2023年度（令和5年度）　修士論文

Word2Vecと対訳単語対を利用した対義語・類義語の自動抽出

令和6年2月13日

鳥取大学大学院　持続性社会創生科学研究科
工学専攻　情報エレクトロニクスコース

自然言語処理研究室

M22J4052M 柳原 弘哉

概要

対義語と類義語の分類・抽出は自然言語処理分野において非常に重要なタスクである．言葉の示す対義性・類似性等を原理的に解明することで言語理解・言語処理の向上に貢献する．しかし，類義語に関する研究と比較して，対義語に関する研究は極端に少ない．その理由として，対義の関係にある単語を抽出するタスクが非常に困難な点が挙げられる．そのため，対義語に関する他の研究では，人手で作成したデータセットを利用する手法が主流である．しかし，人手によるデータセットの構築では手間と時間がかかる．特に，大規模なデータセットではコストも高くなる問題点がある．また，近年の自然言語処理では教師あり機械学習を含むデータ駆動型のアプローチが主流である．しかし，データから得られるパターンを利用するデータ駆動型の手法では，単に教師データのパターンを学習する傾向にあり，言語の原理や構造の理解に貢献しない可能性がある．

そこで本研究では，コーパスにおける文脈情報と単語の対訳関係を利用することで，人手で作成したデータに依存せず全自動（人手作成の辞書に依存しない方法）で対義語を抽出する手法を提案した．本論文では，対義語には類似性が存在する点，対義語抽出を類似単語からの対義語・類義語の分類タスクとして考える観点から類義語についても言及するが，本研究の焦点は対義語抽出である．そのため，類義語抽出については追加実験として扱う．

目 次

第 1 章　はじめに **1**

第 2 章　関連研究 **3**

2.1　人手作成のパターンによる関係語抽出 3

2.2　教師データによる対義語・類義語の分類 5

第 3 章　問題点と目的 **6**

3.1　問題点 6

3.2　目的 6

第 4 章　対義語・類義語の性質 **7**

4.1　対義語・類義語の文脈類似性 7

4.1.1　類義語対における意味カテゴリと類似性の関係 8

4.1.2　対義語対における意味カテゴリと類似性の関係 8

4.1.3　意味の類似性と文脈の類似性 8

4.2　語義と翻訳の関係 9

4.2.1　言語学習における類似点 9

4.2.2　共通訳と語義の類似性 10

第 5 章　提案手法 **11**

5.1　定義 11

5.1.1　対義語の定義 11

5.1.2　類義語の定義 12

5.2　抽出手法 13

5.2.1　Word2Vec 13

5.2.2　FastAlign 14

5.3　抽出手順 15

5.3.1 対義語 15
5.3.2 類義語 16

第 6 章 実験設定 **17**
6.1 実験条件 17
6.1.1 Word2Vec 17
6.1.2 FastAlign 19
6.1.3 テストデータ 20
6.1.4 フィルタリング 21
6.1.5 データの枝刈り 22
6.2 使用データ 23

第 7 章 実験結果 **24**
7.1 対義語の抽出 24
7.1.1 DEV 実験 24
7.1.2 テスト実験 26
7.2 考察 27
7.2.1 不正解対の原因 27
7.2.2 対義語ではないが対義性を持つ単語対 28
7.2.3 DEV 実験の考察 29

第 8 章 追加実験 **30**
8.1 類義語の抽出 30
8.1.1 DEV 実験 30
8.1.2 テスト実験 32
8.2 不正解の原因考察 33

第 9 章 議論・今後の課題 **34**
9.1 議論 34
9.2 今後の課題 35
9.2.1 精度 35
9.2.2 抽出数 35

第 10 章 おわりに **36**

図 目 次

2.1 Samenko らの手法の構造 5

5.1 “左” と “右” の抽出例 15
5.2 “支援” と “救助” の抽出例 16

6.1 対訳単語の共通割合 21

表 目 次

2.1.1 構文パターンの例 4

5.1.1 対義語の例 11
5.1.2 類義語の例 12

6.1.1 Word2Vec の学習パラメータ 17
6.1.2 cos 類似度が高い上位 3 単語 18
6.1.3 FastAlign による対訳単語の例 19
6.1.4 テストデータの例 20
6.1.5 対訳単語の出現回数 22
6.2.1 データベース 23
6.2.2 処理済みデータ 23

7.1.1 対義語 DEV 実験の結果 25
7.1.2 対義語の実験条件と実験結果 26
7.1.3 人手評価の正解率 26
7.1.4 対義語の出力例 26
7.2.1 曖昧な単語対の例 28
7.2.2 cos 類似度のみのフィルタリングによる対義語の割合 29

8.1.1 類義語 DEV 実験の結果 31
8.1.2 類義語の実験条件と実験結果 32
8.1.3 人手評価の正解率 32
8.1.4 類義語の出力例 32

9.1.1 提案手法による対義語と類義語の相違点 34

第1章　はじめに

本論文では，対義語と類義語の抽出について説明する．しかし，本研究の主要な焦点は対義語にあり，対義語抽出の一環として類義語を扱う．そのため，類義語抽出については追加実験で行う．

対義語と類義語の分類・抽出は自然言語処理分野において重要なタスクである．対義性・類似性等の言葉の関係性を原理的に解明することで言語の理解や言語処理の向上に貢献する．しかし，類義語抽出に関する研究が多い反面，対義語抽出に関する研究は少ない．その理由として，対義の関係にある単語を抽出するタスクが非常に困難な点が挙げられる．そのため，対義語抽出に関する他の研究では，人手で作成したデータセットを利用する手法が主流である．しかし，人手によるデータセットの構築では手間と時間がかかる．特に大規模なデータセットではコストも高くなる問題がある．また，近年の自然言語処理では教師あり学習を含むデータ駆動型のアプローチが主流である．しかし，データから得られるパターンを学習するデータ駆動型の手法では，単に教師データのパターンを学習する傾向にあり，言語の原理や構造の理解に貢献しない可能性がある．

そこで本研究では，人手で作成したデータに依存しない全自動の対義語抽出を目的とする．抽出の手法として，対義性・類似性の性質に基づいた提案を行う．具体的には，「**対義語・類義語の文脈の類似性**」と「**語義と翻訳の関係**」の性質に着目することで，対義語抽出のタスクを類似単語からの対義語と類義語の分類タスクとして捉える．

本研究の主な主張点を以下に整理する．

- 本研究は，対義語抽出において人手で作成された辞書またはデータセットに依存しておらず，抽出が全自動であるという新規性がある．
- 本研究は，教師あり学習を使用していない点で，単なる正解ラベルのパターン学習ではなく，言語理解に対する原理的なアプローチを試みている．
- 対義語対・類義語対同士は類似する文脈で出現する性質から，対義語は類義語の一部であり，類義語抽出の一環と捉えることができる．
- 翻訳が言語を横断して同一の内容を表現する性質から，共通する翻訳を持つ単語対は共通する意味を持つ単語対として捉えることができる．

本論文の構成は以下の通りである．

第 2 章 対義語・類義語抽出に関連する他の研究を説明する．

第 3 章 関連研究における問題点を説明する．

第 4 章 本研究で取り扱う対義語・類義語の性質について説明する．

第 5 章 提案手法について説明する．

第 6 章 本研究における実験内容の詳細と設定について説明する．

第 7 章 対義語抽出の実験について説明する．

第 8 章 追加実験として類義語抽出について説明する．

第 9 章 議論と今後の課題について記述する．

第 10 章 本研究における内容のまとめを行う．

第2章　関連研究

本章では，過去の研究における対義語と類義語の抽出・分類に関する論文を紹介する．2.1 節では，Chklovski[1] らが行った構文パターンを利用した関係語の抽出の研究について記述する．2.2 節では，Samenko[2] らが行った単語埋め込みモデルを利用した対義語と類義語の分類に関する研究について記述する．

2.1　人手作成のパターンによる関係語抽出

Chklovski ら [1] は，特定の構文パターンを含む文章を検索することで**類似性**，強度，**対義性**，有効化，発生順序の 5 種類の関係にある動詞対の抽出を半自動的な手法で試みた．構文パターンは，web 上で十分に共起する動詞対の調査により手動で選択され，単語 50 対から 35 パターンが得られた．実験結果は，2 人の評価者によって判定された．2 人のうち 1 人でも正解と判断する論理和の条件で，正解率は類似性 63.4%，強度 75.0%，対義性 50.0%，有効化 100%，発生順 72.9%となった．構文パターンの例を表 2.1.1 に，各関係語の意味を以下に示す．

表 2.1.1: 構文パターンの例

関係	構文パターン
類似性	Xed and Yed to X and Y
強度	X even Y not only Xed but Yed
対義性	either X or Y whether to X or Y
有効化	Xed by Ying the to X by Ying or
発生順序	to X and then Y to X and later Y

類似性 :同じ行動を示す動詞に関して，異なる含意が生じる関係
例) “最大化する” と “高める”，“生産する” と “創造する”

強度 :2 つの動詞が類似している場合に，一方が強力で徹底的・包括的な関係
例) “汚染する” と “毒する”，“許可する” と “権限を与える”

対義性 :2 つの動詞が，意味的に対立・反対の関係
例) “組み立てる” と “分解する”，“禁止する” と “許可する”

有効化 :一方の動詞に対して，もう一方の動詞が前提として達成されている関係
例) “評価する” と “レビューする”，“達成する” と “完了する”

発生順序 :2 つの動詞が時間的に前後する関係
例) “結婚する” と “離婚する”，“入学する” と “卒業する’

2.2 教師データによる対義語・類義語の分類

単語をベクトル空間にマッピングすることで単語間の意味的な関係や類似性を数学的に表現可能にする単語埋め込みモデルの手法がある．しかし，従来のモデルでは，対義語と類義語は cos 類似度 (ベクトルの距離) の分布が類似する傾向にある．Samenko ら [2] は，現代の単語埋め込みモデルには，対義語と類義語を区別するための情報が含まれると主張し，教師データを利用することで，対義語と類義語を区別する新たな単語埋め込みモデルを提案した．対義語・類義語の辞書を教師データとすることで，ベクトル空間における類似単語の距離が小さく，対義単語の距離が大きくなるように最適化を行った．生成された新しいモデルでは，テストデータに使用した対義語対と類義語対の cos 類似度が異なる分布を示しており，主張の妥当性が確認された．

具体的には，通常の「単語埋め込みモデル」に加えて，「類義語データセットで調整したモデル」と「対義語データセットで調整モデル」の計 3 種類のモデルに対して，損失関数の一つである Triplet Loss の手法を利用した．Triplet Loss は，Anchor，Positive，Negative と呼ばれる 3 種類のサンプルから構成され，Anchor と Positive の距離を近づけ，Anchor と Negative の距離を遠ざける計算を行う．つまり，「単語埋め込みモデル」，「類義語で調整したモデル」，「対義語で調整モデル」はそれぞれ Anchor，Positive，Negative に対応している．図 2.1 に提案手法の構造を示す．

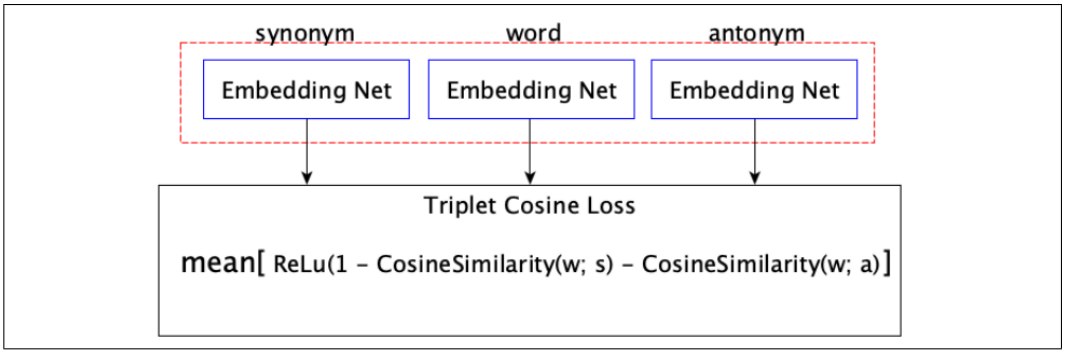

図 2.1: Samenko らの手法の構造

第3章　問題点と目的

第 2 章で説明した関連研究を踏まえて，対義語と類義語の分類・抽出における問題点と，本研究における目的を記述する．

3.1　問題点

対義語抽出に関する研究は，類義語抽出に関する研究と比較して少なく，対義語のみを抽出する研究は基本的に無い．その理由として，対義の関係にある単語を抽出するタスクが非常に困難な点が挙げられる．そのため，人手で作成したデータセットを利用する手法が主流である．第 2 章で紹介した研究でも，人手で選択した構文パターンと対義語・類義語の辞書を利用していた．しかし，人手によるデータセットの構築では手間と時間がかかる．特に，大規模なデータセットではコストも高くなる問題点がある．また，近年の自然言語処理では 2.2 節に示す教師あり機械学習を含むデータ駆動型のアプローチが主流である．しかし，データから得られるパターンを利用するデータ駆動型の手法では，単に教師データのパターンを学習する傾向にあり，言語の原理や構造を理解することが困難な可能性がある．

3.2　目的

人手で作成したデータに依存せず，完全自動な対義語抽出の提案を目的とする．

第4章　対義語・類義語の性質

本章では，抽出の手法に利用する対義語と類義語の性質について記述する．4.1 節では，「対義語・類義語の文脈類似性」について記述する．4.2 節では，「語義と翻訳の関係」について記述する．

4.1　対義語・類義語の文脈類似性

言語学において，単語の意味的な連想を考慮して特定の概念と関連付けることで，単語を異なる意味のカテゴリに分類することがある．例えば，「車」は「交通手段」と連想され，「交通」という概念と関連付けられる．単語と関連付けられた概念は意味カテゴリと呼ばれ，言語が持つ多様な側面を捉えて理解するための手段となる．

対義語対・類義語対同士は，単語の意味カテゴリが共通するため類似性を持つ．さらに，意味カテゴリが共通する単語対は意味が類似するため，同一文で置き換えられる可能性がある．そのため，本研究では類義語対・対義語対は前後の文脈が類似すると仮定する．対義語・類義語の文脈が類似すると考える背景を 4.1.1 小節，4.1.2 小節，4.1.3 小節に分けて説明する．

4.1.1 類義語対における意味カテゴリと類似性の関係

類義語対は意味カテゴリが共通するため，意味が類似する．

類義語同士は，共通する意味を示すため，単語を構成する意味カテゴリが共通する．例えば，“喜び” と “幸せ” はポジティブな感情を表しており，感情を示す点や感情の方向性という点で共通する意味カテゴリである．しかし，完全には共通せず，一部異なるニュアンスを含む．例えば，“喜び” と “幸せ” では，“喜び” は，外部からの刺激で引き起こされる比較的短い期間の感情を表現する傾向がある．対して，“幸せ” は，持続的で定常的な感情を表現する傾向にある．したがって，意味カテゴリの共通性により意味の類似性が生じている．反対に，意味カテゴリが共通しない単語同士は相互に関連性がないため，単語対を成さないと考えることができる．

4.1.2 対義語対における意味カテゴリと類似性の関係

対義語対も類義語対と同様に意味カテゴリが共通するため，意味が類似する．

対義語同士は，反対の性質を示すため，“対義語” と “類義語” は対義の関係にある．しかし，対義語対は単語の要素全てが対義の関係ではない．むしろ，大部分の要素は類似しており，意味カテゴリも類義語と同様に共通する．例えば，対義語対である “白” と “黒” は明暗の対比で対義の関係にあるが，色・光度という共通する意味カテゴリに属する．したがって，対義語対は意味カテゴリにおける大部分が共通するため，意味が類似すると考えることができる．

4.1.3 意味の類似性と文脈の類似性

意味の類似性と文脈の類似性は，相互関係にある．

自然言語処理の分野には，分布仮説 [3] という重要な概念が存在する．この仮説によると，「単語の意味はその周囲の文脈によって決定される」と考えられている．そのため，単語が持つ意味は，単語が使用される文脈に依存し，同じ文脈で使用される単語は類似する意味を持つ可能性が高いとされる．また，同じ文脈で使用される単語は，意味的に代替可能である性質を持つ．

4.2　語義と翻訳の関係

単語の性質を示す語義と単語を別言語に変換する翻訳は，言語学習の観点で類似点を持つ．また，「語義が同じ」と「翻訳が同じ」は，同一の性質を表す．そのため，「語義」と「翻訳」の間には対応関係が存在すると仮定する．対応関係が存在すると考える背景を 4.2.1 小節と 4.2.2 小節に分けて説明する．

4.2.1　言語学習における類似点

単一言語における語義と二言語間における翻訳には，言語学習の観点から類似点がある．

言葉は，規定された記号や音の組み合わせであり，単語の性質や概念を示す「語義」は一般的に辞書的な定義によって決定される．つまり，単語を適切に使用するためには，辞書的な定義を事前に理解している必要がある．そのため，国語の学習では，未知の単語に対応する語義を理解することで語彙を増やすことができる．例えば，“本” という単語を理解するためには「書籍．書物．[1]」といった語義を覚える必要がある．

一方で，「翻訳」は異なる言語で表現された共通の情報である．しかし，各言語は独自の知識体系を持つため言語同士の直接的な対応を考えることは困難である．つまり，翻訳を適切に行うためには，言語間の言葉の対応を事前に理解している必要がある．そのため，外国語の学習では，別言語の単語に対応する母国語の単語を理解することで語彙を増やすことができる．例えば，“book” という単語を理解するためには “本” との対応を知る必要がある．

[1]広辞苑 第 6 版，岩波書店，2008 年

4.2.2 共通訳と語義の類似性

同じ翻訳を持つ単語であれば同じ意味を持つ単語であり，共通する翻訳を持つ単語対は類似する意味を持つ．

翻訳は，言語を横断して同一の内容を表現することを目的としており，適切な翻訳であればソース文とターゲット文は言語が異なるだけで同じ内容である．しかし，単語レベルの翻訳では，意味に関する制限が少ない．そのため，各言語における独自の言語体系が影響することで言語間の概念にずれが生じ，複数の翻訳対応を取ることがある．しかし，複数の対応は単語の概念を補完し合うものであり，意味的には類似すると考えられる．したがって，共通する翻訳を持つ単語対は意味的に関連している可能性がある．例えば，「冬になると、私は首にマフラーを巻きます。」と「*In winter, I wrapped a scarf around my neck.*」は言語が異なるだけで，内容は同一である．ただ，単語レベルの翻訳において “首” を翻訳する場合，単に頭部と胴を繋げる “neck” を示すのか，胴から切り離された頭部である “head” を示すのか曖昧であり複数の翻訳対応を取る可能性がある．しかし，共通する “首” を翻訳に持つ “neck” と “head” は関連する単語で概念を補完すると考えることができる．

第5章　提案手法

本章の流れは次の通りである．まず 5.1 節では，4 章で説明した「対義語・類義語の文脈類似性」と「語義と翻訳の関係」に基づき，本研究における対義語と類義語の定義を記述する．次に 5.2 節では，定義に基づき実際に実験を行うための具体的な手法とその説明を記述する．最後に 5.3 節では，定義に基づく対義語・類義語抽出において手法の処理の手順を記述する．

5.1　定義

5.1.1　対義語の定義

- 意味カテゴリが共通　(4.1 節)
- 文脈が類似　(4.1 節)
- 翻訳 (=意味) が異なる　(4.2 節)

定義に基づいた対義語の例を表 5.1.1 に示す．“左” と “右” の例では，同一文で単語が置き換え可能であり，“東” と “西” の例では，同一文内で単語が共起するため，文脈が類似する．

表 5.1.1: 対義語の例

“左” と “右”
交差点を左に曲がる．　交差点を右に曲がる．
左 = left ≠ right = 右
“東” と “西”
太陽は，東から昇り西に沈む．
東 = east ≠ west = 西

5.1.2 類義語の定義

- 意味カテゴリが共通　(4.1 節)
- 文脈が類似　(4.1 節)
- 翻訳 (=意味) が同じ　(4.2 節)

定義に基づいた類義語の例を表 5.1.2 に示す．“病気” と “病” の例では，訳が完全に一致しており，“支援” と “救助” では一部の訳が一致しており，意味が類似する．

表 5.1.2: 類義語の例

“病気” と “病”
病気にかかる．　　病にかかる．
病気 = disease = 病
病気 = illness = 病
“支援” と “救助”
その団体は支援プログラムを実施する．
その団体は救助プログラムを実施する．
支援 = help = 救助
支援 = support ≠ rescue = 救助

5.2 抽出手法

対義語対と類義語対は意味カテゴリが類似するため，類似する文脈で出現する共通点を持つ．しかし，表現する意味が正反対なため，翻訳における単語の対応関係では相違点を持つ．そこで本研究では，文脈が類似する単語対の抽出を行い，翻訳の一致・不一致を調査することで対義語と類義語の分類を行う．つまり，対義語抽出のタスクを類似単語からの対義語と類義語の分類タスクとして実行可能と考える．そこで，抽出の手段として**文脈の類似性**と**単語の翻訳**の情報の利用を提案する．文脈情報の取得には Word2Vec[4][5] で単語埋め込みモデルを生成し，単語の翻訳には FastAlign[6] で日本語-英語の対訳単語対を取得する．

5.2.1 Word2Vec

Word2Vec[4][5] は，単語をベクトル空間にマッピングすることで，単語間の意味的な関係や類似性を数学的に表現する埋め込みモデルの一つである．大規模なコーパスにおいて，ターゲット単語の周囲で共起する単語のパターンを学習することで，文脈の情報を利用したベクトル空間を生成する．Word2Vec における単語同士の関係性は cos 類似度によって表現され，２つの単語ベクトル A と B の cos 類似度は式 5.1 のように計算される．文脈が類似する単語対はベクトル空間内における距離が近いため cos 類似度が高い．

$$Cosine\_Similarity(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{5.1}$$

本研究では, 文脈が類似する単語対を取得するために Word2Vec の有する Word Similarity Search(単語の類似語検索) の機能を利用する．Word Similarity Search では，ターゲット単語と他の単語の cos 類似度を計算し，類似度が高い順に単語を N 個出力することができる．そこで，N=1 として単語双方向からの cos 類似度が最も高くなる単語対を抽出する．具体的には，ターゲット単語 w に類似する単語 sw を出力し，逆に，sw に類似する単語が w となる単語対を収集する．

5.2.2 FastAlign

FastAlign[6] は，IBM model 2[7] のパラメータを簡素化して計算効率を向上させた統計的機械翻訳の手法であり，大規模な対訳コーパスにおける言語間の単語対応の取得に特化している．以下で IBM model 2 の手法，FastAlign と IBM model 2 との相違点を説明する．

IBM model 2

IBM Model 2 では，適切な単語対応を取得するために，ソース言語とターゲット言語の単語間の対応関係をモデル化するアライメント確率と，ソース言語の文がターゲット言語に翻訳される確率分布をモデル化する翻訳確率を学習する．モデル化の手法として EM アルゴリズムが利用されるが，より精密なアライメントと翻訳確率のモデリングを実現するためにモデルが複雑となるため，大規模なデータセットに対する学習では計算リソースを必要とする．

IBM model 2 との相違点

FastAlign では，IBM model 2 と同様に対訳コーパスからアライメント確率と翻訳確率を学習するが，モデル化の手法として対数線形型モデルを採用する．EM アルゴリズムは，最適化等で指数関数の計算が必要となるが，対数線型モデルでは，対数を取ることで，指数関数が加法として表現できるため，計算が容易になる．そのため，計算コストが低く高速な単語対応が可能になり，大規模なコーパスに対しても有効である．

5.3 抽出手順

手順を以下に示す．J1 と J2 は日本語単語対であり，cos 類似度が互いに最大となる単語対を選択する．また，E1 は J1，E2 は J2 の対訳英単語である．

5.3.1 対義語

手順 1 テストデータにおいて cos 類似度が相互に最大となる J1，J2 を抽出

手順 2 J1，J2 の対訳単語 E1，E2 を FastAlign から取得

手順 3 E1，E2 が共通しない J1，J2 を対義語として出力

この方法により，文脈が類似 (=意味カテゴリ共通) し，翻訳 (=意味) の異なる単語対の抽出が可能となる．抽出の例を図 5.1 に示す．

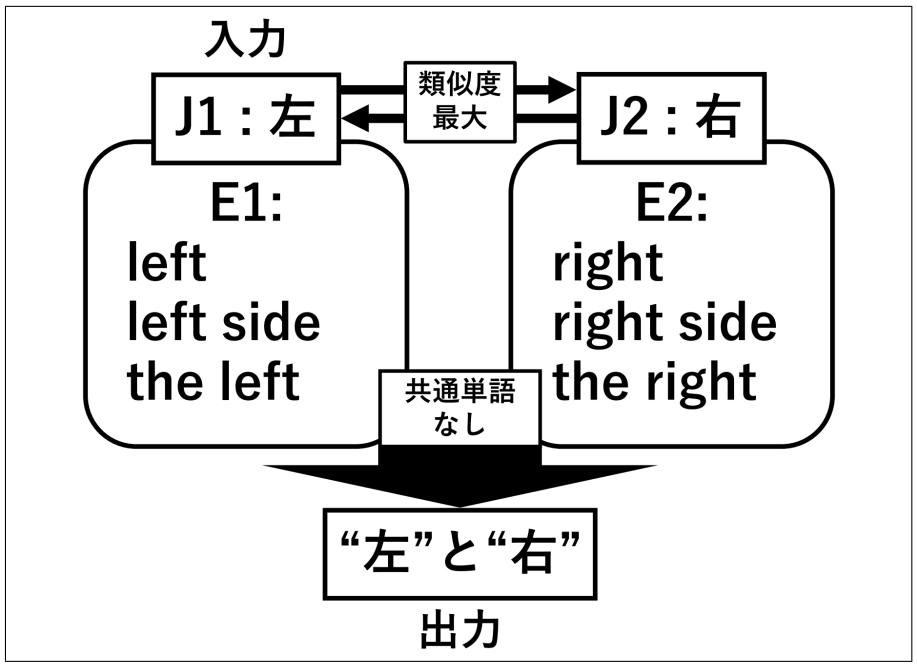

図 5.1: “左” と “右” の抽出例

5.3.2 類義語

手順 1 テストデータにおいて cos 類似度が相互に最大となる J1，J2 を抽出

手順 2 J1，J2 の対訳単語 E1，E2 を FastAlign から取得

手順 3 E1，E2 が共通する J1，J2 を類義語として出力

この方法により，文脈が類似 (=意味カテゴリ共通) し，翻訳 (=意味) が同じ単語対の抽出が可能となる．抽出の例を図 5.2 に示す．

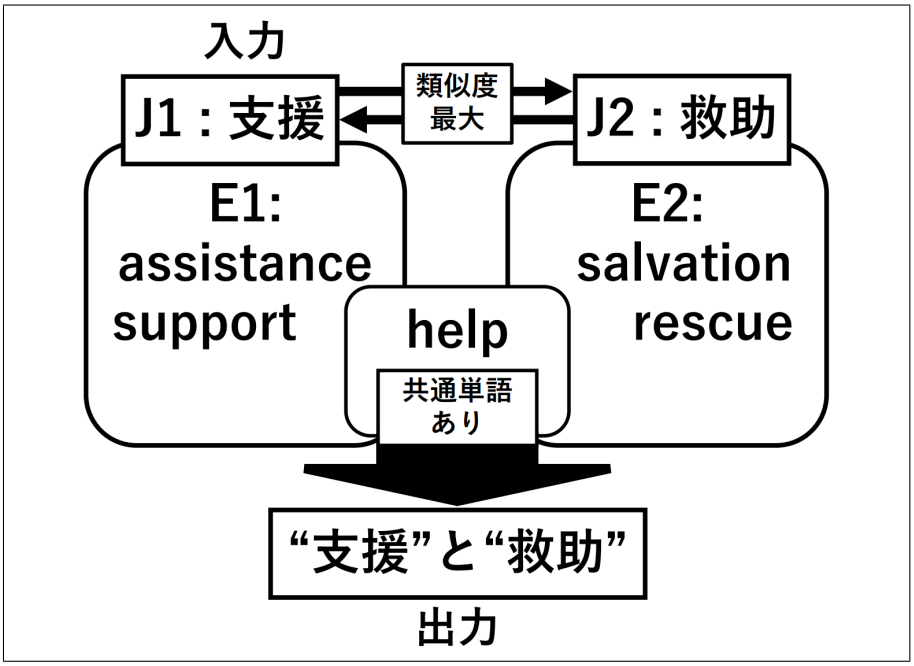

図 5.2: “支援” と “救助” の抽出例

第6章　実験設定

6.1　実験条件

6.1.1　Word2Vec

Word2Vec には，東北大学が公開している日本語 Wikipedia エンティティベクトル [8] のスクリプトを利用した．パラメータはデフォルトを選択した．パラメータの詳細は表 6.1.1 に示す．

また, 学習データには 2023/9/11 における日本語 Wikipedia の記事を使用した．Wikipedia 記事は，mecab-python3 1.0.6 を利用して単語ごとに分割した．また，文章ではなく一文ごとの学習を行うため，記事を “。” 文字おきに改行することで，学習時の window_size における前後の文の影響を考慮した．

表 6.1.1: Word2Vec の学習パラメータ

size	200
window size	10
sample size	10
min count	10
epoch	5

size：生成されるベクトルの次元数

window size：ターゲット単語の文脈として考慮される前後の単語の範囲

sample size：学習データ内における高頻度単語が与える
モデルへの影響を軽減するパラメータ

min count：学習データ内における単語のモデルに使用する最低出現回数

epoch：学習データ全体を繰り返し学習する回数

Word2Vec の出力例

日本語 Wikipedia で学習した Word2Vec モデルを利用して，入力単語に対する cos 類似度が高い上位 3 単語を出力した．表 6.1.2 に結果を示す．使用した日本語 Wikipedia のデータを表 6.2.1 に示す．

表 6.1.2: cos 類似度が高い上位 3 単語

入力単語	出力単語	cos 類似度
有罪	無罪	0.9376630783081055
	有罪判決	0.8941740393638611
	起訴	0.8837166428565979
無罪	有罪	0.9376630187034607
	無罪判決	0.8937172293663025
	起訴	0.8362336754798889
未来	未来を	0.7669742107391357
	未来へ	0.7356337308883667
	未来のために	0.7033901214599609
過去	全て	0.6904784440994263
	実際	0.6588191986083984
	含め	0.6540564298629761

6.1.2 FastAlign

FastAlign には，Apache License, Version 2.0 で提供されるオープンソースのソフトウェアを利用した．

また，使用するデータとして JParaCrawl[9] を使用した．JParaCrawl は，多言語で書かれた並列な Web 記事から並列なデータを収集することで，対訳コーパスを構築するプロジェクトである．本研究では，JParaCrawl において対訳データの品質を示す bicleaner の値が 0.70 以上のデータを使用した．日本語文のトークン化には mecab-python3 1.0.6 を利用した．

FastAlign の出力例

JParaCrawl のデータを使用して取得した日本語-英語の単語対応の結果を表 6.1.3 に示す．使用した JParaCrawl のデータは表 6.2.1 に示す．

表 6.1.3: FastAlign による対訳単語の例

入力単語：日本語	出力単語：英語 (単語の対応する頻度が高い順)
有罪	guilty, was, found, convicted, conviction, guilt, verdict, alleged, handed, judged, murder, points, returneda
無罪	innocence, acquitted, innocent, guilty, not, charge, found, was, acquittal, defendant, establishes, he, insisted, persuaded, pleaded, ”not, He, allegations, conviction, declared, escaped, fact, guilty”, plead, protest
未来	future, bright, futuristic, Different, Future, We, brilliant, expectations, hold, people’s, pictured, promising, responsible, wannabe
過去	past, record, years, last, highest, history, The, amount, has, Past, Reports, drastic, have, hit, look, All, But, Forget, Indeed, It, Parental, They, What, accelerated, agency, all_time, back, based, buried, depletion, don’t, doubled, figure, forgive, high, in, joy, knocking, largest, lived, nonpayment, referred, residents, sales, scourged, shed, spheres, the

6.1.3　テストデータ

日本語の文を単語に分割したデータをテストデータとする．mecab-python3 1.0.6 を使用して日本語文を単語ごとに分割し，単語を取得した．

また，使用するデータとして電子辞書等から抽出して作成された日英単文対訳文 [10] の日本語文を使用した．

テストデータの例

日本語の文を分割して取得したテストデータの例を表 6.1.4 に示す．使用した日英単文対訳文のデータは表 6.2.1 に示す．

表 6.1.4: テストデータの例

青, 青々, 青い, 青かび, 青く, 青ざめ, 青果, 青筋, 青空, 青菜, 青山, 青山学院大,
青酸, 青酸カリ, 青写真, 青春, 青書, 青少年, 青色, 青信号, 青森, 青青, 青虫, 青天,
青天の霹靂, 青島, 青銅, 青年, 青梅, 青白, 青白い, 青白く, 青函, 青木, 青葉, 静,
静か, 静けさ, 静まっ, 静まら, 静まり, 静まりかえっ, 静まり返っ, 静まる, 静め,
静める, 静岡, 静穏, 静観, 静香, 静止, 静寂, 静粛, 静水, 静男, 静電気, 静物, 静脈,
静養, 斉, 斉射, 斉唱, 斉藤, 税, 税引き, 税額, 税関, 税金, 税込み, 税収, 税制,
税法, 税務, 税務署, 税率, 脆い, 脆く, 脆弱, 脆性, 隻, 席, 席巻, 席次, 席上, 惜,
惜し, 惜しい, 惜しくも, 惜しげ, 惜しま, 惜しみ, 惜しむ, 惜しん, 惜敗, 惜別, 斥侯,
斥候, 昔, 昔かたぎ, 昔ながら, 昔なじみ, 昔日, 昔話, 石, 石けん, 石こう, 石ころ,
石井, 石灰, 石灰岩, 石垣, 石垣島, 石巻, 石丸, 石器, 石鹸, 石原, 石膏, 石材, 石寺,
石狩川, 石清水八幡宮, 石川, 石川島播磨重工業, 石造り, 石炭, 石段, 石頭, 石碑,
石綿, 石毛, 石油, 積, 積ま, 積み, 積み下ろし, 積み荷, 積み込ま, 積み込み, 積み込む,
積み込ん, 積み残さ, 積み残し, 積み重なっ, 積み重なる, 積み重ね, 積み出し,
積み上げ, 積み替え, 積み替える

6.1.4　フィルタリング

Word2Vec の cos 類似度

表 6.1.2 より，Word2Vec において cos 類似度が高い順に単語を出力した場合，入力が “有罪” や “無罪” では cos 類似度が高く，単語の類似性も高いが，“未来” や “過去” では cos 類似度が低く，単語の類似性も低い．そのため，抽出手順において cos 類似度の値に閾値を設けることで類似性の低い単語のフィルタリングを行う．具体的な cos 類似度の値は DEV 実験の結果で決定する．

対訳単語の共通割合

表 6.1.3 より，FastAlign による対訳単語の取得では “無罪：guilty” 等の誤った対応が確認できる．しかし，5.1 節の定義より対義語と類義語の分類に翻訳の一致・不一致を利用するため，翻訳の精度が抽出結果に大きく影響する．そこで，翻訳の誤りを前提として対訳単語の共通割合を考える．抽出手順において日本語単語対の各対訳に占める共通する対訳単語の割合で対義語・類義語の識別を行う．具体的な共通割合の値は DEV 実験の結果で決定する．

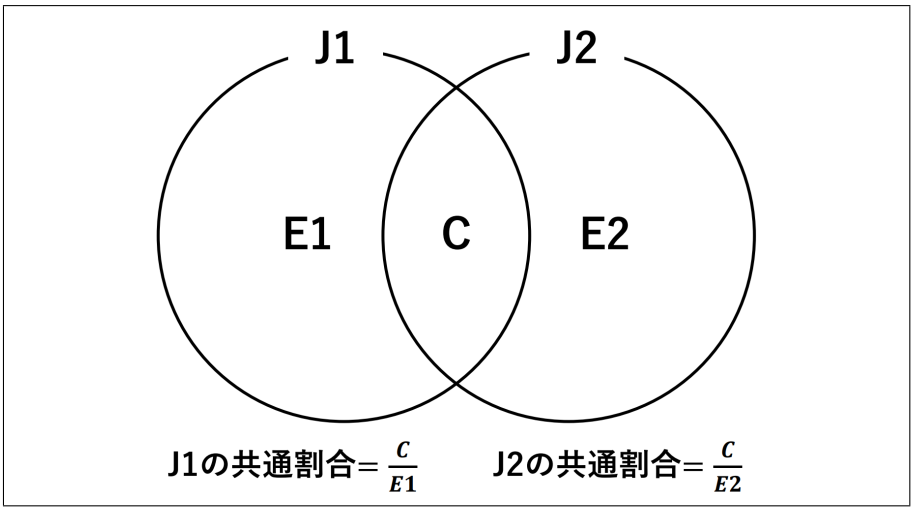

図 6.1: 対訳単語の共通割合

共通割合について，J1 と J2 の両方が閾値を満たす AND と，J1 と J2 の一方が閾値を満たせば良い OR の 2 種類の方式で実験を行った．

6.1.5 データの枝刈り

- 英数字は対義語・類義語になり得ないため，英数字を含む単語対を除外する．

  例) “6” と “8”，“1961” と “1971”

- 同一単語の読み仮名の関係を削減するために，平仮名のみで構成される文字列を含む単語対の除外

  例) “大学” と “だいがく”，“ネコ” と “ねこ”

- 頻度が少なく信頼性の低い対訳単語の影響を軽減するために，JParaCrawl で取得した対訳英単語の合計の出現回数が５以下の日本語単語を除外 (表 6.1.5 に例を示す)

表 6.1.5: 対訳単語の出現回数

5 以下の単語			
日本語単語	対訳英単語	数	合計
瞑想	meditation	2	2
紆余曲折	turns	2	4
	twists	2	
黴菌	infected	3	5
	became	2	

5 より多い単語			
日本語単語	対訳英単語	数	合計
姉妹	sisters	12	12
罠	in	4	10
	caught	2	
	rabbit	2	
	was	2	

6.2 使用データ

使用するデータベースを半分に分割し，各データは DEV 実験とテスト実験で使用する．DEV 実験は，6.1.4 小節で説明した Word2Vec の cos 類似度と対訳単語の共通割合の閾値を決定するため行う．その後，テスト実験において決定した閾値を適用する．使用したデータベースを表 6.2.1，データベースを使用して作成した対訳単語・テストデータを表 6.2.2 に示す．また，6.1 節に基づいて対訳単語対は JParaCrawl から生成し，テストデータは日英単文対訳文から生成した．

表 6.2.1: データベース

日本語 Wiki	全データ	1,385,000記事
	DEV 実験	693,590記事
	テスト実験	693,590記事
JParaCrawl	bicleaner0.7 以上	18,383,212 文
	DEV 実験	9,191,606 文
	テスト実験	9,191,606 文
日英単文対訳		163,188 文

表 6.2.2: 処理済みデータ

対訳単語対	DEV 実験	206,458,091 対
	テスト実験	206,464,657 対
テストデータ		43,148単語

第7章　実験結果

7.1　対義語の抽出

7.1.1　DEV 実験

対訳単語の共通割合を 20%，30%，40%以下の 3 種類，Word2vec の cos 類似度を 0.80，0.85，0.90，0.95 以上の 4 種類で実験を行った．評価者は著者 1 名である．表 7.1.1 に結果を示す．表中の曖昧の項目は 7.2.2 小節で説明する．

表 7.1.1: 対義語 DEV 実験の結果

共通割合	cos 類似度	方式	出力	人手評価 (100 対)	
				正解率	曖昧
20%以下	0.80 以上	AND	916	29 %	7
		OR	1,436	26 %	7
	0.85 以上	AND	362	35 %	9
		OR	570	38 %	11
	0.90 以上	AND	96	34 %(33 / 96)	14
		OR	160	40 %	13
	0.95 以上	AND	22	34 %(10 / 22)	0
		OR	33	51 %(17 / 33)	0
30%以下	0.80 以上	AND	1,074	30 %	8
		OR	1,728	20 %	3
	0.85 以上	AND	421	33 %	12
		OR	676	40 %	9
	0.90 以上	AND	119	43 %	13
		OR	188	41 %	8
	0.95 以上	AND	23	65 %(15 / 23)	0
		OR	29	62 %(18 / 29)	0
40%以下	0.80 以上	AND	1,277	25 %	8
		OR	2,072	24 %	7
	0.85 以上	AND	510	30 %	9
		OR	816	34 %	3
	0.90 以上	AND	151	49 %	13
		OR	223	45 %	11
	0.95 以上	AND	26	**69 %**(18 / 26)	0
		OR	36	**69 %**(25 / 36)	0

7.1.2　テスト実験

DEV 実験において正解率の最も高かった共通割合 40%以下，cos 類似度 0.95 以上の設定で実験を行い，結果を人手で評価した．評価者は 4 名である．実験条件と実験結果を表 7.1.2 に，評価結果を表 7.1.3 に，出力例を表 7.1.4 に示す．

表 7.1.2: 対義語の実験条件と実験結果

共通割合	cos 類似度	方式	出力数
40%以下	0.95 以上	OR	**44**

表 7.1.3: 人手評価の正解率

評価者 A	評価者 B	評価者 C	評価者 D	平均
59% (26 / 44)	55% (24 / 44)	59% (26 / 44)	57% (25 / 44)	**57%**

表 7.1.4: 対義語の出力例

J1	J2	評価者				cos 類似度	J1 共通割合	J2 共通割合
		A	B	C	D			
右舷	左舷	○	○	○	○	0.965	14.3%	26.3%
偶数	奇数	○	○	○	○	0.953	24.8%	81.4%
東岸	西岸	○	○	○	○	0.953	29.0%	7.7%
先輩	後輩	○	○	○	○	0.968	65.5%	31.8%
北西	南西	○	×	○	○	0.980	7.5%	23.0%
立像	坐像	○	○	○	×	0.957	35.2%	34.4%
火曜	木曜	×	×	×	×	0.978	0%	0%
少佐	中佐	×	×	×	×	0.955	12.8%	27.7%
筋骨	隆々	×	×	×	×	0.966	14.5%	100%
脚注	使い方	×	×	×	×	0.992	0.9%	0.02%

7.2 考察

7.2.1 不正解対の原因

テスト実験における不正解の原因を以下で考察する．

順序関係

曜日，階級，漢数字等の順序を持つ概念の単語対が出力された．順序を持つ概念は，単語自体が持つ意味よりも前後の概念の相対的な関係性が重要視されると考えられる．例えば，曜日は陰陽五行説や占星術等の特定の文脈において特定の意味や性質が存在する．しかし，一般的には一週間を区切るための単位として扱われ，単語自体の意味は無視される．つまり，単語を置き換えても文全体の内容に影響せず，同じ文脈で使用できる可能性が高い．そのため，cos 類似度が高くなる．また，各単語は順序の性質を除けば独立した概念であるため，翻訳が共通しないことで対義語抽出の手法において出力したと考えられる．

複合名詞・共起語

“筋骨” と “隆々”，“脚注” と “使い方” の様に複合名詞，または助詞を挟んで連続する名詞の対が出力された．ぶ連続する単語同士は，文内における位置が近いため，周囲に出現する文脈が同じと考えられる．例えば，「その部屋に足を踏み入れると、筋骨隆々の彼が目の前に立っていた。」という文において，“筋骨” と “隆々” それぞれの周囲に出現する名詞を考えると，共通して “部屋”，“足”，“彼”，“目”，“前” になる．そのため，cos 類似度が高くなることが理由と考えられる．また，“脚注” と “使い方” は，Wikipedia のフレーズ「脚注の使い方」として頻出するため，Word2Vec のモデルに使用したコーパスも影響すると推測される．

7.2.2 対義語ではないが対義性を持つ単語対

評価が難しい単語対を “曖昧な単語対” と判断した．人手評価では正解に含めないが，解釈次第では対義性を持つと考えることができ，提案手法が対義の関係にある単語を抽出できる性質を示す．表 7.2.1 に例を示す．

表 7.2.1: 曖昧な単語対の例

J1	J2	評価
協奏曲	ソナタ	△
ヤンキース	ドジャース	△
スコットランド	イングランド	△

協奏曲とソナタ：

共にクラシック音楽の形式である．一般的に，ソナタはピアノやヴァイオリン等の楽器が単独で演奏する形式が多い．対して，協奏曲はピアノやヴァイオリン等の楽器が主題を奏で，オーケストラが主題の補佐する形式が多い．つまり，音楽ジャンルとメイン楽器で共通するが，個と集団という観点で対義の関係にある．

ヤンキースとドジャース：

共に MLB の名門チームである．リーグは異なるが，ライバル関係にある．また，拠点が東岸と西岸であり，共に金満球団という観点で対比される．つまり，プロ野球リーグや球団の経済状況で共通するが，ライバル関係や拠点の方角の観点で対義の関係にある．

7.2.3 DEV 実験の考察

表 7.1.1 より，cos 類似度の閾値を大きくするほど対義語の出力数が減少し，正解率は上昇する．つまり，対義語は文脈の類似性による影響が大きく，より類似する文脈で出現することを示している．しかし，cos 類似度の閾値を高くするほど条件の制限が厳しくなることで出力数が減少すると考えられる．逆に，対訳単語の共通割合による閾値では，出力数と正解率に大きな変化がなく，対義語における対訳単語の共通割合の影響が小さいことが考えられる．

cos 類似度の閾値によるフィルタリングの影響を確認するために，閾値が 0.80 以上，0.95 以上の単語 100 対に含まれる対義語の割合を調査した．結果を表 7.2.2 に示す．

表 7.2.2: cos 類似度のみのフィルタリングによる対義語の割合

cos 類似度	単語対の数	対義語の数
0.80 以上	5,969	12 / 100
0.95 以上	124	55 / 100

表より，cos 類似度によるフィルタリングとして，閾値が高い条件の方が単語数に占める対義語対の割合が大きいことが確認できる．加えて，cos 類似度と対訳単語の共通割合でフィルタリングした表 7.1.1 と，cos 類似度のみでフィルタリングした表 7.2.2 の結果を比較すると，同じ cos 類似度の閾値でも表 7.1.1 の方が単語対に占める対義語の割合が大きいため，対訳単語の共通割合によるフィルタリングの有効性も確認できる．

第8章　追加実験

8.1　類義語の抽出

8.1.1　DEV 実験

対訳単語の共通割合を 60%，70%，80%以上の 3 種類，Word2Vec の cos 類似度を 0.80，0.85，0.90，0.95 以上の 4 種類で実験を行い，結果 100 対を人手で評価した．ただし，今回の実験において異表記を区別する処理を行っていないため，異表記は類義語として評価するものとする．評価者は著者 1 名である．表 8.1.1 に結果を示す．

表 8.1.1: 類義語 DEV 実験の結果

共通割合	cos 類似度	方式	出力	人手評価 (100 対)	
				正解率	曖昧
60%以上	0.80 以上	AND	1,693	88 %	1
		OR	2,620	77 %	0
	0.85 以上	AND	747	79 %	0
		OR	1,080	76 %	1
	0.90 以上	AND	186	64 %	3
		OR	266	57 %	1
	0.95 以上	AND	44	32 %(14 / 44)	0
		OR	64	25 %(16 / 64)	0
70%以上	0.80 以上	AND	1,262	82 %	0
		OR	2,238	76 %	0
	0.85 以上	AND	580	86 %	1
		OR	949	73 %	3
	0.90 以上	AND	149	63 %	3
		OR	238	68 %	2
	0.95 以上	AND	32	31 %(10 / 32)	0
		OR	55	29 %(16 / 55)	0
80%以上	**0.80 以上**	**AND**	739	**92 %**	2
		OR	1,753	82 %	2
	0.85 以上	AND	366	90 %	0
		OR	773	84 %	0
	0.90 以上	AND	92	64 %(59 / 92)	4
		OR	191	62 %	2
	0.95 以上	AND	18	11 %(2 / 18)	0
		OR	47	2 %(1 / 47)	0

8.1.2 テスト実験

DEV 実験において正解率の最も高かった共通割合 80%以上，cos 類似度 0.80 以上の設定で実験を行い，結果 100 対を人手で評価した．ただし，今回の実験において異表記を区別する処理を行っていないため，異表記は類義語として評価するものとする．評価者は 4 名である．実験条件と実験結果を表 8.1.2 に，評価結果を表 8.1.3 に，出力例を表 8.1.4 に示す．

表 8.1.2: 類義語の実験条件と実験結果

共通割合	cos 類似度	方式	出力数
80%以上	0.80 以上	AND	**751**

表 8.1.3: 人手評価の正解率

評価者 A	評価者 B	評価者 C	評価者 D	平均
86%	85%	94%	47%	**78%**

表 8.1.4: 類義語の出力例

J1	J2	評価者				cos 類似度	J1 共通割合	J2 共通割合
		A	B	C	D			
色調	色合い	○	○	○	○	0.825	84.6%	86.3%
住む	暮らす	○	○	○	○	0.853	83.9%	94.1%
特産	名産	○	○	○	○	0.844	86.8%	84.1%
多年草	多年生	○	○	○	×	0.831	84.3%	87.3%
等しく	等しい	○	○	○	×	0.887	87.0%	92.5%
出会う	出会っ	○	×	○	×	0.889	86.7%	87.3%
遅かれ	早かれ	×	○	×	×	0.976	99.3%	99.5%
兄	弟	×	○	○	×	0.948	88.5%	92.9%
前者	後者	×	○	○	×	0.966	85.7%	87.0%
防災	災害	×	○	×	×	0.802	92.5%	84.4%
小学校	中学校	×	○	○	○	0.915	94.3%	90.9%

8.2 不正解の原因考察

テスト実験における不正解として対義語が出力された．対義語が出力された原因を以下で考察する．

連続する対義語

“遅かれ” と “早かれ” 等の対義語対が出力された．「遅かれ早かれ」の様に対義の関係にある単語対を 1 つの熟語・フレーズとして表現する性質により，7.2.1 小節と同様の現象が起きていると考えられる．また，統計的機械翻訳の手法を利用しているため，対訳単語の対応はソース言語の単語とターゲット言語の単語の共起頻度が利用される．そのため，連続して出現する単語も同時に共起頻度が高くなると考えられる．つまり，意味的に関連性を持たない前後の単語が同一の対訳を取得することになり，誤った対訳単語が取得される．実際，JParaCrawl のデータにおいて，“遅かれ” と “早かれ” は 98%の文で連続しており，高い確率で翻訳が共通すると考えられる．

共通訳を持つ対義語

評価結果が分かれたが，著者は対義語と判断した “兄” と “弟” が出力された．“兄” と “弟” は共通して “brother” を訳語に持つ．そのため，翻訳をが共通する対義語が存在する場合，手法の限界であると考えられる．

第9章　議論・今後の課題

9.1　議論

テスト実験より，“文脈情報” と単語の “対訳関係” を利用することで，正解ラベルを必要とせず，対義・類似の関係にある単語対の自動的な抽出可能性が示された．また，DEV 実験の結果である表 7.1.1 と表 8.1.1 より，対義語は cos 類似度が高い条件で精度が高く，類義語は cos 類似度が低い条件で精度が高いことから，対義語と類義語の cos 類似度の相違点が確認できる．一方で，同じ cos 類似度の条件下でも，対義語は対訳単語の共通割合が 40%以下の条件で出力しており，類義語は対訳単語の共通割合 60%以上の条件で出力している点から，対義語と類義語における対訳単語の共通割合の相違点が確認できる．表 9.1.1 に提案手法による対義語と類義語の相違点を示す．

表 9.1.1: 提案手法による対義語と類義語の相違点

cos 類似度	対義語　＞　類義語
単語の共通割合	対義語　＜　類義語

9.2 今後の課題

本研究では，対義語・類義語抽出において “文脈情報” と単語の “対訳関係” を利用する手法を提案した．しかし，精度に関しては向上の余地があり，抽出数は今後の課題であると考える．

9.2.1 精度

不正解対の考察で指摘した原因を個別にフィルタリングすることで，誤り率の軽減が考えられる．さらに，閾値・パラメータのチューニングにより精度向上が考えられる．本研究の DEV 実験では，合計 7 種類の閾値で実施したが，網羅的に検証することで，より適切な設定が得られる可能性がある．加えて，Word2Vec や FastAlign のパラメータは単一の設定で実験したため，チューニングすることでモデルの改善も考えられる．

9.2.2 抽出数

本研究では，対義語の抽出可能性と精度に主眼を置いた．そのため，正解のサンプル数が少なく，考察が十分に行えていない可能性がある．また，対義語・類義語のカバー率についても考慮しておらず，今後の課題である．

第10章　おわりに

対義語と類義語の分類・抽出は自然言語処理分野において非常に重要なタスクである．しかし，対義の性質を抽出する困難性から対義語抽出の研究は少ない．加えて，対義語抽出の主流な手法には，人手で作成したデータセットを必要とするという問題点があった．

本研究では，対義語と類義語に関する「対義語・類義語の文脈類似性」と「語義と翻訳の関係」に基づき，対義語の持つ意味の類似性に着目した．対義語抽出のタスクを類似単語からの対義語と類義語の分類タスクと捉えることで，人手で作成したデータを必要としない全自動の抽出を試みた．テスト実験より，対義語抽出では 44 対が得られ，正解率は平均で 57%だった．加えて，正解とは評価しなかったが，解釈の方法次第では，対義の関係にある単語対が抽出できた．また，類義語抽出では 751 対が得られ，100 対の評価は平均で 78%の正解率だった．以上の結果より，提案手法の有効性が示せた．しかし，使用データの最適化，実験のパラメータ・条件の調整，新たなフィルターの導入等の改善により，今後さらなる精度の向上が期待できる．

謝辞

人手評価には，以下の 3 名の協力を得ました．感謝いたします．(名村太一，松本武尊，丸山京祐)

最後に, 本研究のご指導をいただきました鳥取大学工学部知能情報工学科自然言語処理研究室の村上仁一准教授, 村田真樹教授に厚く御礼申し上げます. そして, 平素よりお力添えいただいた自然言語処理研究室の皆様をはじめ, 参考にさせていただいた論文の著者の方々に深く感謝いたします.

参考文献

[1] Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 33–40, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[2] Igor Samenko, Alexey Tikhonov, and Ivan P. Yamshchikov. *Intuitive Contrasting Map for Antonym Embeddings*. IOS Press, October 2021.

[3] Zellig S. Harris. Distributional structure. *Word*, Vol. 10, pp. 146–162, 1954.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[6] Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[7] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, 1993.

[8] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. A joint neural model for fine-grained named entity classification of wikipedia articles. *IEICE Transactions on Information and Systems*, Vol. E101-D, No. 1, pp. 73–81, 2018. Special Section on Semantic Web and Linked Data.

[9] Makoto Morishita, Jun Suzuki, and Masaaki Nagata. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 3603–3609, Marseille, France, May 2020. European Language Resources Association.

[10] 村上仁一, 藤波進. 日本語と英語の対訳文対の収集と著作権の考察. 第一回コーパス日本語学ワークショップ, 2012.