令和 5 年度 卒業論文

特許文書からの効果語を抽出手法の改善

釧路工業高等専門学校

情報工学分野

ライモン ウィジャヤ

2023 年 12 月

論文要旨

特許文書の活用例として, 文書中の各種実験パラメータや物性値に代表される数値データを利用した数値ベースの物性予測 [3] や, 特許文書から物理量や数値を利用した技術分析 [1][2] などが知れられている.

先行研究 [3] では, 特許文書に出現する特許技術の効果を表す効果語を複合名詞であると仮定し, 特許文書中の実施例文と発明の効果文それぞれに出現する複合名詞の類似度の高い単語を効果語として抽出を行った. しかし技術に関する「処理名」や「物質名」などの効果を表さない単語が抽出されていた.

本研究では, 先行研究のプログラムの概念を基にし, 新しい機能を追加する. 一つ目は構文解析による「物質名」や「処理名」の除去を行うフィルタリング機能である. 二つ目は特許文書に特化した言語モデルを利用し, 効果語を抽出する. 本提案手法により「処理名」と「物質名」の抽出率削減および効果語の抽出精度の向上を目指す.

本手法により, 実験を行った結果, 効果語抽出率は 28%向上し, 処理名は 51%削減できた. しかし, 物質名は 6%, その他は 17%増加した. 実験結果を考察し改善方法を提案する.

目 次

図 目 次

表 目 次

第1章
序論

1.1 研究背景

マテリアルズ・インフォマティクスは, 2011 年に発表した「Materials Genome Initiative (MGI)」においても推進を打ち出されており, 分野のホットトピックスとなりつつある. 他にも, 各種実験パラメータや物性値に代表される数値データを活用することにより数値ベースの物性予測や技術の重要度を測ることもでき, 実験やシミュレーションからの大量の材料データで機械学習手法のインテリジェントな能力を活用して、新しい材料、機能、原理などを探がせるため, 数値データを取得することはマテリアルズ・インフォマティクスに限らず特許分析においても有用である [4][5]. そのため技術文献中より数値データを取得・活用することは重要である [3].

先行研究 [3] では, 特許文書中から教師なしで数値データである物理量と数値を抽出し, 技術の重要度を測るための特許分析に関する研究を行った. 数値データを抽出するために, 実施例文中の発明の効果を表す語である「効果語」の周辺に存在するとする. しかし, 研究の実験結果では効果語よりも効果語でない語の方が多く抽出してしまった. 例えば, 処理名や物質名などである. 処理名と物質名を削除することが

必要である. 従って, 本論文ではそれらを削減する手法を提案する.

1.2 本研究の目的

先行研究で課題とされている, 効果語ではない語: 物質名および手法名を取り除きまた取りこぼしのあった効果語を抽出し, 効果語の抽出精度を向上を目指す. 抽出精度を向上するには, 特許文書の発明の効果文をフィルタリングをする, 特許文書に特化した言語モデルを適用する.

効果語の抽出方法の概念は先行研究をもとにして, 効果語の抽出プログラムの改善を行う (図 1.1).

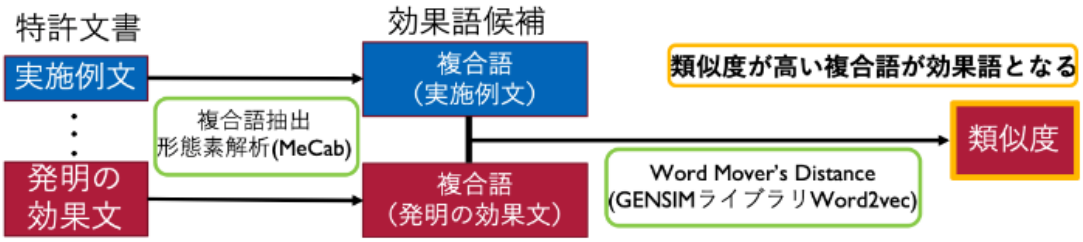


図 1.1: 効果語の抽出概念

1.3 本論文の構成

本論文は 6 章で構成される.

2 章では, 本研究に使われているライブラリと特許言語モデルのパラメータについて述べる.

3 章では, 新たな効果語抽出手法について述べる.

4 章では, 本研究の結果について述べる.

5 章では, 結果の考察について述べる.
最後に 6 章にて, 本論文の成果をまとめ, 今後の研究課題について述べる.

第2章
準備

本手法は形態素解析器 Spacy[1], 構文解析器 CaboCha[2], 分散表現 Word2Vec[3], 類似度計算は WordMoverDistance[4]を用いて python3 で実装する．各手法について説明する.

2.1 Spacy

「Spacy」は, 英語や日本語なども処理する多言語対応の深層学習ベースの自然言語処理ライブラリである．プロダクト向けに設計されており, 大量の文章の自然言語処理を行うアプリケーションの構築に適している [6].

本研究では, 形態素解析を行うには Spacy を利用し, 使用している辞書は「Ginza」である．「Ginza」は文章の単語分割と品質タグ付けを行う形態素解析であり, 人名・

[1]Spacy: https://spacy.io
[2]CaboCha: https://taku910.github.io
[3]Word2Vec: https://radimrehurek.com/gensim/models/word2vec.html
[4]WordMoverDistance: https://radimerhurek.com/gensim/auto_examples/tutorials/run_wmd.html

土地・組織などの固有表現を抽出する「固有表現抽出」を可能とする自然言語処理タスクを実行することが可能である.

2.2 CaboCha

CaboCha は, Support Vector Machines[5]に基づく日本語係り受け解析器である. 「係り受け解析」とは, 文章を「文節」または「単語」で分割した後, 「文節」または「単語」の「係り受け関係」(どの語がどの語を修飾しているか) を判別する処理である.CaboCha による構文解析の結果は図 2.1 に示す.

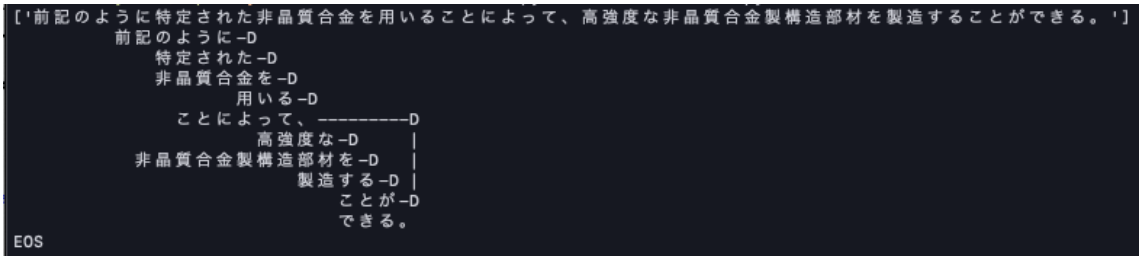


図 2.1: CaboCha による解析例

2.3 WordMoverDistance

本研究では, 語と語の類似度を求めるのが重要である. 類似度を求めるには WordMoverDistance(WMD) で語と語間の距離を計算し, 類似度の値が 0 になれば, 似ているあるいは同じということになる. WMD を使うには, Word2Vec の言語モデルが必要であり, そのモデルは python で用意されている Gensim という自然言語処理ライブラリにある. 本研究で使用する言語モデルは長岡技術科大学が作成した特許文書に特化した言語モデルであり, 学習データは NTCIR(National Institute of Informatics Test Collection for Information Resources Systems) の特許データ (1990-2000 年) を

[5]SVM : https://ja.wikipedia.org/wiki/サポートベクターマシン

用いて word2vec の学習を行う. しかし, 学習データは特許情報プラットフォーム[6]からそのまま HTML ファイルとして学習できないので前処理を行う必要がある. 形態素解析 Mecab0.996 を用いて以下の通り行う.

1. 英大文字から英小文字への変換
2. 全角英数字から半角英数字への変換
3. HTML タグ < .*? > と特許タグ【.*?】の削除
4. ストップワードは使用していない

全処理を行ったあと, Word2Vec モデルを作成する. 学習するためには, 学習パラメータがあり, そのパラメータは表 2.1 に示す. 「Window」とは学習に使う前後の単語数, 「Negative」とはネガティブサンプリングに用いる単語数, 「Min_count」とは n 回未満登場する単語を破棄, 「Embed_size」とは n 個までだけが学習の対象の数, 「Epoch」とはｎ回まで学習する数, 「Workers」とは複数のスレッドで処理するパラメータである.epoch 数については epoch=5 の場合, 処理に 1 週間程度かかる.

表 2.1: 学習パラメータ

Option	**Value**
Window	10
Negative	10
Min_count	10
Embed_size	300
Epoch	5
Workers	8

[6]https://www.j-platplat.inpit.go.jp/

第3章
本論

本研究では, 処理が二つ大きくわけられ, 「効果語の候補抽出部」と「効果語の抽出部」である. 図 3.1 にはそれぞれの部にどんな処理を行うかを示す. そして, 実験は, 特許庁が公開している特許情報プラットフォーム[6]から合金分野に関する特許文書を取得し 144 件を実験データとする.

3.1 効果語の候補抽出部

効果語の候補抽出部では処理が二つある. Cabocha によるフィルタリングと Spacy による形態素解析である.

[6]特許庁:https://www.j-platplat.inpit.go.jp/

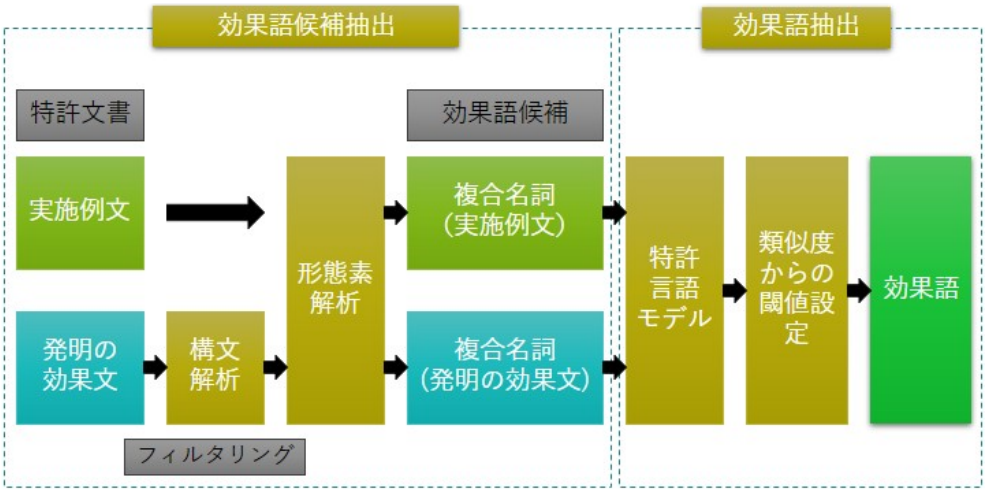


図 3.1: 特許文書からの効果語自動抽出の構成

3.1.1 Cabocha によるフィルタリング

フィルタリングの目的は発明の効果文に対し, 従属節に「〜による」と「〜することで」と「〜における」および「〜用いる」の表現がある場合, その従属節になる係先節を削除する. その表現にある語が処理名や物質名がたくさん含んでいて, 削除することで, 効果語の抽出を向上できると考えられる.

例えば, 発明の効果文にはこの文があるとする.

例 1： “ 前記のように特定された非晶質合金を用いることによって, 高強度な非晶質合金製構造部材を製造することができる. ”

この文には「〜による」という表現があり, Cabocha で解析し, 結果は図 2.1 に描いてあるが, 「前期のように」, 「特定された」, 「非晶質合金を」と「用いる」という文節は結局係先は「ことによって」に指す. なので, フィルタリングにより, その文節を削除し, 最後に「〜ことによって」という文節も削除する. 結果は以下のようになる.

結果： “ て, 高強度な非晶質合金製構造部材を製造することができる ”

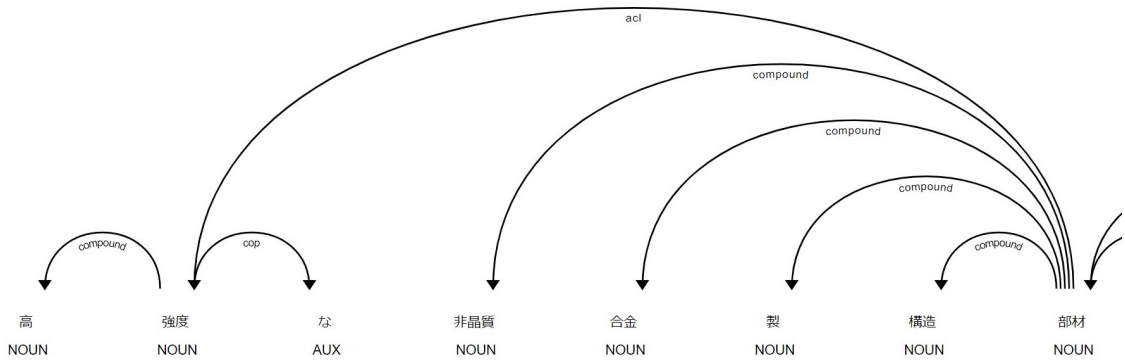


図 3.2: 例 2 の形態素解析結果

削除した文に対し, 形態素解析を行い, Spacy で複合名詞を抽出し, リストに保存する. これは効果語候補を抽出するために使い, 「禁止語リスト」という. 全発明の効果文を処理したら, フィルタリング処理は終りになる. その結果を次の形態素解析処理に渡す.

3.1.2 形態素解析

この処理は実施文とフィルタリングした発明の効果文を対象にし, 複合名詞を抽出する.Spacy では複合名詞を抽出する機能はないが, 次の工夫で複合名詞を抽出する. 例えば, このような文があるとする. 例 2 に使われている文は例 1 の結果である.

例 2 : “ 高強度な非晶質合金製構造部材を製造することができる. ”

これは Spacy で形態素解析を行われ, 結果は図 3.2 に示す.

図 3.2 をみると, 名詞と名詞の間には「Compound」複合という意味で, 例えば, 「高」と「強度」に書いてあるが, このようなケースは「高強度」として抽出する. 「非晶質合金製構造部材」も同じ処理で行われる.

例 3 : “ 合金組成の偏移が小さくて, しかも粒径が小さい電池用水素吸蔵合金粉末が

得られる. この水素吸蔵合金を負極に用いる電池の充放電サイクル寿命が長くなるとともに, そのばらつきが小さくなる. ”

この例ではフィルタリングにより, 「この水素吸蔵合金を負極に用いる」という分を削除され, 「水素吸蔵合金」は「禁止語リスト」に入れる. 残った文を形態素解析を行い, 抽出した複合名詞の一つは「非晶質合金製構造部材」がある. この二つ語が同じかどうかする判定方法は特許言語モデルを利用し, 結果がしきい値以下ならば, 二つ語が似ていて, 効果語候補として抽出しない. しきい値は目安に 2.4 にし, この二つ語の類似度を求める結果は 2.31 であり, しきい値の以下にあるから, 「非晶質合金製構造部材」は効果語候補に入れられない.

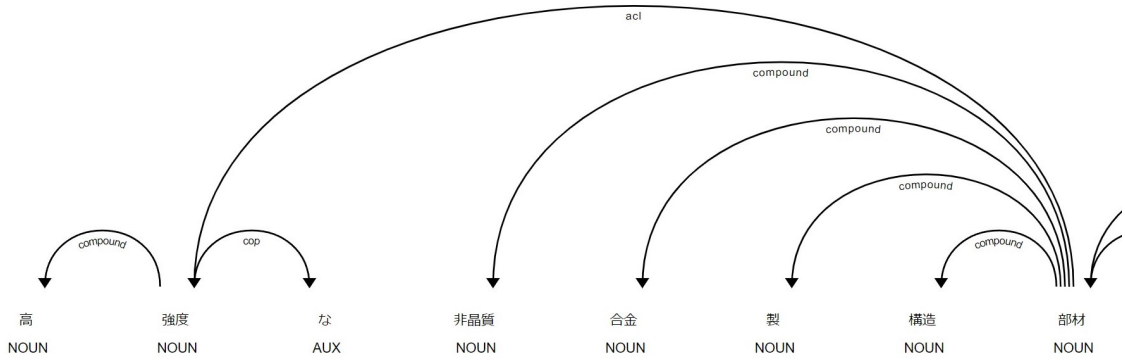


図 3.3: 例 3 の形態素解析結果の一部

3.2 効果語抽出部

効果語抽出部では, 効果語であるかどうかのしきい値を求め, 類似度を計算する処理である.

3.2.1 しきい値の求め方

しきい値の求めるには, まずは, 実験に使われている 144 件の特許文書に対して, 一つずつ発明の効果文と実施例文の効果語候補の類似度を求める. 求めた結果は「類似度結果配列」に保存する (図 3.4).「類似度結果配列」を昇順にソーティングし, グラフにする結果を図 3.5a に示している.

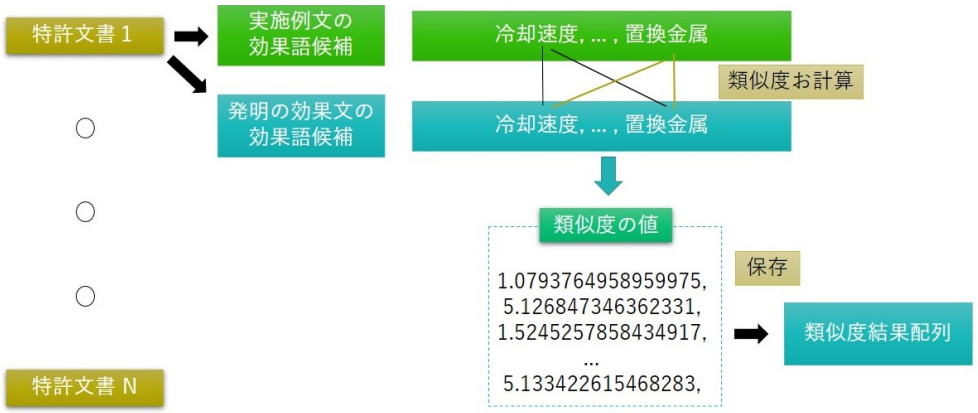
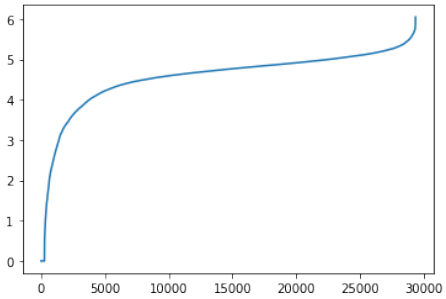
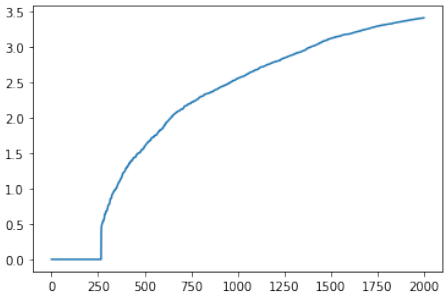


図 3.4: しきい値の求め方構成

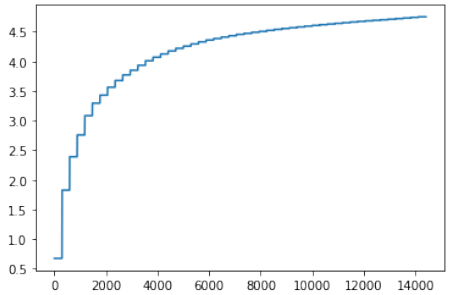
次は, データの相対的な位置を見るのに「百分位数」(Q) を利用する. 要するに, データを 1% ごとに分割し, 平均を求める. 類似度が 0 という結果が 300 個と近く, 全データは 29331 個あり, ちょうど 1% の全データになるから, 「百分位数」で計算する. 計算結果は図 3.1 に載っている. そして, 分位数間の差を求め, 表 3.1 をみると, 一番大きいのは Q_2 と Q_1 の間の差なので, しきい値の目安は Q_2 の結果で 1.8266(1.8) である. さらに, しきい値の調整を行うために, しきい値で 1.8 で抽出した効果語に存在し, しきい値 1.7 で抽出した効果語にない語を調べる. また, しきい値 1.9 で抽出した結果についても同様に調べる.



(a) 全データ



(b) 2000 目まで拡大した



(c) 分位数の差

図 3.5: 類似度結果配列グラフ

表 3.2 において, しきい値 1.7 で抽出される効果語はしきい値 1.8 で抽出される効果語と比較すると 5 語の取りこぼしがある．その 5 語中 (高純度水素, 最大水素吸蔵量等)2 語は効果語と判断するので, しきい値 1.7 に指定できない．一方，しきい値 1.9 の効果語としきい値 1.8 の効果語には差分は生じず同じである．未知な特許文書に対し効果語を抽出する場合, 効果語をより多く抽出するためには, しきい値が大きい方が良いと判断し, 実験ではしきい値を 1.9 と設定する．

表 3.1: 分位数の結果と差

分位数目 (Q_n)	結果	差
Q_1	0.6758	0
Q_2	1.8266	1.1507
Q_3	2.3904	0.5638
...	...	...
Q_{99}	5.4709	0.0779
Q_{100}	5.5802	0.0093

表 3.2: しきい値 1.7 と 1.8 の時と 1.8 と 1.9 の時の効果語抽出差分

比較するしきい値	抽出した効果語の差分
1.8 - 1.7	高純度水素 最大水素吸蔵量等 製造加工中 前記複合材 水素吸収特性
1.8 - 1.9	なし

第4章
実験と結果

本章では実験と結果について述べる.

4.1 実験設定

実験は, 特許情報プラットフォームのマテリアルインフォマティクスに関連する合金分野から 144 件の特許文書を入力とし実験を行う. そのうち, 15 件は「発明の効果文」がないため除外する. 従って 129 件の特許文書を入力とする. 効果語の抽出精度の評価は先行研究の結果と比較し, Wikipedia 言語モデルと特許文書に特化した言語モデルの精度の比較を行う. 効果語の評価は著者の目視により行う.

4.2 実験結果

実験の結果を次に述べる. 129 件の特許文書中, 5 件がフィルタリングにより, 効果語候補を抽出できなかった. また, 30 件は類似度がしきい値以の結果となったため,

効果語を抽出できなかった.94 件から抽出した効果語の分布を図 4.1 に示す.

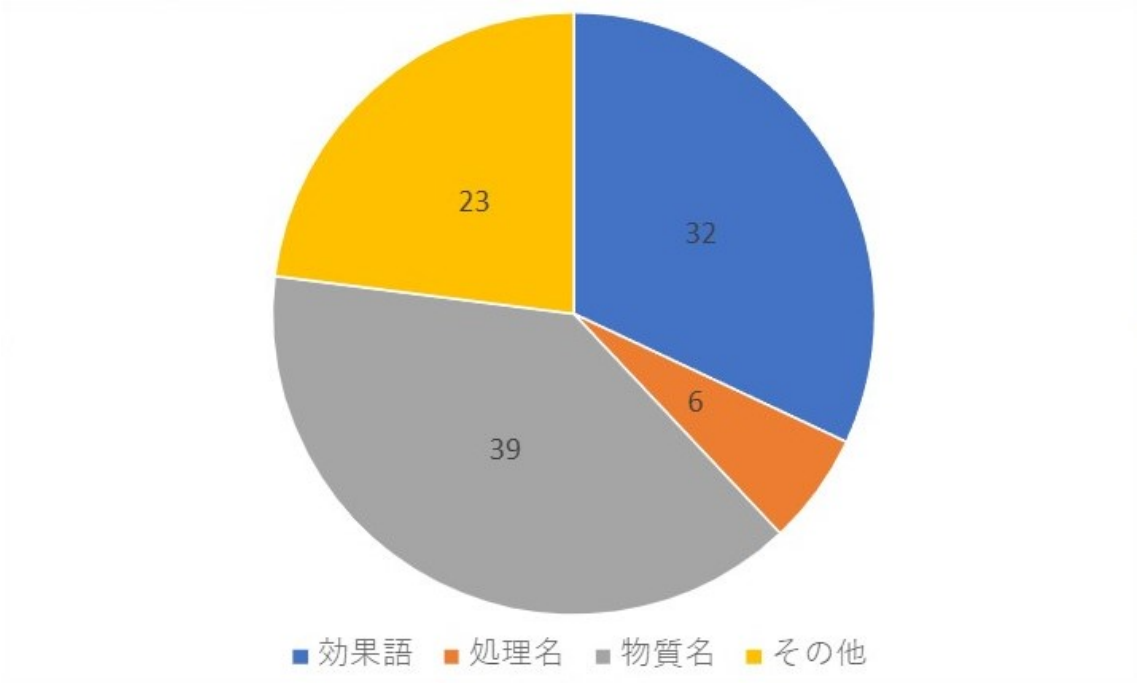


図 4.1: 抽出した効果語の分布 (単位:%)

この結果の分布は 4 種に分け, 「効果語」, 「物質名」, 「処理名」および「その他」とする.「その他」は前出の 3 種類に分類できない, 例えば, 装置を表す「前期容器」や構造の説明である「範囲内」または「表層部」のような語である.

「効果語」として抽出できた語は 32%, 「処理名」は 6%, 「物質名」は 39%, 「その他」は 23%であった.

表 4.1: 本手法で抽出した効果語の例

効果語	処理名	物質名	その他
高強度	切削加工	酸素ガス	表面層
冷却速度	機械加工	金属間化合物	合金表面
熱処理時間	微粉化	系水素吸蔵金属	範囲内
放電容量	初期活性化	次亜リン酸ナトリウム	晶構造
効果語	合金化	二次元形状記憶素材	金属膜

第5章
考察

本研究結果と先行研究結果の比較を図 5.1 に示す.

「効果語」は 28%を多く抽出することができた. また, 「処理名」は 51%削減できた. しかし, 「物質名」は 6%, 「その他」は 17%多く抽出してしまった. この結果より, 複合名詞抽出の手法は先行研究よりも複合名詞を多く抽出できたことを示している. また, フィルタリングにより, 「処理名」を削減することに貢献できたことを示している. しかし, 複合名詞を多く抽出できたことにより, 「物質名」と「その他」が増えてしまったと考える.

特許言語モデルと Wikipedia 言語モデルのパフォーマンスの違いを調べるために, 長岡技術科大学との共同研究の手法と本研究の結果の効果語の一致度を求める. 共同研究では, 効果語を抽出するために, 発明の効果文にある複合名詞を抽出し, それを発明の効果の効果語候補と定義し, 複合名詞が「請求項」から抽出した複合名詞が重複していれば, 効果語候補から削除し, 削除しない語は効果語とする手法である. 一致度の結果は表 5.1 に示す. 特許言語モデルは Wikipedia 言語モデルより, 効果語を多く抽出できた.

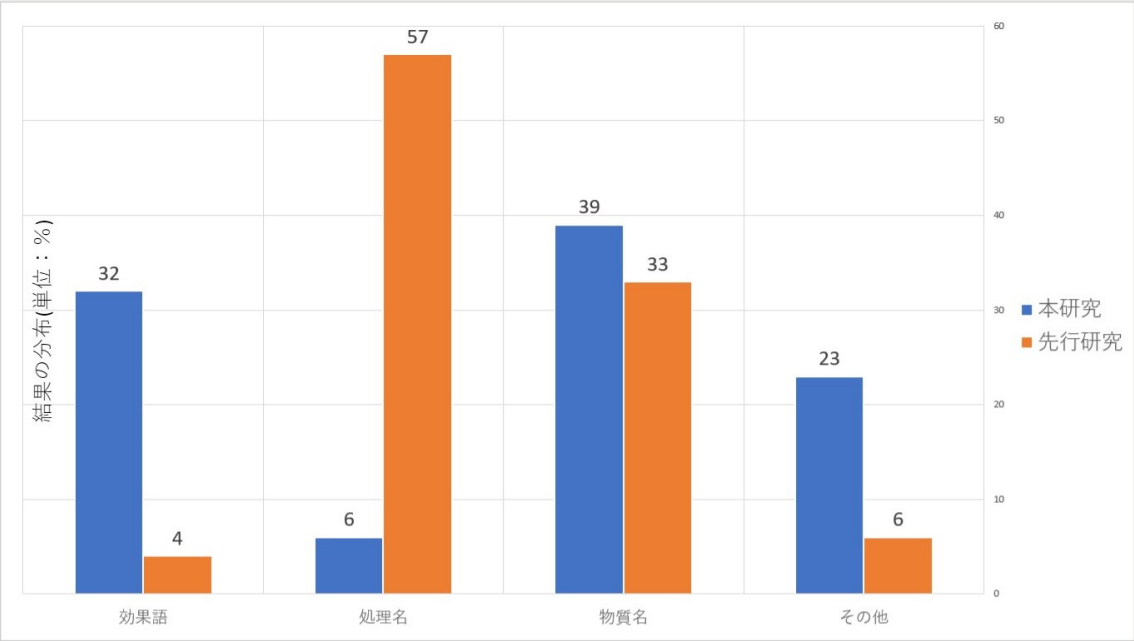


図 5.1: 本研究と先行研究の比較結果

表 5.1: 言語モデルの違いによる一致度の比較

言語モデル	**一致度** (単位:%)
特許言語モデル	56.4
Wikipedia 言語モデル	40.5

第6章
まとめ

論文や特許文書をはじめとする技術文献から各種実験パラメータや物性値に代表される数値データ（物理量＋数値）を抽出するための 効果語の抽出精度を向上するための研究を行った.

「処理名」を削除するためのフィルタリング処理により, 特許文書中の「処理名」を削除することができた. しかし, フィルタリングによる削除した文には効果語があるのに削除してしまうというケースがあった, そのため, 特許文書の 5 件から効果語候補が得られなかった. 改善するためには例外処理を行う必要がある.

また, 発明の効果文を完全に抽出できないというケースがあった. 効果語候補を抽出できない. そのため, 確実に発明の効果文を抽出するプログラムの改善必要がある.

本研究では, Wikipedia 言語モデルと特許言語モデルの利用を比較し, 特許言語モデルを利用することで効果語の抽出が向上できたということがわかった.

効果語の周辺に数値データが存在する.「物質名」や「その他」を削除するには, 実施例文に対し, 構文解析を行い, 「物質名」や「その他」という語が, 数値データを持っている文節が係り先を指していなければ, 結果から削除できると考える.

今後は, 特許文書から発明の効果文の抽出処理を改善し, 例外の処理を行う. また, マテリアルインフォマティクスに関連の特許文書をスクレイピングし, 特許言語モデルを再学習することで, 精度の向上が期待できる.

謝辞

本研究は, 筆者が釧路高専専攻科在学中の, 令和 3 年 4 月より令和 4 年 2 月までの１年間にわたり行ったものである. 本研究の遂行にあたり, 適切なる御指導, 御助言, 多大なるご援助を頂き, 常に有益な討論をして頂いた, 釧路高専情報工学科中島教授ならびに教育研究支援センター二谷技術専門職員に深く感謝し御礼申し上げます. 本研究を遂行にあたりご協力くださった, 長岡技術科学大学情報・経営システム工学野中准教授ならびに作本研級指導補助員に心より感謝申し上げます. また, 常に有益で適切な御討論をして頂き, 貴重な御意見を頂いた釧路高専情報工学科情報通信システム研究室所属の専攻科生並びに本科生の諸氏に厚く御礼申し上げます.

参考文献

[1] 作本猛, 野中尋史, 田中裕真, 立花龍弐, 坂地泰紀, 酒井浩之, 小林暁雄, “特許文書中の実験表現に関する属性定義と抽出方法の確立”, NLP 若手の会 (Yans) 第 14 回シンポジウム, 2019.

[2] Hirofumi Nonaka, Akio Kobayashi, Hiroki Sakaji, Yusuke Suzuki, Hiroyuki Sakai, Shigeru Masuyama, “Extraction of the Effect and the Technology Terms from a Patent Document”, Journal of Japan Industrial Management Associastion, Vol.63, pp.105-111, 2012.

[3] 大橋英一郎, 中島陽子,“特許文書からの教師なし数値データの抽出手法の開発”, 釧路高専卒業研究, 2020.

[4] Xin Gang Zhao , & Lijun Zhang, “JAMIP: an Artificial-Intelligence Aided Data-Driven Infrastructure for Computational Materials Informatics”, Science Bulletin, Vol 66, pp.1973-1985, 2021.

[5] P.Gorai, D.Gao, B.Ortiz, S.Miller, S. A.Barnett, T.Mason, Q. Lv, V.Stevanovic, and E. S. Toberer, “TE Design Lab: A Virtual Laboratory for Thermoelectric Material Design”, Vol 112, pp368-37, 2016.

[6] 布留川, 英一 (2021) 『BERT/GPT-3/DALL-E 自然言語処理・画像処理・音声処理人工知能プログラミング実習入門：最先端のフレームワークの実力を試そう！』東京: ポーンデジタル.

付 録A
コードの説明

本研究で用いたデータ, プログラム, 結果を格納するファイルについて説明する. また, これらは「FIN-PROG」フォルダに格納している.

1.1 メインプログラム

newPatentProg.py 特許文書インスタンスを生成するプログラム.

experiment.py 特許文書の実施例文から複合名詞を抽出するプログラム.

deletehousent.py 特許文書の発明の効果文に対しフィルタリングするプログラム.

hatsu_new.py フィルタリングした結果から複合名詞を抽出するプログラム.

inv_and_ban.py 発明の効果文から抽出した複合名詞とフィルタリングにより抽出した複合名詞を類似度を求め, しきい値以下であれば, 効果語候補から削除するプログラム.

findingshikiichi.py しきい値を求めるプログラム.

findDistance.py 類似度を求め, 効果語を抽出するプログラム.

1.2 関数

IGNORE_WORDS.py

ig(変数) ストップワードのリスト.

Matcher Spacy で一つの関数であり, 言葉のパターンを探すオブジェクトを返す関数.

Matcher_Inv フィルタリングに適用され, 言葉のパターンを探すオブジェクトを返す関数.

discard.py

discard_Katakana カタカナを削除する関数.

discard_Ascii Ascii 文字を削減する関数.

discard_word 要らない語を削除する関数.

discard_setsuzoku 接続詞を削除する関数.

extract.py

exct_experimental_section 特許文書から実施例文を抽出する関数.

exct_invention_section 特許文書から発明の効果文を抽出する関数.

delete_all_conclusion_word 「この発明により」または「本発明によれば」という文を削除する関数.

clear_the_method_word フィルタリング手法の関数

check_kakari_uke 文節の係先が「〜により」という文節であるかどうかを調べる関数.

checkMethodWord フィルタリングを対象になる文では「〜による〜」の後に文がまだあれば, その文を削除しない関数.

findWords.py

find_noun_and_compound 複合名詞を抽出する関数.

check_symbol シンボルをあるかどうかを調べる関数.

myloader.py

load_jsonfile json ファイルをロードする関数.

load_picklefile ブジェクトファイルをロード関数.

save_jsonfile 辞書変数を json ファイルとして保存する関数.

save_picklefile リストや辞書などをオブジェクトファイルとして保存する関数.

tools.py

make_list_into_string リストを String 型に変換する関数.

load_spacy 形態素解析 Spacy と ja_ginza 辞書を呼び, 形態素解析インスタンスを返す関数.

1.3　クラス

Patent.py 一つの特許文書クラスである.

フィールド

path ファイルまでの PATH.

name 特許文書のファイル名前.

doc 特許文書の内容.

doc_experiment 特許文書の実施例文.

doc_invention 特許文書の発明の効果文.

new_doc_inv_word フィルタリングした発明の効果文.

関数

load_file 特許文書ファイルを読込み関数.

load_experiment 特許文書の実施例文を抽出する関数.

load_invention 特許文書の発明の効果文を抽出する関数.

print_experiment 実施例文を表示させる関数.

print_invention 発明の効果文を表示させる関数.

print_doc 特許文書の内容を表示させる関数.

print_name 特許文書のファイル名前を表示させる関数.

具体的な説明やコードの実行流れは「note.txt」に書いてある.

1.4　データと生成されたファイル実行結果

effect_words フォルダであり, 特許文書を格納している.

patent.object 「effect_words」にある全ての特許文書インスタンスであり, リストとして格納している.

out-experiment.json 実施例文から抽出した複合名詞である.

out-experiment-filtered.json(*) 実施例文から抽出した結果より, 確実に物質名を削減した結果.

allvalue.lst フィルタリングからの複合名詞と発明の効果文の複合名詞の全ての類似度結果.

inv-filt-with-distance.json フィルタリングにある複合名詞が発明の効果文の効果語候補の結果から取り外した結果.

inv_filt_with_dist_after_discarding.json 「inv-filt-with-distance.json」結果から確実に物質名である語を削減した結果.

all_res_distance.lst 発明の効果文と実施例文の全ての類似度結果.

distance_T.json 発明の効果文と実施例文の類似度がしきい値以下の結果.

distance_T_hazure.json 発明の効果文と実施例文の類似度がしきい値以上の結果.

all_koukago_from_hatsumei.json 「distance_T.json」結果から発明の効果文からの語のまとめの結果.

all_koukago_from_jisshirei.json 「distance_T.json」結果から実施例文からの語のまとめの結果.

template.json 「effect_words」にある特許文書を辞書型した結果.

***.pd(*)** Pandas ファイルであり, 「inv-filt-with-distance.json」を求める際に, 生成されたファイル. Debug ようである.

(*) については無視できるファイルである. 実験では使用していない.