記が含まれていればどのような異表記セットでも付与するので、例えば 1.1 で 「ホテル」という synonym を持っている synset には「ホテル 0」と「火照る 0」の2つの異表記セットを付与してしまう
II. 同じ表記で読みが複数ある場合、いずれかが不適切な場合があり、例えば元々「面」という synonym が登録されている synset には「メン」と読む異表記セットと「ツラ」と読むものの両方が付与されてしまう

そのため、この手順の後にも人手チェックを行っており、本稿執筆時点ではその作業も 9 割以上完了している。それが全て完了次第、同一 synset に付与された異表記セットで読みの一致するものは一つのセットにまとめる予定で、 synsetID=02765464-v については「飲み込む 0」と「呑みこむ　0」が統合されて図3 のようになる。

2.2.5. 異表記対応の効果

異表記対応後、JWN 全体としては以下のように変化したが、これらの数字には異表記対応とともに行った誤りデータ修正結果を含んでいる。

表 1: 異表記対応による JWN のステータス変化

	JWN 1.1	異表記対応後	
synset 数	57,238		57,125
synonym 異なり数 （語数）	93,834	異表記セット数	83,142
		全て表記の異なり数	212,747
synonym-synset ペア	158,058	異表記セット-synset ペア	148,445

異表記対応結果を synset 単位の変化で見ると、前出の synsetID=02765464-v の synonym は、対応前は 6 つの synonym だったものが、4 つの異表記セットにまとめることが可能で、また「吸込む」もカバーされるようになった。

呑みこむ, 呑込む, 吸引, 吸い込む, 吸収

↓

吸い込む（ スイコム, 吸込む, 吸いこむ, すいこむ ）
吸収 （ キュウシュウ, きゅうしゅう ）
吸引 （ キュウイン, きゅういん ）
飲み込む （ノミコム, 飲込む, 呑み込む, 呑込む, 呑みこむ, のみ込む, のみこむ ）

下線ありは本対応作業で追加された漢字混じり表記

図 3： synsetID=02765464-v における対応結果

また、コーパスの自動カバー率が向上した。 5.1. 節で述べるコーパスについて表 2 に表す。内容語数・カバー済み内容語数には本来ワードネットでは積極的に扱わない固有名詞も含まれているため、 JWN にとって多少厳しい評価方法となっている。

表 2: 各コーパスのカバー率

“カバー済内容語数”と”カバー率”の上段 : JWN 1.1, 下段 : 異表記対応後

	総語数	内容語数	カバー済内容語数	カバー率 (%)
踊る人形	13,483	8,076	4,805	59.4
			6,681	82.7
まだらの紐	13,902	8,389	5,126	61.1
			7,083	84.4
伽藍とバザール	18,067	10,065	8,264	83.2
			8,991	90.5
京大コーパス(記事)	24,615	12,736	9,904	77.8
			10,615	83.3

3. 先行研究

3.1. 機械学習を用いたもの

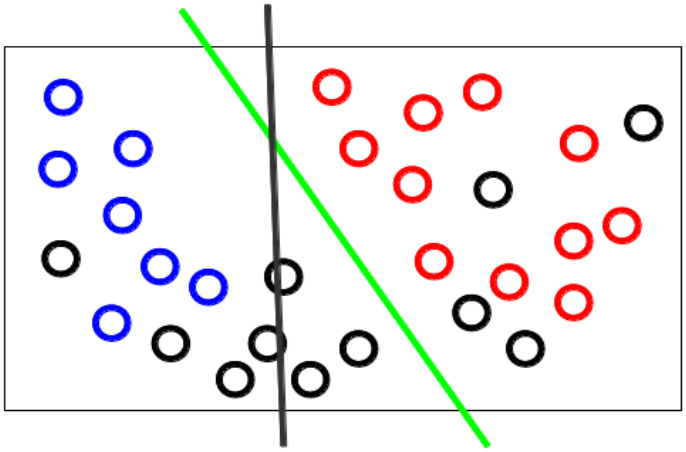
WSD 手法には、コーパスから得たデータを用いて機械学習を行うものが多い。

そのうち教師あり学習を用いた WSD においては、教師データを用いて分類器をトレーニングし、(対象語の出現する)入力文との比較によって対象語の曖昧性解消を行うが、教師データは「ラベルありデータ」とも呼ばれ、これはコーパスアノテーションによって得られる正解(ラベル: どの語義かという情報)と判断材料(素性: 品詞や周辺語句等)から成る。

ただし教師ありデータを数多く得るには多大なコストがかかるため、半教師あり学習が用いられる場合もある。これは「ラベルなしデータ」も教師データに含めてしまう手法である。
これには以下の利点がある。

1. 比較的少数のラベルありデータで学習を行える
2. データの多く分布する部分でラベル境界を引くことが避けられる（図 4 参照）
3. ラベルありデータのない語句でも何らかの結果が出せる

また、ラベルなしデータのみで学習を行う場合は「教師なし学習」と呼ばれる。



青丸： ラベル A, 赤丸： ラベル B, 黒丸： ラベルなし
灰線： 教師あり学習時のラベル AB 境界 , 緑線： 半教師あり学習時の境界
図 4 ： 教師あり学習と半教師あり学習

本節では例として 藤田と藤野（2012）の、半教師あり学習による手法について触れる。

藤田らは、 WSD の技術コンテストである Semeval 2010[6] の日本語タスクで用いられたデータ

6 http://semeval2.fbk.eu/semeval2.php/

セットと、「現代日本語書き言葉均衡コーパス（BCCWJ)」[7]のモニター公開データを利用し、学習と評価を行っている。

Semeval 2010 のデータの内訳は、WSD 対象語が 50 語で、 それぞれラベルありデータと評価用データが 50 文ずつ提供されているが、これらは BCCWJ から人手で作成されている。 辞書は岩波国語辞典（岩波書店，1994）である。

教師データとしては Semeval のラベルありデータ（TRN）と BCCWJ 公開データから自動獲得したラベルありデータ(AUTO)・ラベルなしデータとを用いており、素性は WSD 対象語の出現形、基本形(原形)、品詞、品詞大分類(名詞、動詞、形容詞、...といった粒度のもの)、対象語の前後2語ずつ、前後3語以内の bigrams、 trigrams、 skipbigrms、 また同一文に出現する全内容語の基本形である。第一節の a）に当てはめてみると、素性は a'）となる。

a) ドライバー で ねじ を 緩め た
a') {ドライバー, ドライバー, 名詞-一般, 名詞, で ねじ, ねじ を, で ねじ を, で を, ねじ, 緩める}

実験は WSD 対象語の一部の語義をラベルありデータ（TRN）から削除して擬似未知語義を作り出し、そこに TRN を徐々に戻す、あるいは AUTO を徐々に追加して行き、精度変化を見るという形で行われた。

尚ラベルなしデータも教師なしデータとして扱っているが、これは常に各対象語につき 300 文ずつであった。

評価データによる評価では、途中経過は本研究と直接関係しないため割愛するが、TRN を 30 文戻した状態では精度約 82% 、 AUTO を 30 文追加した状態では精度約 78% であった。また、予め AUTO を追加した状態から TRN を戻して行った場合でも精度が落ちないため、その有効性を示している。教師あり学習では全く対応できない未知語義についても、精度は約 20% であった。

3.2. ワードネット の定義文を用いたもの

ワードネット を用いた WSD としては、まず LESK アルゴリズム（Lesk, M. 1986）によるものが挙げられる。このアルゴリズムは、 文脈（WSD 対象語の出現する文の、対象語以外の内容語）と対象語の各語義の定義文とを比較し、その重複度から語義を推定するものである。

第一節の a ）と「ドライバー」の各語義の定義文の場合、

7 http://www.ninjal.ac.jp/corpus_center/bccwj/

ａ）ドライバー で ねじ を 緩め た
04154565-n： ねじ を 回す 手工 具
10034906-n： 自動車 を 運転 する 人
03244047-n： 打つ 面 が ほぼ 垂直 な ゴルフ クラブ（ ウッド ）で 、ティー・グラウンド から ロング ショット を 打つ の に 使用 さ れる

となり、 a ）と重複する語は synsetID=04154565-n の定義文に出現する「ねじ」のみであることから、 a ）と対応するのは 04154565-n であると予測される。

ただし、この手法の場合は対象文に定義文で用いられている語(表記)そのものが出現していなければならないという弱点がある。 それに対応するため、 Banerjee and Pedersen (2002) は、対象語のみならずそれとリンクしている(例えば上位・下位関係にある）概念の定義文と対象文との比較という形に拡張した。
上記の例で言えば、［04154565-n とその上位・下位概念｛03489162-n、 03489162-n、...｝の定義文]、［10034906-n とその上位下位概念の定義文]、［03244047-n とその上位下位概念の定義文］が a ）と比較される。

またこの段階において、 WSD 対象語の前後の n 個の内容語についても比較対象となる。 a ）の場合は｛「ねじ」とその上位・下位概念, 「緩める」の上位・下位概念｝のそれぞれの定義文も「ドライバー」の曖昧性解消に用いられるため、結果的に「ねじ」「緩める」の曖昧性解消も行われる。

これら２つの対応によって、計算量は増えるものの対象語とワードネットの語義が結び付きやすくなる。

日本語については Baldwin et al. (2010) は、定義文・例文の日本語訳が付与される以前のバージョンである JWN 0.9 の synonym と Hinoki Sensebank (Tanaka et al. 2006) の語義を結び付け、 LESK スタイルの WSD を行っている。

Hinoki Sensebank は NTT コミュニケーション科学基礎研究所によって開発された、 Hinoki Dictionary とそれを用いて行われた語義アノテーションデータ等から成る言語資源である。 Hinoki Dictionary 部分は”親密度”が高いとされる 28,000 の”基本語”の 46,000 の語義（表記と意味のペア）について、基本語と機能語のみを用いた定義文・例文が付与されている。 ワードネットとの構成の違いは、ワードネットが synonym の集合（synset）に対して定義文が付与されているのに対して、 Hinoki では語、ワードネット流に言えば synonym に対して定義文が付与されている点である。 2.1. 節の図２に示した動物の「蝙蝠」を指す synset で例えると、 synset 全体ではなく「蝙蝠」と「蚊食い鳥」双方に定義文が与えられているのと同じである。

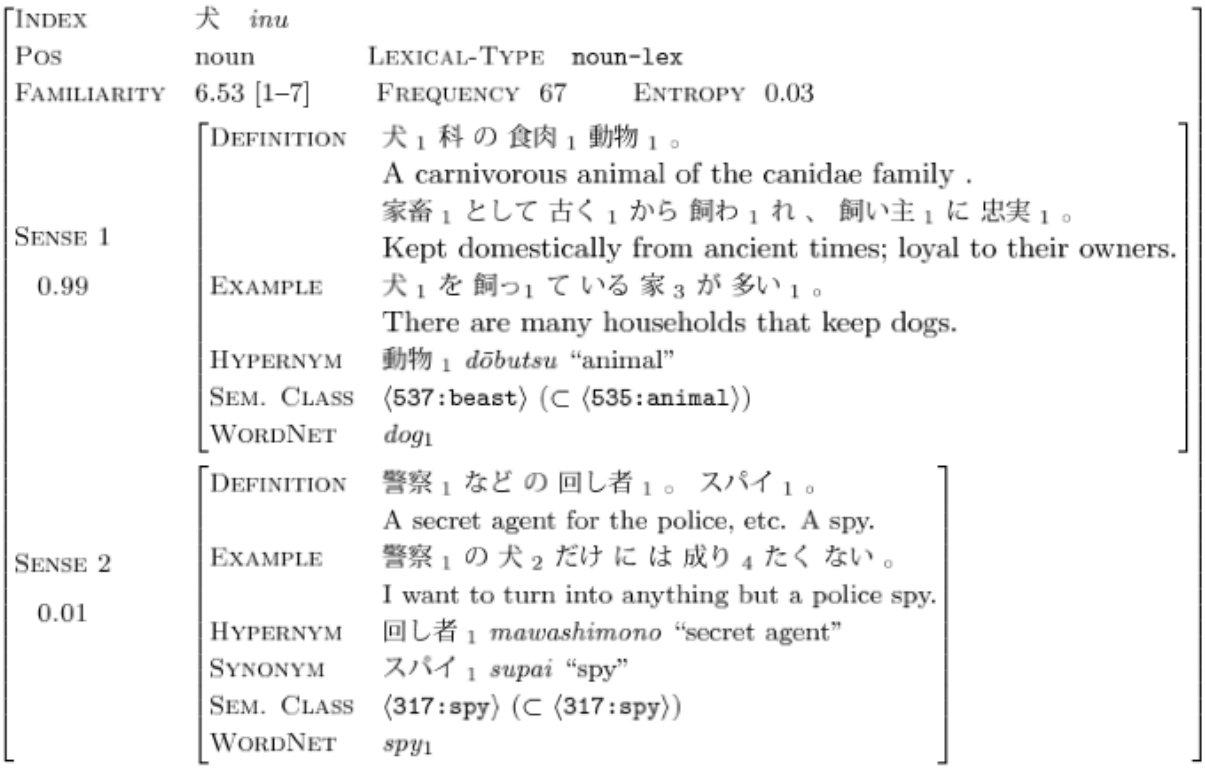


図 5 : Hinoki Sensebank における「犬」の関連情報

図 5 に Baldwin et al. (2010) による例を示す。定義文・例文で ”表記+下付数字”で表されているものは語義情報であり、「犬 1」は「犬」の SENSE 1 であることを意味する。

このうち、「犬」の曖昧性解消に利用される定義文の内容語はそれぞれ以下のとおりである。

「犬 1」 : {犬, 食肉, 動物, 家畜, 古い, 飼う, 飼い主, 忠実}
「犬 2」 : {警察, 回し者, スパイ}

Baldwin et al. (2010) では、 定義文に語義タグが付与されているという Hinoki Sensebank の特性を生かし、定義文の拡張という形で LESK アルゴリズムの変更を行っている。

その一つ目として、対象文と比較される定義文について、そこに出現する全ての内容語の定義文の内容語をそれに含めている。「犬」が曖昧性解消の対象語だとすると、「犬 1」の定義文に対して {「犬 1」,「食肉 1」,「動物 1」,「家畜 1」,「古い 1」,「飼う 1」,「飼い主 1」「忠実 1」} の定義文の内容語を全て加え、同様に「犬 2」の定義文についても {「警察 1」,「回し者 1」,「スパイ 1」} の定義文の内容語を全て加えることになる。ここで「犬 1」の定義文の内容語に対し

「犬1」の定義文の内容語を加えており、この部分が重複することになるが、類似度(対象語がその語義で用いられている可能性を測る指標)計算の際は重複部分も別個のものとして扱う。

二つ目として、対象語の語義の synonym, hypernym, hyponym の定義文に出現する語を定義文に含めている。これは Banerjee & Pedersen (2002) の拡張に近いが、 synonym の定義文も利用している点と、synset 間リンク関係のうち直接の上位・下位関係にあるものの情報のみを用いる点で異なる。これらを総合すると、{犬 1, 犬 1, 食肉 1, 動物 1, 家畜 1, 古い 1, 飼う 1, 飼い主 1, 忠実 1, 動物 1, 犬 1 の下位概念, 犬 1 の内容語の synonym} の各々の定義文の内容語と {犬 2, 警察 1, 回し者 1, スパイ 1, 犬 2 の上位・下位概念, 犬 2 の内容語の synonym} の定義文の内容語と対象文を比較することになる。

最終的には入力文と拡張された各定義文の類似度を求めて最もスコアの高いものに決定される。類似度は以下の DICE 係数 (1) によって算出される。これは 2 つの集合 (A, B) に共通する要素の数を、両集合の要素数の平均で除算したもので、本節においては入力文の内容語の集合が A 、拡張された各定義文の内容語の集合が B に当たる。

$$simDICE(A,B)=\frac{2|A\cap B|}{A\cup B} \quad \text{--- (1)}$$

3.3. グラフ理論を用いたもの

ワードネットの構造はグラフとしてで捉えることができる。即ち synset がノード、リンクがエッジである。

グラフは WSD に利用でき、例えば PageRank をグラフ上で実行することも可能である。

3.3.1. PageRank

PageRank (Brin and Page, 1998) はグラフ理論に基づいて各ウェブページの重要度を測るアルゴリズムで、基本的な考えは、いわば「重要なページは被リンク数が多く、重要なページからリンクしているページも重要度が高いと推測される。従って重要なページからの被リンク数の多いページほどその重要度が高い」というものである。

実際には、全てのページがある同じ得点を持っていると仮定し、それぞれのページはその時保持している得点を出次数（そのページからリンクしているページの数）で除算した得点を各リンク先ページに分配する。これを任意の条件に達するまで反復した段階で各ページの得点を比較し、相対的に重要度を決定する。この条件は、計算が収束するまで、あるいは収束するであろう反復回数に設定されることが多い。ただし、実際のウエブページのハイパーリンク構造においては収束を阻害する要素も存在するが、 Langville and Meyer (2004) にあるように、例えばリンク先

となるウェブページを持たない dangling node であっても他ページへ得点を分配できるようにアルゴリズムを修正する対応が取られている等、収束が保証されている。

PageRank ベクトル **Pr** については Agirre et al.（2009）による以下の方程式(2)が成立する。

$$\mathbf{Pr} = cM\mathbf{Pr} + (1-c)\mathbf{v} \qquad \text{---(2)}$$

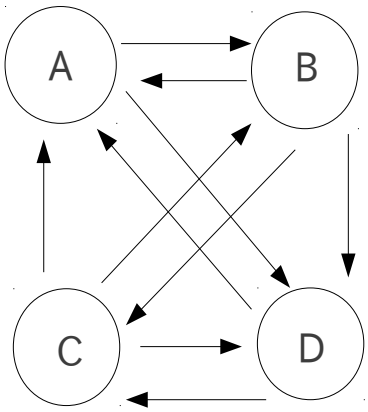


図６：ネットワーク例

$$\begin{pmatrix} 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & 0 \end{pmatrix}$$

図７： 図４の遷移確率行列 *M*

方程式（2）の *M* は N × N（N は全ノード数）の遷移確率行列で、v は 各要素が 1/N の N × 1 の行列であり、c は dumping factor と呼ばれる 0 から 1 の間のスカラで、通常は 0.85 が用いられる。

遷移確率行列は、隣接行列を転置し、非零要素を出次数で除算して各列における要素の総和が「 1」となるようにした行列で、各行について見ると ”他のページからどの程度の確率でそのページに遷移してくるか”に重点が置かれている。

damping factor は、あるページの閲覧者が次にそのページのリンク先ページのいずれかを閲覧する確率とされる。 PageRank は web ページ閲覧者の行動をモデル化したものという考えのもと、閲覧者は必ずしもリンク先ページを参照するとは限らず、全く関係のないページを参照しに行く場合があるためにこの係数が設けられている。

方程式（2）をあるネットワーク（図6）に例に当てはめると、遷移確率行列は 図7 のようになり、dumping factor を仮に「1」と設定すると、PageRank 計算の推移としては、初期状態で各ページが 1 点を保持しているとすると一回目の計算で A=0.833, B=0.833, C=1, D=1.33、二回目の計算で A=1, B=0.747, C=1.087, D=1.163 となる。

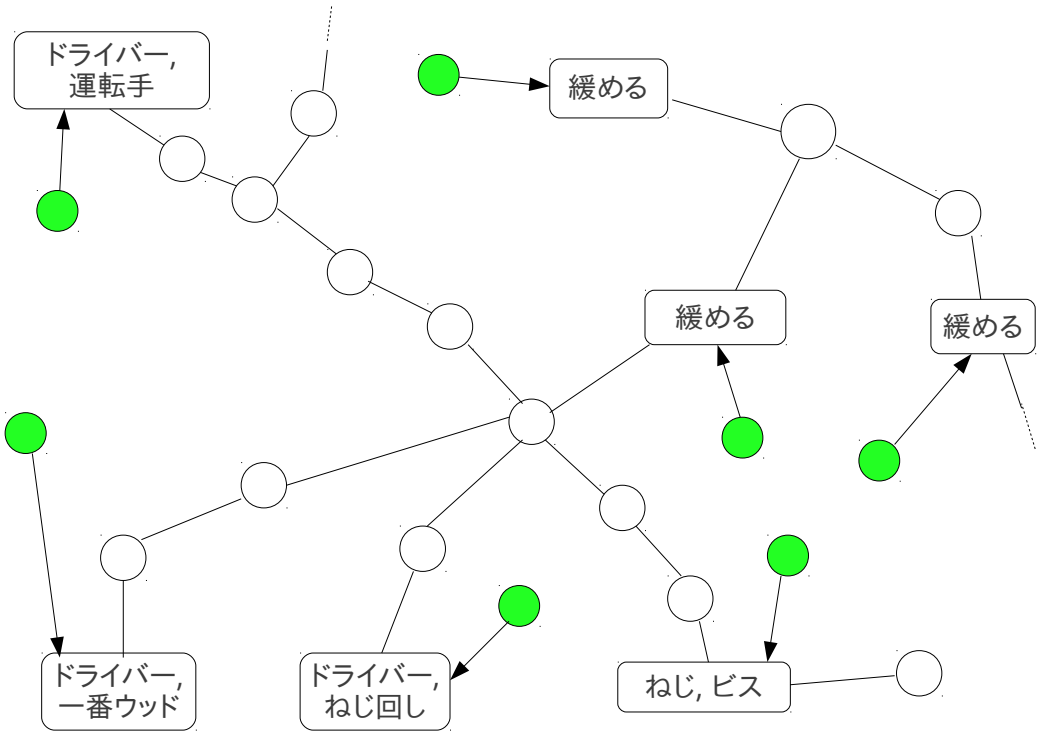
4. 本研究で用いる手法

本節では、実際に本研究で用いる Personalized PageRank と、それを実装したツールである ukb_wsd と本研究で実際に ukb_wsd と入出力を行うファイル形式である KAF について述べる。

4.1. Personalized PageRank

WSD のために PageRank をそのままグラフ上で実行することも可能だが、初期状態で全ノードの得点が同一であるため、どのような入力文であっても結果が変わらない。例えば入力文の内容がどのようなものであったとしても「ドライバー」の語義がある一つの語義で固定されてしまう。

有用な結果を得るためには、入力文ごとに関連する部分グラフ(サブグラフ)のみをまず抽出しなければならず、その分コストが増大してしまう。



丸四角形が対象文の内容語を synonym に持つノード (synset)
無色丸がそれ以外のノード
実線は無向エッジ, 矢印は有向エッジ
緑色丸が内容語を synonym に持つノードに得点を付加するためのノードで、
最初に得点付加を行った後は Personalized PageRank の計算に関与しない

図 8 : Personalized PageRank を適用した JWN ネットワーク

それに替わる手段として、Agirre et al.（2009）は、グラフ全体を用いた WSD が可能になるよう PageRank 自体を改良した。それが Personalized PageRank である。

Personalised PageRank では、WSD 対象文の全ての内容語の属するノード（つまり synset）に向けて、新たなノードを有向エッジでリンクする。こうすると、対象文の内容語の属するノードにのみ外から得点が加わることになり、全てのノードの得点が一様ではなくなる（方程式（2）の v の値が一様でなくなる）ためにグラフ全体を用いることが可能となる。

図 8 に Personalized PageRank を用いた際のグラフ構造を示す。煩雑になるのを避けるためかなり簡略化しており、また、一部は理想的な箇所にエッジを設定した。例えば JWN では「緩める」の登録されている synset について、実際には「ドライバー」や「ねじ」の synset とのリンクはもっと遠回りである。

4.2. ukb_wsd

ukb_wsd は UKB[8] と呼ばれるツールコレクションのうちの 1 つで、グラフ上で Personalized PageRank を実行して WSD を行うツールであり、KB と呼ばれるグラフ内のリンク関係を記したバイナリ(①)、語とその所属する概念を結びつけた辞書(txt ファイル，②)、WSD を行う入力文のセット(txt ファイル，③)を入力として取り、各 WSD 対象語についてそれが属する各 synset のスコア（probability）を出力する。また、リンクの重みを使用する等のオプションも利用できる。

ukb_wsd はグラフ理論で捉えられるものであれば言語を問わず利用できるため研究で用いる。

以降に入力例とファイル①～③の内容例、出力例を示す。

4.2.1. ターミナルにおける入力例

ukb_wsd をターミナル上で実行する場合、必ずオプションや引数が必要となる。「-K」は後続する引数が コンパイルされた KB ファイルであることを表し、「-D」は同じく後続が辞書ファイルであることを示す。入力文ファイルに対してのオプションは不要であるが、引数は KB ファイル 、辞書ファイル、入力文ファイルの順序で並んでいる必要がある。引数「--ppr」 は通常の Personalized PageRank の実行を意味する。

```
% ./ukb_wsd --ppr -K wn17.bin -D wn17_dict.txt context.txt
                    ①            ②          ③
```

4.2.2. KB ファイル

本小節では KB ファイル（前節の ①）のバイナリコンパイル以前のテキストファイルについて述べる。フィールド ”u”はリンク元 synset の ID、フィールド ”v”はリンク先 synset の

8 http://ixa2.si.ehu.es/ukb/

ID、フィールド ”s” は抽出元（”source”）で、本節の例では PWN 1.7 である。フィールド ”d”はリンクの方向で、値が0である場合は無向のリンクであることを示す。

.
u:00002908-n v:10833956-n s:17 d:0
u:00002908-n v:10834114-n s:17 d:0
u:00002908-n v:10834267-n s:17 d:0
u:00003866-n v:06524733-n s:17 d:0
u:00004081-n v:00002908-n s:17 d:0
u:00004081-n v:01088766-n s:17 d:0
.
.
.

4.2.3. 辞書ファイル

4.2.1. の ② に当たる辞書ファイルにおいては、各行の第一フィールドには語が、第二フィールド以降にはその語の属する synset の ID が登録される。区切り文字には半角スペースが用いられているため、英語の複合語等スペースが必要な場合は「_」等の文字に置き換える必要がある。

.
.
accent 05829284-n 11937469-n 05847144-n 05794040-n 05585154-n 00741717-v 00720588-v
accent_mark 05585154-n
accented 02248019-a 02246728-a
accenting 00916300-n
accentor 01154372-n
.
.
.

4.2.4. 入力文ファイル

4.2.1. の ③ に当たる入力文ファイルにおいては、二行ずつ入力文の情報が記述される。一行目には入力文の ID、二行目がその内容語であり WSD 対象語のセットである。全ての語は出現形でなく原形に変換されている必要がある。区切り文字には半角スペースと「#」が用いられる。

以下は「ctx_01」という ID を持った文についての記述内容である。

```
ctx_01
man#n#w1#1 kill#v#w2#1 cat#n#w3#1 hammer#n#w4#1
```

4.2.5. 出力例

4.2.1. のコマンドを入力し、 4.2.2. から 4.2.4. のファイルを用いた場合、以下のような出力を得る。{文 ID, 語 ID, 語義候補とスコア, 対象語} の順に並んでいる。

```
!! ./ukb_wsd --ppr --allranks -K wn17.bin -D wn17_dict.txt context.txt
ctx_01 w1  08249817-n/0.207209 02078052-n/0.0929814 ... !! man
ctx_01 w2  00980806-v/0.114124 00267266-v/0.0808975 ... !! kill
ctx_01 w3  01738104-n/0.24285 02542955-n/0.129899 ... !! cat
ctx_01 w4  02970821-n/0.16454 06073651-n/0.110485 ... !! hammer
```

4.3. KAF

4.2.4. の形式の入力ファイルの場合、軽量ではあるが結局のところ語を原形に直し、フォーマットも合わせなければならない。しかも、元の文章にある機能語や出現形が失われてしまう。そこで本研究では、入力文のファイル形式として KAF（Kyoto Annotation Framework）を用い、ukb_wsd との入出力データの受け渡しには naf_ukb.pl というスクリプトを用いる。

KAF は環境等の地球規模の社会問題について、ワードネットを通じて多言語間で知識共有を実現する目的の Kyoto Project[9] で策定された XML である。これを用いることによって入力文とアノテーション結果や WSD 実行結果を一つのファイルに盛り込むことができる他、そのファイルをそのまま外部に公開できる。

KAF ファイルは 2 つの部分に大別され、 ”<text>” セクションにはコーパスに出現する各文を形態素分割したもの、”<terms>” セクションには形態素解析器によって付与された語の原形や品詞、アノテーション結果といった語に関する情報が記述される。 両セクションの対応付けは ”<text>”セクションの ”<wf wid>” と ”<terms>” セクションの ”<target id>” を同値とすることによる。

9 http://kyoto-project.eu/xmlgroup.iit.cnr.it/kyoto/

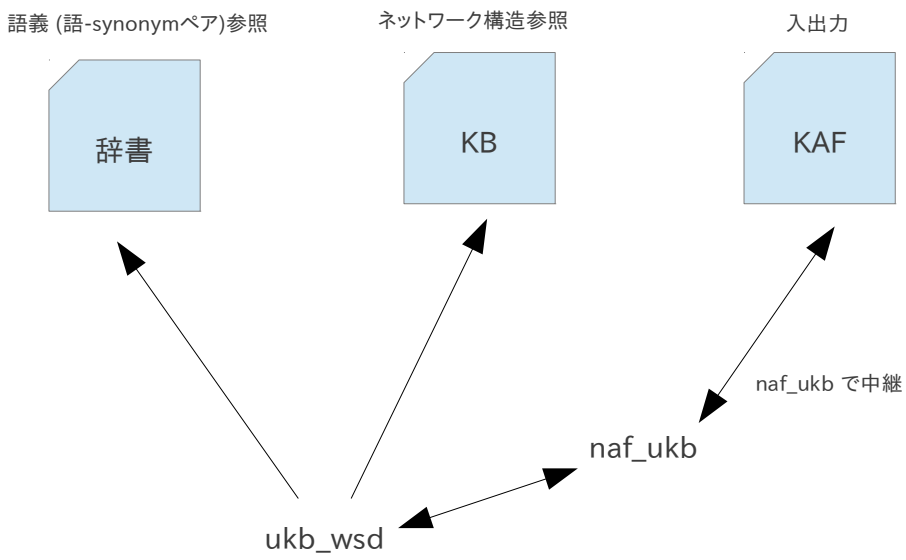


図 9 : ukb_wsd と各ファイルの関係

本研究で用いるコーパスには IPA 辞書の品詞情報が付与されているが、以下のように ukb_wsd が利用できるようワードネットの品詞体系のものに書き換えた。

名詞　→　n
（ただし ”名詞-形容動詞語幹” の場合は ”a”）
動詞　→　v
形容詞、連体詞　→　a
副詞　→　r
上記以外（ただし 上記 4 品詞の ”非自立” のものも含む）　→　x

以下、KAF ファイルの内容例を示す。 5. 節の手動アノテーション結果は”<senseAlt>” フィールドに記述される。

```
<?xml version="1.0" encoding="utf-8"?>
<KAF xml:lang="ja" doc="candb_jp">
```

```
<text>
    <wf wid="w3000:0" sent="3000" para="3000">伽藍</wf>
.
.
    <wf wid="w3009:0" sent="3009" para="3009">本論</wf>
    <wf wid="w3009:1" sent="3009" para="3009">で</wf>
    <wf wid="w3009:2" sent="3009" para="3009">は</wf>
    <wf wid="w3009:3" sent="3009" para="3009">その</wf>
    <wf wid="w3009:4" sent="3009" para="3009">理論</wf>
    <wf wid="w3009:5" sent="3009" para="3009">を</wf>
.
.
</text>
<terms>
.
.
   <term tid="t47" type="open" lemma="本論" pos="n-一般">
       <span>
        <target id="w3009:0"/>
       </span>
    <senseAlt><sense sensecode="06394701-n" confidence="1"/></senseAlt></term>
   <term tid="t48" type="open" lemma="理論" pos="n-一般">
       <span>
        <target id="w3009:4"/>
       </span>
    <senseAlt><sense sensecode="05890249-n" confidence="1"/></senseAlt></term>
.
.
</terms>
</KAF>
```

5. 本研究に用いるデータセット

本研究においては ukb_wsd への入力と結果の検証、また語義等の頻度情報を得るために、日本語コーパスに手動で JWN 1.1 の語義タグが付与された既存のコーパスと、JWN の定義文・例文に 2.2. 節の異表記対応版 JWN の語義タグを自動付与したものを用いる。後者は新たに本研究で構築したものである。

5.1. 手動アノテーションデータ

手動アノテーションは言わば手動で WSD を行ったものであり、ヒューマンエラーは避けられないものの精度は十二分に高いと考えられる。本研究では 2010 年度に NICT により作成された合計 3,173 文の日本語アノテーションデータを利用する。

I.	踊る人形 (シャーロックホームズ)	--- 698 文
II.	まだらの紐 (同上)	--- 702 文
III.	伽藍とバザール (エリックレイモンド)	--- 773 文
IV.	京大コーパス記事部分	--- 1,000 文

このうち、頻度情報は上記全てから算出したが、WSD 結果の検証に用いたものは Ⅲ. のみである。ジャンルとしては Ⅰ. および Ⅱ. は小説、Ⅲ. はエッセー、Ⅳ. は新聞記事であり、全て英語版が存在しており、共同研究先である南洋理工大学の Computational Linguistics Lab[10] において PWN 3.0 の語義タグを付与する手動アノテーションが進んでいる。

アノテーション前のデータは、Ⅰ. および Ⅱ. は青空文庫[11]、 Ⅲ. はプロジェクト杉田玄白[12]のものであり、改変後再配布が可能なため今後 JWN と同様のライセンスで公開予定である。

5.2. 日本語ワードネットの定義文・例文

5.1. 節で挙げたものの他に、頻度情報を得るために JWN の定義文と例文について自動アノテーションを行った。

JWN の定義文・例文は PWN 3.0 にある定義文・例文すべてを日本語化したものであるため、PWN 3.0 の定義文・例文に手動アノテーションを施した Princeton WordNet Gloss Corpus[13] とは文単位で対応が取れており、当該コーパスの語に synsetID という形で付与されている語義タグと、

10 http://compling.hss.ntu.edu.sg/index.html
11 http://www.aozora.gr.jp/
12http://www.genpaku.org/
13 http://wordnet.princeton.edu/glosstag.shtml

本節で日本語文に付与する語義タグとのマッチングという形を取ることが可能である。

図 10 は synsetID=14672373-n の二文目の定義文における、マッチング結果の例である。英文に付与されている synsetID により、日本語文側の「チップ」 に付与されるべき synsetID は 03020034-n であることが判明する。

synsetID=14672373-n
synonym : コルタン
definition
JPN: 携帯電話とコンピュータチップで使われる;
ENG: used in cell phones and computer chips;

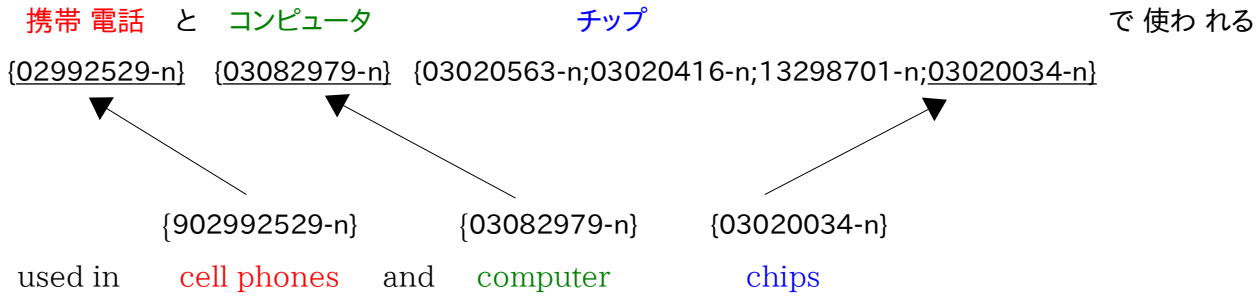


図 10 ： 日英マッチング例

5.2.1. 前処理

前処理として IPA 辞書を搭載した MeCab により、日本語文には形態素解析を行い、JWN の synonym については分かち書きを行った。

尚、本稿において”単語”は MeCab が”一形態素から成り立つ”と判断した語、例えば上記の「コンピュータ」や「チップ」、”複合語”は”複数形態素から成り立つ”と判断した語、同じく「携帯電話」のような語を指す。

MeCab に用いた IPA 辞書については、本マッチング作業の特性に合わせて手を加えた。

I. 「一種」

JWN の定義文には 「一種」 という表現が比較的良く出現するが、これは英語の ”a kind”, ”a sort” に該当する。

PWN の Gloss Tag Corpus においては ”a kind” や ”a sort” ではなく “kind” や “sort” に対して synset ID が付与されている。 そこで辞書ファイルから 「一種」 を削除して「一/種」と分かち書きされるようにしたところ、「種」と “kind” または

“sort”が synset ID=05839024-n でマッチするようになった。

II.”動詞連用形+「方」”

「考え方」のような ”動詞連用形+「方」” の形を取るものも一語扱いで IPA 辞書に登録されている。これらは、英語では多く ”way(s) of ~” あるいは ”manner(s) of ~” と対応するが、 Gloss Corpus では ”way” や ”manner” の単位で synsetID が付与されており、このままでは日英マッチングが不可能であった。そのため、「方」と ”way” や ”manner” をマッチングさせるために、”動詞連用形+「方」” の形を取る 41 語について IPA 辞書から削除した。ただし、削除したのは品詞が ”名詞-一般” となっているもののみで、「明け方」のような時間表現に用いるもの（品詞は全て ”名詞-副詞可能”） に関しては、同じ構成ではあるが悪影響はないので手をつけていない。

5.2.2. 単語の日英マッチング

MeCab による処理以降は単語と複合語で別の並行フェーズとして英語とのマッチングを行ったが、まず単語におけるマッチング手順について述べる。

手順 1　　品詞による機能語除外

日本語文について、ワードネットにおいては機能語は扱わないため、 MeCab の出力した品詞情報をもとに助詞等の機能語はマッチング対象から除外した。 IPA 品詞の ”名詞-形容動詞語幹”や ”連体詞” がワードネットの adjective に含まれる等、品詞体系に差があるため、ここでは機能語の除外以外に品詞情報は利用していない。この手順を行って残ったものが単語における内容語である。

手順 2　　各内容語に JWN の synset ID を付与

次に、各内容語について、 MeCab の出力した語の原形（lemma）と一致する synonym が属する全ての synset の ID をマッチング候補として付与した。また日本語文の中に英単語が含まれる場合があり、 それらにもマッチング候補を付与するため、PWN 3.0 に存在する synonym も用いた。これによって、前出の synsetID=14672373-n の定義文の「コンピュータ」と「チップ」に対して {03082979-n} と {03020563-n, 03020416-n, 13298701-n, 03020034-n} がそれぞれ付与される。

ただし、この手順において以下の語については候補を付与しなかった。

I. 「よう」

「~のようだ」 に類する表現がほとんどであり、単体ではマッチすべき synset が JWN にないため。

II.「て」に後続する「いる」
ほぼ「~している」という表現であり、この場合の「いる」は aspect 情報を付与するための機能語と考えられるため。

III. 品詞が「名詞-サ変接続」である語に後続する「する」
「勉強/する」のような場合がこれに当たる。
JWN においては、この「する」も動詞であることを示す機能語と捉えているため。

手順 3　　対応する英文に付与された synset ID とのマッチング

Gloss Corpus 内の対応する英文に付与された synset ID に、 日本語文のマッチング候補と一致するものがあるかどうか調べる。このとき英文側の単語とその出現位置(語順)については考慮していない。

前出の例では、日本語側の各内容語に付与されている synsetID の中に、 {902992529-n, 03082979-n, 03020034-n} のいずれかと一致するものがあるかを一つ一つ調べて行く。

5.2.3. Exact match できないもの

この手順において、 exact match、即ち前出の synsetID=14672373-n の場合のように日本語文の内容語に付与されたマッチング候補の中に、英文に付与された synset ID と完全に一致するものがある場合だけでは充分とは言えなかった。例えば以下の synsetID=04259630-n のような場合である。

synsetID=04259630-n
synonym: ソンブレロ
definition
JPN: アメリカ 南西 部 と メキシコ で 着用 さ れる
ENG: worn in American southwest and in Mexico

図 11:　exact match しない例（赤線部）

図 11 の「アメリカ」の候補は 09044862-n という名詞 synset 、一方「American」に付与されている のは synsetID=02927512-a という形容詞 synset である。このような品詞の差異を吸収するために、 PWN にある pertainym と derivational form という関係性を利用し、英文側のマッチング候補として加えた。尚、これら 2 つの関係性は synset 間でなく synonym 間のリンク関係であるため、 今のところ JWN では採用していない。

I. pertainym

これは ”形容詞 synonym → 関連する名詞 synonym” あるいは ”副詞 synonym → 関連する形容詞 synonym” の一方向のリンクであり、図 11 の例における “「synsetID=02927512-a の American」 → 「synsetID=09044862-n の America」” がこれに当たる。このため、 synsetID=02927512-a の「American」は {exact:02927512-a, pertainym:09044862-n} という情報を持つことになる。

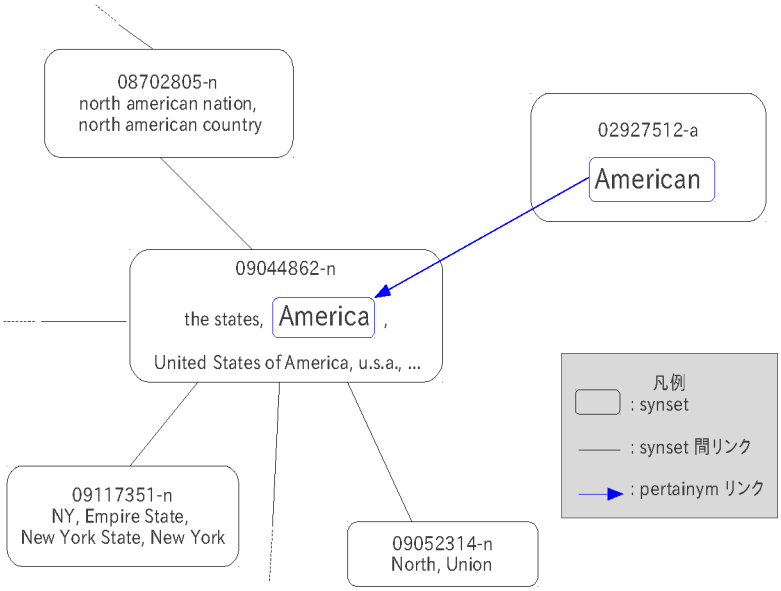


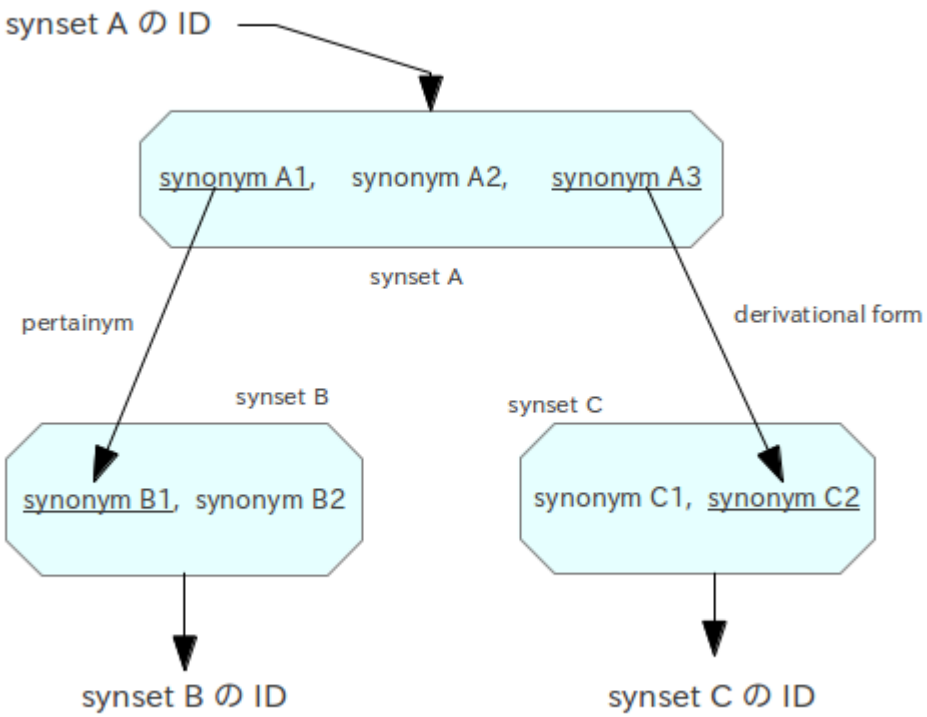
図 12 : pertainym リンク

II. derivational form

”動詞 synonym → 関連する synonym” 等、 I. 以外の関係性を持つ synonym へのリンクである。

実際に pertainym と derivational form を導き出す流れとしては、図 13 のように” 英文側の語に付与されている synset ID → その synset 全ての synonym → それらの pertainym or derivational form → それらの属する synset ID “ という手順を踏んでいるが、これは ”同一 synset に属する synonym は等価である” というワードネットにおける前提を利用したためである。

また、exact match > pertainym > derivational form の順に信頼性が低下するため、 exact match できなかった場合に pertainym でマッチするか調べ、それでもマッチしなかった場合のみ derivational form の情報を利用することとした。



synset A の ID から {pertainym:synset B, derivational form:synset C} という情報を得る

図 13 : pertainym と derivational form の情報からの synsetID 獲得

5.2.4. 複合語の日英マッチング

複合語については、まずその検出から始め、それ以降は単語の手順 2 以降と同じ手順を踏む。単語とは独立に行うため、結果どうしの関連性はない。

複合語検出

単語の場合は単に MeCab の出力にある原形と synonym の一致を調べれば良いが、用言の複合語の場合は第二形態素以降に活用の有無が存在する。そのため以下のように検出を行った。

まず図 14 のように、まず 手元版 JWN の synonym を MeCab で分かち書きした後の第一形態素目をインデックス とした辞書作成し、値には第二形態素目、第三形態素目、... と順に格納する。例えばの「つ

なぎ/合わせる」は「つなぎ」 (用言でもこの部分は変化しない)でインデックスされる。

次に、入力文の i 番目の形態素が辞書のインデックスと一致した場に i+1 番目の形態素の出現形/原形と 辞書の値の先頭 (つまり複合語の第二形態素目)の文字列を比較し、いずれかと一致すれば以降も入力文 i+2 番目の形態素の出現形・原形と辞書の値の 2 番目の文字列...のように同様の比較を繰り返す。

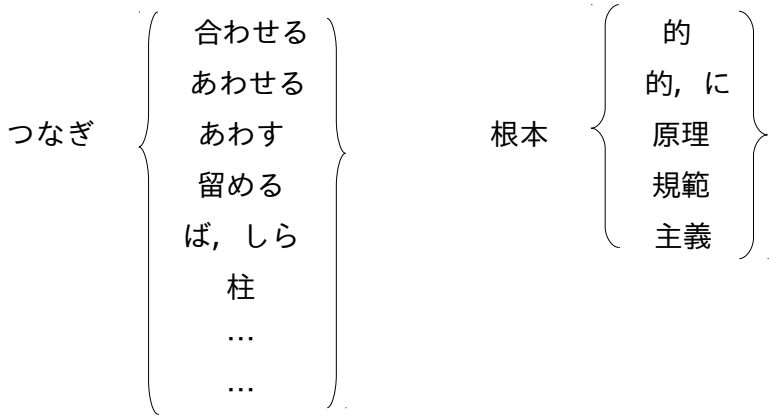


図 14： 複合語辞書

synsetID=01295275-v　例文

Jpn:　組み 合わさる よう これら 2 つ の 部分 を つなぎ 合わせ て ください

Eng:　join these two parts so that they fit together

出現形 ： . . . . . の 部分 を つなぎ ① 合わせ て ください

原形 ： . . . . . の 部分 を つなぎ 合わせる ② て くださる

図 15： 複合語の検出

図 15 の synsetID=01295275-v の例文の場合、以下のように 9 番目の形態素の出現形が辞書のインデックス「つなぎ」と一致し (①) 、辞書の値の先頭である「合わせる」が文の 10 番目の形態素の原形と一致する (②) ため、文の 9 番目と 10 番目の形態素を併せて「つなぎ/合わせる」という複合語であると判断する。

5.2.5. マッチング結果

本節の日英マッチングによる自動アノテーションで得られた語義タグ（synsetID）は表3 のとおりである。

表 3 : 日英マッチングの結果

定義文･例文の (複) の内容語数は複合語と検出された数を示す。

	内容語数	exact	pertainym	derivational
定義文 (単)	940,002	222,673	6,612	28,854
定義文 (複)	94,480	35,439	1,043	2,058
例文 (単)	227,681	11,562	1,103	3,057
例文 (複)	22,688	2,770	225	304

Gloss Corpus に付与されていた synsetID の延べ数は定義文448,280・例文 47,327 であり、実際にマッチングできた数と開きがあるが、次のような原因が考えられる。そのうち主だったものは I.1. と Ⅱ. である。

I. 日本語文側に対応する語が出現しているが、 JWN に synonym として登録されていない

1. JWN の synonym は、まず複数の言語から対日辞書を用いて半自動で作成したが、あまり一般的でない語はそれらの辞書に登録されていない場合があり、この段階では synonym に成り得ない。その後人手チェックが行われているが、既存の synset-synonym のペアが妥当なものであるかのチェックが中心であり、元々 JWN にない語の追加までは充分手が回っていない
2. 異表記対応が充分ではない部分で、長音符（「ー」, ”伸ばし棒”とも呼ばれる） を母音に置き換える対応は行っていない

II. 日本語訳の問題

明らかな誤訳と思われる場合もあるが、翻訳の難しい語句であってもとにかく日本語化したために不適切な訳になっている場合もある

III. 形態素に関わる問題

1. 「党員」と ”member of the party”のように日本語（IPA 辞書）では一語となっているが、対応する英語表現が複数の語になっており、英語側ではその一部、この例では ”member”と ”party”の単位で synsetID が付与されている
2. MeCab の分かち書きのエラー

6. 実験と結果

本実験では日本語コーパス「伽藍とバザール」の各文について ukb_wsd を用いて行った。

まず ukb_wsd と受け渡しを行った KB ファイル、辞書ファイル、 KAF ファイルのデータについて述べる。

KB ファイル

ネットワークの構造を ukb_wsd に渡す KB ファイルには、PWN 3.0 の概念間リンクと、 PWN の定義文から抽出した概念間リンクをマージしたものを用いた。

前者については JWN の概念間リンクと同一である。後者は synset の ID と、英語定義文に出現する synonym の synsetID を結び付けたものである。 2.1. 節の図 2 の synsetID=02139199-n の場合、定義文の中で判明している語義は{wing:02151625-n, form:02621395-v,...} であるため、それぞれの synsetID と 02139199-n のペアが KB ファイルに登録される。

辞書ファイルと KAF

辞書ファイルは手動アノテーション時のバージョンである JWN 1.1 (jp11) を用い、KAF ファイルは「伽藍とバザール」の日本語手動アノテーション結果から生成したもの (CBJ) を用いた。CBJ 中の評価対象語数は 6,617 である。

MFS と MFC

また、 WSD のベースラインを得るため、正解率向上に用いるために語義頻度情報を得た。具体的には、コーパス毎に語義（synset-synonym ペア）頻度を求め、各語について最も頻度の高い語義（Most Frequent Sense, MFS）一つに絞り込むものである。例えば「伽藍とバザール」には「経路」という語が 16 回出現するが、そのうち 14 回が synsetID=08616311-n の synonym として、 2 回が synsetID=06260121-n の synonym としてである。このとき、「経路」に対応する synsetID を 06260121-n のみとする。本研究では MFS を用いる際は出現回数が 5 回以上の語のもの（MFS5）を採用した。

もう一つの絞り込み手法として MFC（Most Frequent Concept）の辞書への適用も行った。これはコーパスごとに付与された概念のうち頻度の高いものであり、本研究では 5 回以上出現したもの（MFC5）とした。

MFC5 では当該 synset のどの synonym としてコーパスに出現したかは考慮されない。例えば「伽藍とバザール」には synsetID=05142863-n が 6 回出現するため、その synonym である「利益」等がコーパスに出現した場合に、それらに対応する synsetID は 05142863-n のみに絞られる。ただし、 synsetID=01410606-a は 12 回出現し、その synonym の一つである「同じ」に対

応づけられるものは本来は 01410606-a のみとなるものの、 synsetID=02062670-a も synonym の一つが「同じ」であり、 6 回出現するために両者ともが対応する synsetID となる。

表 4 は各コーパスから得た MFS5 と MFC5 の数字であるが、「cb」が「伽藍とバザール」、「nv」が小説（「踊る人形」と「まだらの紐」を統合)、「ky」が京大コーパス、「df」が JWN の定義文、「ex」が例文を意味する。

表 4 : MFS5 と MFC5 の獲得数

	cb	nv	ky	df	ex
MFS5	1,070	352	410	8,223	822
MFC5	377	364	393	9,285	314

6.1. ukb_wsd 第一実験

まず最初に CBJ を入力として ukb_wsd を実行したところ、正解率は 0.2881 で、ランダムで語義を選んだ場合の正解率 0.3377 よりも下回る結果となった。

6.2. 正解率向上に向けた変更

英語版アノテーションデータの適用

そこで、英語版「伽藍とバザール」の手動アノテーション結果を日本語版に対応させたものとのマージを行い、それを用いて再び ukb_wsd を実行したところ、正解率は 0.4304 に向上した。マージ後のデータ (CBC) は具体的には、日本語版コーパスアノテーションで synsetID が付与されなかった語に対し、英語版の対応データから synsetID を付与したものである。

例えば「ベータテスタ」は「伽藍とバザール」で 6 回出現するが、JWN 1.1 に存在しない語であったために WSD 対象語にならず、当然 CBJ にも記述していない。しかし英語版アノテーションデータからの対応データでは synsetID=07944618-n という語義タグが付与されている。また、JWN 1.1 に存在していても何らかの理由で日本語版アノテーションで語義タグが付与できなかったものの中にも、英語版からのデータでは付与されている場合もあった。これらのようなデータを CBJ に追加して行き、CBC が作成された。

辞書ファイルについては、英語版の手動アノテーション時には南洋理工大学で独自に JWN 1.1 に 420 語義を追加したもの (jp11e) を利用していたため、CBC を入力とする際はそれを用いた。

ただし、英語版からの対応データの中にエラーデータが見られたため、その部分の修正を行った。具体的には、品詞が "名詞-サ変接続"の語の直後の「する」または「できる」にも synsetID が付与されていたものであるが、これは JWN では機能語的に扱うために WSD 対象にならず、また実際にそれらに付与されてい

た synsetID も誤ったものであったため、全て KAF ファイルから除外している。これによって CBC の WSD 評価対象語数は 7,070 となった。

品詞情報の不利用

しかし、ukb_wsd は KAF ファイルに記述されている対象語の品詞のみに語義候補を絞ってしまうために、手動アノテーション結果と WSD 結果が食い違ってしまうことがあった。例えば、「フリー」は「伽藍とバザール」においては形容詞の意味で用いられているが、ukb_wsd は KAF ファイルに記述されている "n" つまり名詞を品詞とする語義のみを返してくるため、このままでは一致することがない。そこで KAF ファイルにある品詞情報に拘らない、つまり品詞情報を参照しない形に ukb_wsd を改良したところ、CBC を用いた際の正解率が 0.4322 に向上した。一見さほど効果がないように見えるが、ukb_wsd が何らかの結果を返した語の数が 6,129 から 7,012 に増加した上での正解率であるため、効果は充分と考える。

最頻出語義の利用

次に、辞書ファイルである jp11e に MFS5 を適用して事前に語義候補を絞り込んだ上で、CBC を入力として ukb_wsd を実行した。基本的にどのコーパスから得た MFS5 を適用しても、適用しない場合より正解率が上がるが、特に小説から得た MFS5 を適用した場合が最も高く 0.4514 であった。これは「伽藍とバザール」はエッセーであり、小説とはドメインが近いことが要因として考えられる。

また、MFS5 との比較として MFC5 の適用も試みた。その際の正解率については付録に記すが、多くの場合は MFS5 を適用した場合と両者とも適用していない場合の中間であった。このことから、ある適切な方針をもって語義の絞り込みを掛けることは有効であり、特に MFS の効果が高いと言える。

6.3. ベースライン

本実験の比較対象として、ukb_wsd を用いずにランダムあるいは MFS を用いた場合について述べる。

まずランダム選択の正解率は、CBJ の場合は 0.3377 (対象語数は 6,617) で、CBC の場合が 0.3679 (対象語数は 7,070) であった。

次に CBC に対して MFS5 単体で語義選択をした場合の結果と、MFS5 を用いつつそれを適用できない語はランダムで語義を選択した場合の結果は、表 5 のようになった。

表 5 : MFS5 による語義選択

	cb		nv		ky		df		ex	
	単体	+random	単体	+random	単体	+random	単体	+random	単体	+random
対象語数	3,863	7,070	1,169	7,070	1,268	7,070	3,993	7,070	1,837	7,070
正解率	0.8237	0.6010	0.4311	0.3970	0.4117	0.4193	0.3108	0.3972	0.2319	0.3847

6.4. ukb_wsd 実験最終結果

本節では、日本語データのみ (CBJ) → 英語版からの対応情報 (CBC) → 品詞情報不利用 → MFS の適用の順で行ってきたが、品詞情報と MFS については個別に適用したデータも作成したため、以下の表 6 に示す。

表 6 : ukb_wsd 実行結果

青字・赤字は正解率上位 3 つ

	正解	不正解	missing	正解率(全体)	正解率 (w/o “missing”)
ukb 日	1,262	3,228	2,177	0.1907	0.2811
ukb 英	2,638	3,491	941	0.3731	0.4304
ukb 英 + 全品詞	3,031	3,981	58	0.4287	0.4322
ukb 英 + cbmfs	3,924	1,860	1,286	0.5550	0.6784
ukb 英 + cbmfs + 全品詞	4,700	2,283	87	0.6648	0.6731
ukb 英 + nvmfs	2,738	3,362	970	0.3873	0.4489
ukb 英 + nvmfs + 全品詞	3,165	3,847	58	0.4477	0.4514
ukb 英 + kymfs	2,611	3,358	1,101	0.3693	0.4374
ukb 英 + kymfs + 全品詞	3,136	3,876	58	0.4436	0.4472
ukb 英 + dfmfs	2,467	3,037	1,566	0.3489	0.4482
ukb 英 + dfmfs + 全品詞	2,992	4,020	58	0.4232	0.4267
ukb 英 +exfmfs	2,538	3,185	1,347	0.3590	0.4435
ukb 英 + exmfs + 全品詞	3,023	3,976	71	0.4276	0.4319

表の「ukb 日」は ukb_wsd で CBJ を入力として WSD を行った場合、「ukb 英」は CBC を入力とした場合を示し、「全品詞」は品詞情報を利用しない場合を示す。「mfs」は MFS5 を適用したことを示し、「mfs」の直前にある「cb」は「伽藍とバザール」、「nv」は小説、「ky」は京大コーパス、「df」は JWN 定義文、「ex」は JWN 例文から得た MFS5 である。

「cbmfs」については、「伽藍とバザール」から得た MFS5 を「伽藍とバザール」に適用するものであり、正解率が高いのは当然であるため、あくまで参考としての情報である。

「missing」は ukb_wsd が結果を返さないことを表し、以下の場合に発生する。

I. 辞書ファイル (jp11 または jp11e) に語はあるものの KAF ファイルにある品詞情報と食い違う

II. 全ての語義候補の得点が閾値 (本研究では 0.01) 以下となる

これらのうち基本的に Ⅰ. が要因となることが多いことから、品詞情報の利用/不利用での違いを示すため、表に含める。

正解率は ”missing” を計算に入れたものと入れてないものの 2 種類について示す。本稿で ”正解率” と言う場合は後者を指している。このとき、各試行における正解率の母数となる対象語数は異なってしまうため、”mising” の数の差と正解率の増減の双方を勘案する必要がある。

また、語の持つ曖昧性の数と正解率の関係を、表 6 の ”ukb 英 + 全品詞” と ”ukb 英 + nvmfs + 全品詞” のデータにおいて調査したところ、双方とも語義を 3 つ持つ語の正解率が 45% 程度、語義を 2 つ持つものの場合は 50% 程度であった。また、語義が 4 つ以上になると正解率が 50% を越えることがなく、語の曖昧性が増えるとともに正解率が低下する傾向にあった。

7. 考察

ukb_wsd を用いた WSD において、 CBJ を入力とした場合は正解率が 30% を下回り、 CBC を入力とすると 40% を越えた。 辞書ファイル jp11 と jp11e の間で語義が追加されているのもその要因だが、一文に含まれる内容語数の差も影響している可能性がある。Agirre et al.（2009）では内容語数の影響について直接的な言及は避けているが、一文に含まれる内容語数が3 語以内のものは前後の文と統合し、なるべく 20 語程度となるような条件で実験を行っている。CBJ は 8.6 語/文 で、CBC は 9.1 語/文となっている。

次に、MFS は小説のものを適用した場合が最も正解率が高くなっているが、これは「伽藍とバザール」がエッセーであり、小説とドメインが近いためと考えられる。次点では京大コーパスの MFS で、正解率の差はさほど大きくはないが、小説の MFS5 が352 語義で京大コーパスの MFS5 の 410 語義よりも少ないため、 やはり前者の方が効果的と考えられる。

また、同じ条件で ”missing” の数のみ差がある場合（つまり表 6 において ”全品詞”となっているか否か）において、小説や京大コーパスの MFS5 を適用した場合は正解率が向上しているが、それ以外でも ”missing” の数の増加に比べて正解率の低下は小さい。”missing” は主に品詞体系の違い、その多くは日英の言語間の品詞体系の違いから来るものであり、品詞体系に違いがある場合は品詞情報をヒントとして参照しない方が良いと言える。

本研究において一番高いケースの正解率は約 45% である。Agirre et al.（2009）は英語で Senseval 2 と 3 のデータセットを用いて実験し、正解率がほぼ 50~60% であったため、それよりも低いことになる。原因として本節で述べた一文あたりの内容語数も関係している可能性があるが、 JWN は PWN 3.0 の概念間ネットワーク構造をそのまま用いており、それが日本語と合っていないという可能性もある。

8. まとめと今後の課題

本研究では、まず JWN の異表記対応を行って表記のカバー数を上げ、それを用いて定義文・例文の自動アノテーションを行った。次に既存の手動アノテーション済みエッセーコーパスを入力に WSD を行った。日本語版のみのデータでは充分な正解率とは言えなかったため、英語版のアノテーションデータで補強し、
ツールの修正を行い、各コーパスから得た語義頻度情報で語義を絞り込んだところ、正解率は最大で 0.4514 となった。

今後の課題として、まず異表記対応については、伸ばし棒を母音に変換する必要性が自動アノテーション時に判明した。またカバーできる表記数が増えたということは曖昧性が増したということでもあるため、本研究で利用したコーパス以外からも語義と表記の頻度情報を得て、よく出現するもの/しないものの区別が必要である。

次に JWN 定義文・例文の自動アノテーションについて、日英でマッチングできなかったものは、今まで JWN に存在しなかった語義の追加や、定義文・例文の日本語訳の修正の際の有意義な手掛かりとなる。また、マッチング数を増やすためには上位・下位などの synset どうしの関係性を利用したり、英文側の語にも新たに候補となる synsetID を付与して、日本語側との一致を調べる、といったことも考えられる。

最後に ukb_wsd を用いた WSD については、「伽藍とバザール」以外の MFS を適用してみる、MFS の条件を単純に出現回数を閾値としたものから変更してみる、JWN 定義文のアノテーションから得た情報から KB ファイルを変更してみる、等で正解率の変化を調べる必要がある。

付録

表 7 : MFC5 による語義選択

凡例は表 5 と同様

	cb		nv		ky		df		ex	
	単体	+random	単体	+random	単体	+random	単体	+random	単体	+random
対象語数	4,541	7,070	1,936	7,070	2,292	7,070	5,603	7,070	1,548	7,070
正解率	0.6720	0.6019	0.3300	0.3974	0.3551	0.4167	0.3081	0.3951	0.2657	0.3908

表 8 : MFC5 を適用した場合の ukb_wsd の実行結果

凡例は表 6 と同様

	正解	不正解	missing	正解率(全体)	正解率 (w/o “missing”)
ukb 英 + cbmfc +全品詞	4,213	2,798	59	0.5959	0.6009
ukb 英 + nvmfc +全品詞	3,124	3,866	80	0.4419	0.4469
ukb 英 + kymfc +全品詞	3,153	3,859	58	0.4460	0.4497
ukb 英 + dfmfc +全品詞	2,873	4,139	58	0.4063	0.4097
ukb 英 + exmfc +全品詞	2,994	3,990	86	0.4234	0.4287

参考文献

[1] Eneko Agirre and Aitor Soroa. Personalizing PageRank for Word Sense Disambiguation. Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009). 2009.
[2] S. Banerjee and T. Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February, 2002.
[3] Timothy Baldwin, Su Nam Kim, Francis Bond, Sanae Fujita, David Martinez, Takaaki Tanaka. A Reexamination of MRD-Based Word Sense Disambiguation. ACM. 2010.
[4] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. 1998.
[5] Christiane Fellbaum (Eds.). WordNet: an electronic lexical database. MIT Press. 1998.
[6] Sanae Fujita and Akinori Fujino. Word Sense Disambiguation by Combining Labeled Data Expansion and Semi-Supervised Learning Method. IJCNLP-2011, pp. 676-685. 2013.
[7] 藤田早苗, 藤野昭典. 少数のラベルありデータからの語義曖昧性解消. 言語処理学会第 18 回年次大会(NLP2012). 2012.
[8] Hitoshi Isahara, Francis Bond, Kiyotaka Uchimoto, Masao Utiyama and Kyoko Kanzaki. Development of Japanese WordNet. In LREC-2008, Marrakech, 2008.
[9] 栗林孝行 , Francis Bond, 黒田航, 神崎享子, 内元清貴, 井佐原均. 日本語ワードネットにおける異表記拡張の効果. 言語処理学会第 18 回年次大会(NLP2012). 2012.
[10] Amy N. Langville and Carl D. Meyer. Deeper inside pagerank. Internet Mathmatics, 1(3): 335-380. 2004.
[11] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, pages 24-26, New York, NY, USA. ACM. 1986
[12] Takaaki Tanaka, Francis Bond and Sanae Fujita. The Hinoki sensebank – a large-scale word sense tagged corpus of Japanese -. Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006. Australia.

謝辞

本稿執筆にあたり、肌理細かな御指導と御鞭撻を頂きました本学情報メディア基盤センター 井佐原均教授に心より感謝申し上げます。

本稿を御査読頂き、多くの御指摘と御助言を頂きました本学大学院工学研究科 情報・知能工学系 中川聖一教授、同 秋葉友良准教授に深謝申し上げます。

本研究において井佐原教授とともに御指導頂きました 南洋理工大学 Division of Linguistics and Multilingual Studies, School of Humanities and Social Sciences ランシス・ボンド准教授に心より感謝申し上げます。

本研究を進めるにあたり御助言と激励を頂きました本学 情報メディア基盤センター 研究員 神崎享子様に心より感謝申し上げます。

本研究において種々のサポートや激励を頂きました本学 情報・知能工学系 言語情報学研究室の学生各位に心より感謝申し上げます。

折に触れて本研究についての御相談を受けて頂きました本学大学院工学研究科 情報・知能工学専攻 博士前期課程修了生 大塚道長様、同 山本健太郎様に心より感謝申し上げます。

最後に私の我侭を聞き入れて快く奈良から本学に送り出してくれた家族に心より感謝致します。