

# 卒業論文

## IP ランドスケープ支援のための 特許情報のベクトル化を用いた 共起語ネットワーク作成システム

Data Fusion through Web-GIS Visualization  
Using Open Data for Evidence-Based Policy Making

富山県立大学 工学部 情報システム工学科

2020032 平井遥斗

指導教員 奥原 浩之 教授

提出年月: 令和6年（2024）年2月



# 目次

図一覧	ii
表一覧	iii
記号一覧	iv
第1章 はじめに	1
§ 1.1 本研究の背景	1
§ 1.2 本研究の目的	2
§ 1.3 本論文の概要	3
第2章 知的財産戦略と特許情報	4
§ 2.1 知的財産戦略	4
§ 2.2 特許情報処理と活用	6
§ 2.3 テキストマイニングと自然言語処理	9
第3章 特許情報の可視化	12
§ 3.1 特許情報のベクトル化	12
§ 3.2 次元圧縮手法とクラスタリング手法	14
§ 3.3 単語間のつながりと共起語ネットワーク	16
第4章 提案手法	20
§ 4.1 Google Patents からのデータ収集の高速化と分類	20
§ 4.2 クラスターの解釈と共起語ネットワーク	22
§ 4.3 システム化と IPL(Intellectual Property Landscape) への活用	24
第5章 数値実験並びに考察	26
§ 5.1 数値実験の概要	26
§ 5.2 実験結果と考察	26
第6章 おわりに	27
謝辞	28
参考文献	29

# 図一覧

2.1	IP ランドスケープの概要 [7]	6
2.2	特許文章の一例	6
2.3		8
2.4	言葉ネットワーク [12]	8
3.1	Attention 機構	14
3.2	BERT	14
4.1	テキストデータのフォーマット	21
4.2	システムのフロントページ	21
4.3	ユーザー辞書のフォーマット (csv)	23
4.4	出力結果	23
4.5		25
4.6	提案システム	25

## 表一覽

# 記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
MultiHead Attention における単語ベクトルに $W^Q$ を掛けたもの	$Q$
MultiHead Attention における単語ベクトルに $W^K$ を掛けたもの	$K$
MultiHead Attention における単語ベクトルに $W^V$ を掛けたもの	$V$
MultiHead Attention における次元数	$v^T$
MultiHead Attention における訓練される重み行列	$W^O$
Positional Encoding における位置エンベディングの次元数	$i$
Positional Encoding における埋め込みベクトルの次元数	$d_{model}$
Siamese Network における埋め込み表現の次元	$n$
Siamese Network におけるラベルの数	$k$
UMAP における点 $x_i$ と $x_j$ のユークリッド距離	$r_{ij}$
UMAP における点 $x_i$ に対して, $k$ 近傍の集合	$K_i$
UMAP における点の疎密に対応するための変数	$\sigma_i$

## はじめに

### § 1.1 本研究の背景

近年、コロナウィルスの影響やグローバル化、インターネット技術やAI、IoT等のデジタル技術の進展、顧客のニーズの多様化や社会環境などの急速な変化などにより、経営環境は大きく変化している。さまざまな要素が絡みあうことにより、変化のスピードが速く、質・量の変化も大きい状況（Volatility）、将来何が起こるか予測できない状況（Uncertainty）、様々な要素が複雑にからみあっている状況（Complexity）、物事の因果関係があいまいになっている状況（Ambiguity）いわゆる VUCA な時代を迎えている。急激な変化と不確実性が高まる社会に対応し、持続的な発展を目指すには、様々な点に留意し対応を進めてく必要がある [1]。

ICTは、様々な用途に応用できる汎用技術（General Purpose Technology(GPT)）であり、経済成長の原動力となっている。近年は、IoTやビッグデータ、AIといった新たなICTの潮流が注目されている。これらは、さまざまなデータを収集・蓄積し、AIなどを活用して分析することで、現状把握や将来予測、課題解決など、さまざまな価値創出につながる。そのため、第4次産業革命とも呼ばれ、社会や経済に大きなインパクトを与えている。しかし、日本企業のIoT進展指標やIoT導入意向は、調査対象6か国の中で相対的に低い。その理由として、非ICT産業でのIoT化の具体的なイメージが浸透していないことや、人材育成が課題となっていることが示唆されている。IoTやビッグデータ、AIといったICTを活用することで、現状把握や将来予測、課題解決など、さまざまな価値創出が可能になる。また、人が通信の主役ではなく、機械間通信が中心となる次のフェーズでは、ICTの役割はさらに重要になるだろう。これら一連の変化が第4次産業革命であり、今後、これらの技術革新を通じて、日本産業の在り方を変革し、Society 5.0を世界に先駆けて実現することが期待されている。[2]。

そんな中、企業の持続的な発展を図るためには、自社の核となる独自の強みを生かし、他者との差別化を図ることが極めて重要である。具体的には、これまで蓄積してきたコア技術をさらに磨き、製品やサービスにおける優位性を確立する必要がある。長年の研究開発投資によって獲得した特許や、ノウハウ、人材といった無形の資産を最大限に活用することで、他者との差別化を明確に打ち出す戦略が求められる。また、単に技術的優位性だけでなく、事業領域の多角化などの経営戦略の面からのアプローチも欠かせない。市場環境の変化に対応し、革新性を担保するためには多角的な視点での経営が不可欠である.. 中長期的な成長性と短期的な収益のバランスをとりながら、競合となる企業の動向も見極めていく総合力が求められる [3]。

## § 1.2 本研究の目的

本研究の目的は、今日に至るまでに蓄積された莫大な量の特許情報を用いて過去から現在に至るまでの技術開発を可視化し、その中から新たな知見を抽出することである。

特許情報は、技術開発の成果を客観的に反映した貴重な情報源であり、技術動向の把握や競合他社の分析など様々な場面で活用されている。しかし、近年、技術開発の加速とグローバル化に伴い、世界的な特許出願件数が増加している。世界知的所有権機構（WIPO）によれば 2021 年の世界特許出願件数は 340 万件で、前年比 3.6% 増加し、2022 年の世界特許出願件数は前年比 1.7 増となり、2 年連続で過去最高を更新した。2010 年に 199.7 万件であった世界の特許出願件数は、この 10 年間で 1.61 倍に増加し、2019 年には 322.4 万件に達した。

このように莫大となった特許情報を人の手だけで読み込み、分析することはもはや困難を極めている。こうした課題に対処するため、特許情報の分析を支援する人工知能が近年注目を集めている。

また、特許庁の調査によれば、IP ランドスケープが必要であると回答した者は約 8 割であったのに対し、IP ランドスケープを十分に実施できていると回答したものは約 1 割であった [4]。現在、必要性は理解しているがまだ実施に至っている企業が少ないという状態である。



## § 1.3 本論文の概要

本論文は次のように構成される.

**第1章** 本研究の背景と目的について説明する.

**第2章**

**第3章**

**第4章**

**第5章**

**第6章** 本論文における前章までの内容をまとめつつ, 本研究で実現できたことと今後の展望について述べる.



# 知的財産戦略と特許情報

## § 2.1 知的財産戦略

知的財産戦略とは企業が保有する知的財産を経営戦略の一環として取り入れ、企業の競争力を高め、事業目標を達成することを目的とする戦略である。知的財産には、特許、商標、意匠、著作権、ノウハウなど、さまざまな種類があり、これらの知的財産をどのように活用すれば、企業の価値を最大化できるのかを考えることが重要である [5]。

知的財産戦略は、経営戦略と密接に関係しており、企業全体の戦略において各部門や機能の方向性を決定する重要な役割を果たしている。日本において、知的財産戦略は特許などの知的財産（Intellectual Property：IP）と景観や風景を意味する「Landscape」を組み合わせた造語で「IP ランドスケープ」と呼ばれることが多い。

知的財産戦略の目的は、以下の3つにまとめることができる [6]。

### （1）オープンイノベーション創出に貢献する知的財産戦略

- オープンイノベーションによる事業創出に貢献する知的財産戦略

オープンイノベーションによる事業創出とは、近年の変化が激しい事業環境下において、従来のような社内のみで行う研究開発では、新規事業の創出に限界があることを踏まえ、競合企業やスタートアップ、大学等などの外部からの技術やアイデアを自社に取り組みこと等を通じて新たな価値を創造し、事業を創出しようとするもの。

- プラットフォーム戦略の推薦による事業創出に貢献する知的財産戦略

プラットフォーム戦略の推進による事業創出とは、顧客や事業など、様々な主体を同一のプラットフォーム上に集めることで、事業のエコシステムを創出するビジネスモデルであるプラットフォーム戦略の推進により事業を創出しようとするものである。

- プソリューションビジネスの事業創出に貢献する知的財産戦略

ソリューションビジネスとは、従来のモノ売りのビジネスから脱却し、顧客の課題を解決するコト売りへと進化したビジネスである。すなわちソリューションを創出するビジネスである。従来は知財部門が顧客の課題解決に直接関与することは少なかったが、近年は知財部門が積極的に関与し、新たなソリューションのコアを早期に特定し、これを適切に保護する知財ポートフォリオを構築している企業が増えている。

### （2）事業競争力の強化に貢献する知的財産戦略

- コアインピーダンス強化に貢献する知的財産戦略

コアコンピタンスとは、競合他社との差別化につながる競争優位性をもたらす自社の強みであり、これを技術として支えるのがコア技術である。コアコンピタンスを現状からさらに磨き、深化させることは、競争優位性を維持・強化するために重要である。

- グローバル事業展開に貢献する知財財産戦略

グローバル事業展開の形態として、輸出、ライセンス、戦略的提携、買収及び現地子会社の新設等がある。

- M&A による事業ポートフォリオの拡大に貢献する知的財産戦略

M&A による事業ポートフォリオ拡大とは、社外に存在する事業を M&A を実施して買収することで、自社の事業ポートフォリオを拡大することである。M&A は、既存の事業の規模拡大の経済効果や、新規事業への参入新たな技術やノウハウの獲得など、様々な目的で実施される。

### (3) 組織・基盤の強化に貢献する知的財産戦略

- ブランド価値向上に貢献する知的財産戦略

ブランド価値の向上は、顧客からの信頼や好感を高め、他社に対しての競争優位性を構築するだけでなく、資金調達や人事確保の容易化など、企業の組織・基盤の強化にもつながる。ブランド価値は、高い経営理念に基づいた企業活動によって向上させることができる。

- デジタルトランスフォーメーション（DX）等による事業基盤の強化に貢献する知的財産戦略

デジタルトランスフォーメーションによる事業基盤の強化とは、IT やデータ等のデジタル技術を活用して、自社の事業基盤の強化を図るものである。近年、知財情報等を自社の事業基盤を強化するために利用する取り組みが注目を集めており、DX において、知財部門が貢献できることは少なくない。

- SDGs への貢献に関わる知的財産戦略 SDGs（持続可能な開発目標）の取り組みは、国際社会から企業への信頼を高め、グローバルな投資家から高い評価を得るために重要である。また、企業の持続的発展のためにも欠かせないものとなりつつある。

IP ランドスケープでは、自社の経営・事業戦略を決める際に、経営・事業情報に知財情報を取り込んだ分析を実施する。その結果を経営者・事業責任者と共有し、結果に対するフィードバックを受けたり、立案検討のための議論や協議などを行う。

## 経営戦略

経営戦略とは、会社が中長期的な目標を達成するために、資源の配分や事業内容の選択など、経営上の意思決定に関する気泡的な方針である。経営戦略には、企業としての成長戦略や収益力強化戦略、事業ポートフォリオの見直しなどが含まれる。

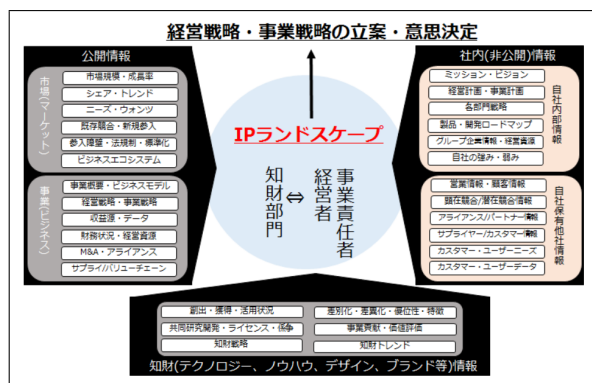


図 2.1: IP ランドスケープの概要 [7]



図 2.2: 特許文章の一例

効果的な経営戦略を立案するためには、自社の強みや特長を理解し、それを活かすことが重要である。また、事業環境の変化や競合他社の対応などを分析し、それに対応した戦略を練る必要がある。ステークホルダーの要望も踏まえた上で、最適な資源配分と事業の選択を行うことが大切である。

経営戦略の立案と実行は経営における最も重要な意思決定プロセスである。環境変化に対応しながら、戦略を実行することで、会社の目標達成と持続的成長を可能となる。このため、経営戦略には、トップの高いコミットメントが不可欠であると言える。

## 経営戦略のフレームワーク

経営戦略の策定では自社を取り巻く外部の環境要因に打ついて分析する外部環境分析や、自社内の環境を分析する内部環境分析を踏まえ自社の強みや弱み、機械や脅威を把握することで、戦略オプションを立案して最適な戦略を選択することが大切である。それらを行うために役立つ代表的なフレームワークとして、PEST 分析ファイブフォース分析、3C 分析、VRIO 分析、SWOT 分析、STP 分析、4P 分析などが挙げられる [8]。

内閣府や特許庁による IP ランドスケープの積極的な推進に代表されるように、IP ランドスケープは研究機関においても積極的に検討されるべき対象であると考えられる。また、その具体的な取り組みの多くに ICT を活用した取り組みが多数行われていることから、IP ランドスケープの効率的な実施には ICT の活用が不可欠であり、情報工学との親和性が高いものと思われる。これらのことから、本研究は情報技術を用いた IP ランドスケープの支援を目的とする。

## § 2.2 特許情報処理と活用

特許情報とは、特許・実用新案・意匠・商標の出願や権利化に伴って生み出される情報である。この情報は、研究開発の重複防止、既存技術の活用、無用な紛争の回避などに役立つ。特許情報は、研究開発の策定から商品化、更には他人の権利調査に至るまでの様々な事業活動において活用されている。

## 特許の一例

本研究で使用する特許の一例を図 2.2 に示す。このように特許はタイトルと要約である Abstract，国際特許分類（International Patent Classification）という特許分類を示す Classification，本文を示す Description，請求項を示す Claims，そして特許自体の ID と出願日などの情報を含んだ部分からなる。

具体的な活用例は，以下のとおりである。[10].

## 特許情報の活用例

- 技術動向調査

将来性を見据えた研究テーマの選定や過去になされた研究との重複回避のために，特許情報を利用して技術動向調査が行われる。特定の技術分野における特許出願の動向や出願件数の推移を調査することにより，過去にどのような技術が存在したか，また，今後開発すべき技術分野の把握の参考になる。

- 出願前の先行技術調査

研究成果として発明がなされたとき，権利化するか否かの判断が必要となる。特許出願をする際に関連する分野の先行技術について調査することにより，権利として認められる見込みのない無駄な出願を未然に防止することができる。

- 権利調査

開発製品が他人の産業財産権を侵害すると，製造・販売の中止や製造品の廃棄，あるいは権利者への損害賠償にまで発展する恐れがある。これらを未然に防止するために，設計から製造前段階にかけて，他人の権利範囲の調査を行う。

- 公知例調査

他の権利者から警告を受けた場合などの対抗手段として，自社の発明・考案を事業化する際に障害となる他人の特許権・実用新案権を無効にするため，その特許・実用新案登録の出願前の公知例を調査する。

- 公知例調査

事業を営む上で多くの場合には競合他社が存在している。その競合他社がどのような戦略で事業を行っているか調査する上で，特許情報は貴重な情報源となる。競合他社の過去から現在に至るまでの出願動向を把握することにより，研究開発動向等を読み取ることが可能である。また，競合他社の出願動向を継続的に監視し，自社にとって障害となる出願等の早期発見に努めることも重要である。

## 特許番号

特許番号とは，特許として認められた発明に付与される 7 桁の番号である。特許番号は「特許代 XXXXXXXX 号」のように表記されている。特許番号は，原則として，出願，公開，登録の審査段階それぞれで付与され，審査段階，年，通し番号で構成されている。この特

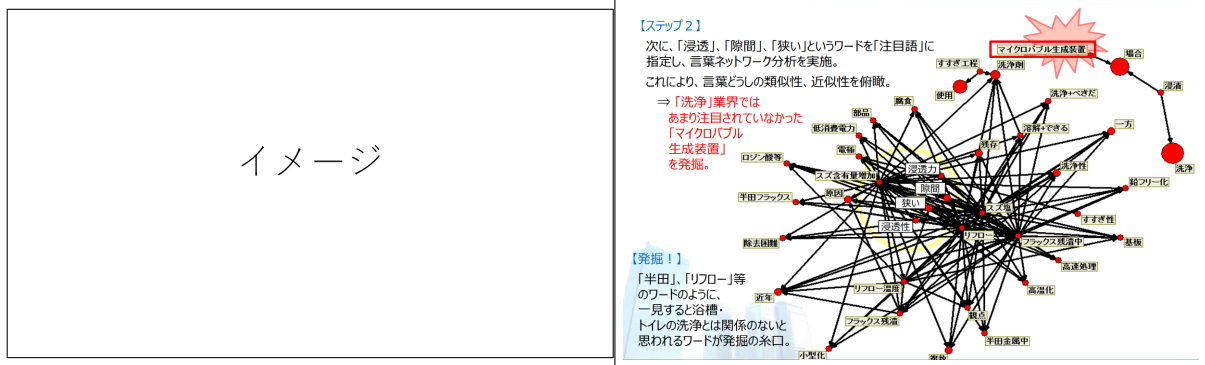


図 2.3:

図 2.4: 言葉ネットワーク [12]

許番号は発明の特定性と公示性の観点から非常に重要なものであり、特許権の貴族や特許文献検索の際の必須情報となっている。また特許関連書類の請求や特許料金の納付の際にもこの番号を提示する必要がある。特許番号は、その発明を社会に対して公表し保護しているという証であり、特許における根幹として機能している。

## 特許情報処理

特許情報処理とは、特許文章や特許データベースの情報をコンピュータで自動的に解析・処理を行う技術のことである。その目的は、莫大な量の特許情報から有用な知見を効率的に見出し、知的財産戦略の立案や技術動向の分析などに活用する。具体的には、自然言語処理やテキストマイニング、情報検索、データマイニングといった手法を用いて、特許文章の意味内容を理解したうえで重要なキーワードを抽出したり、文章間の関連性を判断したりする。また、抽出したデータを可視化し特許マップを作成することも一般的である。特許情報処理の利用目的としては、特許可否判断の支援、先行技術調査の効率化、技術開発のトレンド分析、競合他社の特許戦略の把握などがあげられ、特許業務の生産性向上や質の向上に役立つテクノロジーといえる。以下に実際の活用事例を紹介する [12]。

## IPL への活用事例

< 浴室・トイレの室内洗浄技術の課題を解決する技術を探る >

**ステップ1** 洗浄に関する特許出願の【発明が解決しようとする課題】に記載された文章をテキストマイニングし、係り受け関係のある言葉の頻度をランキング。これによれば、頻度が高く修飾/被修飾関係にある言葉のペアは、「効率-良い」洗浄、「狭い-隙間」部分の洗浄等が、「洗浄」に関する多くの出願が解決使用とする課題であることがわかる。このことから、「洗浄」に関する業界では、安全性が高く、狭い隙間にも浸透する洗浄力の高い洗浄剤・洗浄方法が模索されていることがわかる。

**ステップ2** 次に、「浸透」、「隙間」、「狭い」というワードを指定し、言葉ネットワーク分析を実施。これにより、言葉同士の類似性、近似性を俯瞰。実際の言葉ネットワークを図 2.4 に示す。このことから「洗浄」業界ではあまり注目されていなかった「マイクロバブル生成装置」を発掘

**ステップ3** 特許出願の【発明が解決しようとする課題】に記載されたワード「浸透力」に着目し、「浸透」、「狭い」、「隙間」等のワードを含む発明を抽出したところ、25件がヒット。

**ステップ4** 数十ナノメートルという極めて小さな気泡は、ウルトラファインバブル(UFB)と呼ばれる。UFBは、透明で視認できないことに加え、その気泡が極めて長期間(数ヶ月)液中に存在しうることや、気泡が電荷を帯びること、気泡内部が超高压状態になること等の特異な特性がある。産業界では、その特性を利用したUFBの応用が幅広い分野で検討されている。たとえば、食品分野をはじめとして、化粧品、薬品、医療、半導体や植物育成等、幅広い分野での応用がさかんに考えられており、ウルトラファインバブルに大きな期待。

## § 2.3 テキストマイニングと自然言語処理

テキストマイニングとは、定型化されていない文章から情報を抽出する技術です。SNSやアンケート、コールセンターの対応など、さまざまな場面で活用されている。テキストマイニングは、AI技術の進展により、より高度な分析が可能になった。また、テキストデータの量も増加しており、テキストマイニングツールの種類も増えている。[11]

テキストマイニングを行うことで、企業は、顧客のニーズや市場動向を把握したり、新製品の開発やマーケティングの戦略を策定したりすることができる。

このことによって、単一のデータの可視化のみでは表面化してこなかった課題をくみ取ることや、逆に、課題に対する解決策を一見関係のなさそうな分野から発見するといったことが可能となる。

テキストマイニングにはBeautifulSoupとSeleniumを用いる。Webサイトをスクレイピングする際にはrequestsとBeautifulSoupを用いることが一般的だが、今回用いるGooglePatentsの検索結果を取得するにはrequestの処理を用いることができずChromeDriverを用いている。それらの特許の本文をスクレイピングする際はBeautifulSoupを用いる。この際BeautifulSoupを用いるのは、一般にSeleniumよりBeautifulSoupの方が高速に処理できるためである。

### BeautifulSoup

BeautifulSoup4とは、Webサイト上のHTMLから、必要なデータを抽出するためのPythonのライブラリである。Beautifulsoup4でスクレイピングする際、最初に対象のWebページからHTMLを取得する必要がある。HTMLを取得する方法として、同じくPythonのライブラリである、Requestsのget関数や、Seleniumのpage source関数を使うなどの方法がある。上記の方法によって取得されたHTMLテキストを、BeautifulSoup4のBeautifulSoup関数に渡すことで、BeautifulSoupオブジェクトを作成することができる。また、そのオブジェクトからclassを検索することでWebサイトの必要な情報を抽出する。classを検索するときに、条件を満たすひとつの要素を取得するselect one関数や、条件に合う条件のすべてを取得するselect関数、find関数などがある。selectとfindの違いは引数を指定する条件の指定方法がある。前者は、CSSセレクタを指定して要素を取得し、後者は



class 名や属性キーワードを指定して検索し、class を取得する。これらの関数から取得した Tag オブジェクトである要素から、内部テキストのみを取得するためには、get text 関数を使用することで取得することができる。

## Selenium

Selenium は、Web アプリケーションのテストや Web スクレイピングを行うための python モジュールである。Selenium を用いると、自動的に実際のブラウザを操作・制御することができる。主な機能としては、Chrome や Firefox など様々なブラウザの起動、ページの移動、テキストや画像の取得、フォームへの入力、JavaScript の実行結果の取得などがあげられる。また、複数のタブやウィンドウの制御も可能である。Selenium を使うには、通常は WebDriver と呼ばれるブラウザ制御ライブラリと組み合わせて利用する。WebDriver によって特定のブラウザを自動操作するためのインターフェースが提供される。本研究では ChromeDriver を用いる。Selenium の大きなメリットはブラウザそのものを自動でテストできる点であり、実際のユーザーに近い操作をプログラム上で実現することができる。

## 分かち書き

自然言語処理（NLP）において、分かち書きは、テキストを単語や句などの意味的な単位に分割する処理である。分かち書きは、テキストの意味理解や解析の基礎となる重要な処理であり、多くの NLP タスクで必要となる。

分かち書きの目的は、テキストの意味を正確に理解するために、テキストを単語や句などの意味的な単位に分割することである。例えば、文の意味を理解するためには、文を主語、述語、目的語などの句に分割する必要がある。また、単語の意味を理解するためには、単語を品詞や語義などの単位に分割する必要がある。

分かち書きの処理方法は、大きく分けて以下の 2 つに分けられる [9]。

### 1. ルールベース型

ルールベース型分かち書きは、あらかじめ定義された対象となる言語の文法ルールに基づいて分かち書きを行う方法である。ルールベース型分かち書きは、人手でルールを定義するため、単純な分かち書きを行う場合は比較的容易に実装でき、調整も可能であるが、複雑な分かち書きを行う場合は、ルールを複雑にする必要があり、高度な専門知識が必要となり、誤りが生じやすくなる。

### 2. 統計学習ベース型

統計学習ベース型分かち書きは、機械学習によって導き出されたルールに基づいて分かち書きを行う方法である。統計学習ベース型分かち書きは、複雑な分かち書きを行う場合でも、比較的正確に分かち書きを行うことができる。また、機械学習に大量のテキストデータが必要であり、計算量が大きいという問題もコンピュータの高速化と低価格化により解決に向かっている。

分かち書きの精度は、分かち書きの目的や、分かち書きを行うテキストの種類によって異なる。例えば、新聞記事などのフォーマルなテキストであれば、ルールベース型分かち書きでも比較的高い精度で分かち書きを行うことができる。一方、SNS の投稿などの非フォー

マルなテキストであれば、統計学習ベース分かち書きの方が高い精度で分かち書きを行うことができる。

近年、NLP 技術の進展により、分かち書きの精度も向上している。また、クラウドサービスやオープンソースソフトウェアの普及により、分かち書きの利用が容易になってきている。

また、現状の分かち書きには、以下の課題がある。

- **日本語の曖昧さ**

ルールベース型分かち書きは、あらかじめ定義されたルールに基づいて分かち書きを行う方法である。ルールベース型分かち書きは、単純日本語は、英語と比べて曖昧な表現が多い言語である。例えば、「私は、彼に会いました。」という文は、文法的には「私は、彼に会いに行きました。」という意味にも解釈できる。このような曖昧な表現を正確に分かち書きすることは、困難である。

- **新語や流行語**

常に新しい言葉や表現が生まれてくるため、分かち書きのルールや統計モデルを常に更新する必要がある。

- **誤ったデータの影響**

分かち書きの精度は、分かち書きの対象となるデータの品質に大きく影響を受ける。誤ったデータが含まれていると、分かち書きの精度が低下する。

今回用いる特許の本文には、専門的な用語や複合語が多数含まれているため、それらを正しく抽出する必要がある。そのため、python のモジュール termextract を用いて専門用語や複合語の抽出を行い、それらを分かち書きの辞書に登録する。

## **termextract**

termextract は、東京大学情報理工学系の松本研究室によって開発されたテキストデータから専門用語を抽出するための Python モジュールである。termextract は、テキスト中からキーワードや専門用語を自動抽出するためのモジュールであり、これにより、文章や Web ページなど、非構造テキストから重要な用語を取り出すことができる。

termextract には、頻出単語解析 (TF-IDF)、C 値検出、longest-match 法など、いくつかのアルゴリズムが実装されている。任意のテキストを入力として、単語や語句の出現頻度、文脈に基づき、キーワードを定量的・定性的に抽出する。

また、事前に用意された専門分野の用語辞書を活用でき、ドメイン固有の用語抽出も可能である。生物医学系であれば遺伝子名やタンパク質名など特有の専門用語を、一般用語と区別して抽出できる。

今後、テキストマイニングと自然言語処理は、AI や ML 技術を活用することで、より高度な分析が可能となり、より幅広い分野で活用されるようになって考えられる。また、テキストデータの量の増加に対応するため、テキストマイニングと自然言語処理の自動化や、テキストデータの検索・分析・活用を効率化する技術の開発が進んでいくと考えられる。



## 特許情報の可視化

### § 3.1 特許情報のベクトル化

特許情報は、日々蓄積され、今では莫大な量となっており、それらの分析は困難を極める。そこで、特許情報を効率的に分析するためには、各特許をベクトル化して整理をおこない、全体を俯瞰できるように可視化する必要があると考える。本研究では、特許本文の文章を対象にベクトル化を行う。特許本文には、特許技術の内容が詳細に記載されているため、これらの情報をベクトル化することで、特許の技術分野や技術トレンドなどを把握することができると思う。

具体的には、特許本文を Sentence-Bidirectional Encoder Representation from Transform (Sentence-BERT) を用いることで文章全体を単位にベクトル化を行う [14]。Sentence-BERT は、Bidirectional Encoder Representations from Transformers (BERT) をベースに開発されており、文章の単語の順序を考慮して、文章の意味を表現するベクトルを生成する。

sentence-BERT は、文章の意味を理解する能力に優れているため、自然言語処理の様々なタスクに活用されている。

#### Transformer

近年、翻訳などの入力文章を別の文章で出力するというモデルは、Attention を用いたエンコーダー、デコーダ形式の RNN や CNN が主流であった。しかし、Transformer は、RNN や CNN を用いず Attention のみを用いたモデルである。Transformer は、再帰も畳み込みも一切行わないので並列化が容易であり、他のタスクにも汎用性が高いという特徴がある。Transformer においては Attention を多数並列に配置した Multi-Head Attention が用いられ、一般的に以下の式 (3.2) のように定式化される [13]。

<Multi-Head Attention>

$$MultiHeadAttention(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W_o \quad (3.1)$$

$$\text{where } head_i = ScaledDotProductAttention(QW_i^Q, KW_i^K, VW_i^V) \quad (3.2)$$

ここで、Scaled Dot-Product Attention では、内積を利用したベクトル間の類似性に基づく変換を行う。

<Scaled Dot-Product Attention>

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3.3)$$

3.2では、学習パラメータを持っていない Scaled Dot-Product Attention の表現力を広げるために、入力直前に学習パラメータを持つ Linear 層の追加を行っている。これにより、入力されるベクトルの特徴空間に依存しない注意表現を学習することができる。Linear 層の追加を行った Scaled Dot-Product Attention を一般に Single-Head Attention を呼ぶ。

Scale Dot-Product Attention は、ある単語に対して、その単語が文章に含まれる単語とどれだけ類似しているのかを計算し、それらを確率的に表現したものである。Transformer における Attention の入力には主に以下の 2 種類の入力方法が用いられている。

1. Self-Attention (softmax に与える Query, Key, Value を同じ値にする)
2. Source-Target-Attention (Key, Value を同じ値にし、Query を異なる値にする)

Single-Head Attention では多種多様な意味や文法をもつ単語に対しても単一の注意表現が生成される。そこで、Single-Head Attention を多数並列に配置して Multi-Head にすることで、複数の特徴部分空間における注意表現の獲得をすることができる。

以上のことから、文章を行列で表せることが分かった。しかし、文章というのは、文字を読む方向が重要であり、行列として表され、かつ、一括で処理する場合、文字の順番の概念がなくなってしまう。これは、文章を正しく扱えなくなる可能性がある。そのため、Embedding 層からの行列に位置情報を含んだ行列を足し合わせることで、文字の順番の概念を扱えるようにする必要がある。これを可能にするのが Positional Encoding である。

<Positional Encoding>

$$PE_{(pos, 2i)} = \sin \left( \frac{pos}{10000^{\frac{2i}{d_{model}}}} \right) \quad (3.4)$$

$$PE_{(pos, 2i+1)} = \cos \left( \frac{pos}{10000^{\frac{2i}{d_{model}}}} \right) \quad (3.5)$$

入力文章の単語数が 50 個まで扱えて、Embedding 層の埋め込み次元数が 128 次元の場合、Positional Encoding が生成する行列は 128 次元の行ベクトルが縦に 50 個並んだ行列になる。この行列は、各行のベクトルが絶対に同じものにならないため、この行列から単語の位置情報を表すことができる。具体的には、行ベクトルの各次元は、単語の位置情報に応じて、異なる値が割り当てられている。例えば、最初の行ベクトルの最初の次元は、単語の位置が 0 であることを示し、最後の行のベクトルは、単語の位置が 49 であることを示す。このように、Positional Encoding は、単語の位置情報を行ベクトルに埋め込むことで、Transformer モデルが単語の順序情報を利用できるようにしている。

## BERT

BERT は、Google が提案した最新の言語モデルの一つであり、BERT のキーポイントは、Encoder のみの Transformer アーキテクチャを採用し、Attention メカニズムを用いて単語間の関係性をモデル化していることである。BERT の目的は、あるテキストから単語や語句の意味表現を文章全体の文脈に依存したリッチなベクトル表現で表現することである。こ

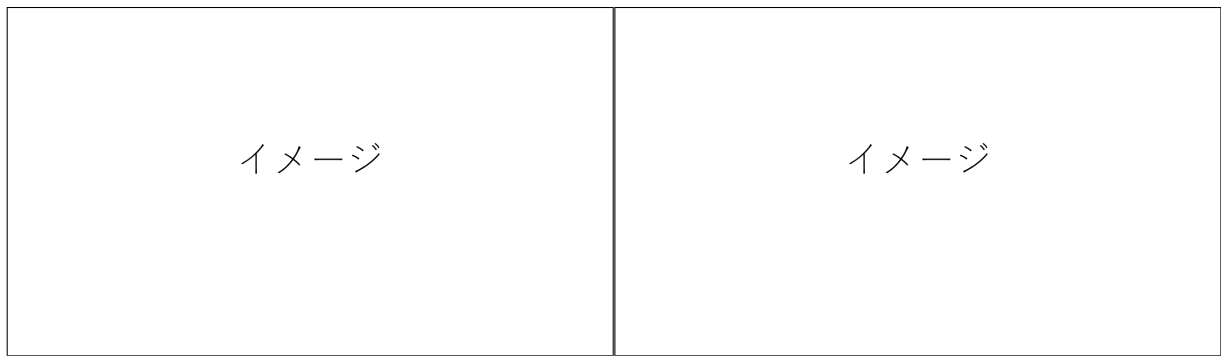


図 3.1: Attention 機構

図 3.2: BERT

これは下流のテキスト分類や質問応答などに有用な汎用的な言語表現を獲得できることを意味する。BERT は Masked LM と次文予測の 2 つのタスクで事前学習を行うことで統計的な言語モデルを構築している。Masked LM ではランダムにマスクした単語を予測することで文章理解能力を高め、次文予測では文章間の関係性を学習している。この事前学習済みモデル BERT モデルは、下流タスクの比較的小さいデータセットで微調整することで転移学習が可能である。結果として、多くの NLP タスクで従来手法を上回る精度を達成しており、BERT は分散表現と転移学習において大きな進展をもたらした。

### Sentence-BERT

BERT では 2 つの文章を入力し、それらの類似度を測ることができる。しかし、複数の文章を入力する場合は BERT では容易ではない。そこで本研究では Sentence-BERT を用いる。BERT で求められた埋め込み表現を pooling し、それらを Softmax 関数を用いて、分類を行う。

<Siamese Network>

$$O = \text{softmax}(W_t(u, v, |u - v|)) \quad W_t \in R^{3n \times k} \quad (3.6)$$

事前学習モデルは Hugging Face や GitHub などのサイト公開されている。また東京大学や京都大学なども独自のモデルを公開している。本研究では Hugging Face に登録されている ”sonoisa/sentence-bert-base-ja-mean-tokens” を用いる。

## § 3.2 次元圧縮手法とクラスタリング手法

今回扱うデータは 768 次元と高次元であるためクラスタリングを行う際に次元の呪いが発生することが考えられるため、次元圧縮を行う。次元圧縮手法には線形次元圧縮手法と、非線形圧縮手法がある。線形次元圧縮手法は、計算がよいであるが、データの非線形的な構造を表現することが難しい。一方で、非線形次元圧縮手法は、データの非線形的な構造を表現することができるが、計算が複雑で、処理に時間がかかる。今回行う次元圧縮では、ベクトル同士の近さを保持する必要がある。ベクトル同士の近さを保持するためには、非線形次元圧縮を用いる必要がある。そこで本研究での次元圧縮手法には Uniform Manifold Approximation and Projection of Dimension Reduction (UMAP) を用いる。

## UMAP

UMAP は 2018 年に, Leland McInnes, John Hecht, James Melville によって提案された手法である [15]. 従来の非線形次元圧縮手法である t-SNE よりも実行時間が高速であり, 圧縮後の情報保持力が高いという特徴がある. また, 4 次元以上にも圧縮することが可能である. UMAP は高次元空間での近いデータ同士を低次元空間でも近く, 異なる点同士を遠ざけるという処理を行うこと, で高次元でのデータ同士の近さや遠さを低次元でも表現できるようにしている.

高次元空間におけるデータ同士の近さは以下のように定義され, 3.9 によって対称化される [16].

< 重み付き k 近傍 (高次元) >

$$v_{j|i} = \exp\left(\frac{-(r_{ij} - \rho_i)}{\sigma}\right) \quad (3.7)$$

$$\rho_i = \min_{j \in K} \{r_{ij}\} \quad (3.8)$$

< 対称化 >

$$v_{ij} = (v_{j|i} + v_{i|j}) - v_{j|i}v_{i|j} \quad (3.9)$$

低次元空間におけるデータ同士の近さは以下のように定義され, 3.11 によって対称化される.

< 重み付き k 近傍 (低次元) >

$$w_{ij} = \exp(-\max\{0, d_{ij} - \rho'\}) = \tilde{w}_{ij} \quad (3.10)$$

< 対称化 >

$$w_{ij} = \frac{1}{1 + a \cdot d_{ij}^{2 \cdot b}} \quad (3.11)$$

トポロジカル表現の最適化によって高次元空間での近さと低次元空間での近さができるだけ同じになるように最適化を行う.

< トポロジカル表現の最適化 >

$$L = \sum \left[ v_{ij} \log \frac{v_{ij}}{w_{ij}} + (1 - v_{ij}) \log \frac{1 - v_{ij}}{1 - w_{ij}} \right] \quad (3.12)$$

## K-means

次元圧縮を行ったデータに対してクラスタリングを行い, 潜在的なグループ化を行う. クラスタの重心を求める必要があるためクラスタリングには k-means を用いる [?].

$n$  個の個体を  $\vec{x}_i = (x_{i1}, \dots, x_{iD})$ ,  $i = 1, \dots, n$  で表し, この個体の集合を  $X$  とする. データを  $K$  個の重なるの無いクラス  $X_k$ ,  $k = 1, \dots, K$  に分類するため, 次の目的関数を用いる.

$$J = \min_{\{\vec{c}_k, k=1, \dots, K\}} \sum_{i=1}^n \sum_{\vec{x} \in X_k} \|\vec{x}_i - \vec{c}_k\|^2 \quad (3.13)$$

ここで、 $\|\vec{x}_i - \vec{c}_k\|^2 = \sum_{d=1}^D (x_{id} - c_{kd})^2$  とし、クラスター中心は  $\vec{c}_k = (c_{k1}, \dots, c_{kD})$  である。式について最小化を行うため、K-Menas アルゴリズムと呼ばれる以下の反復アルゴリズムを使用する。

1.  $\vec{c}_k^{(t)}$  が与えられたとき、それぞれの  $\vec{x}_i$  に関して次式を計算する。

$$a = \operatorname{argmin}_k \|\vec{x}_i - \vec{c}_k^{(t)}\|^2 \quad (3.14)$$

2.  $X_k^{(t)}$  が与えられたとき、次式を計算する。

$$\vec{c}_k^{t+1} = \frac{1}{n_k^{(t)}} \sum_{\vec{x}_i \in X_k^{(t)}} \vec{x}_i, k = 1, \dots, K \quad (3.15)$$

ここで、 $n_k^{(t)}$  は  $X_k^{(t)}$  に属する個体の数であり、 $\vec{c}_k^{t+1}$  は  $X_k^{(t)}$  の  $(k+1)$  回目のクラスター中心でありクラスターの代表点に相当する。

ある小さい定数を  $\epsilon$  とし、すべての  $k$  について終了条件である  $\|\vec{c}_k^{t+1} - \vec{c}_k^t\| < \epsilon$  を満たすまで、(1) と (2) の手順を繰り返す。

## シルエット分析

K-means を行うには初めにクラスター数を与える必要があるため、シルエット分析を用いて最適なクラスター数を決定する。シルエット分析はクラスター内は密に凝集されているほど良い。異なるクラスターは花らているほど良い。この二つをもとに最適なクラスター数を求める。各データサンプル  $x^{(i)}$  に関して、以下のようにシルエット係数を求める [18]。

< 凝縮度 >

$$a^{(i)} = \frac{1}{|C_{in} - 1|} \sum_{x^{(j)} \in C_{in}} \|x^{(i)} - x^{(j)}\| \quad (3.16)$$

< 乖離度 >

$$b^{(i)} = \frac{1}{|C_{near}|} \sum_{x^{(j)} \in C_{near}} \|x^{(i)} - x^{(j)}\| \quad (3.17)$$

< シルエット係数 >

$$s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max(a^{(i)}, b^{(i)})} \quad (3.18)$$

## § 3.3 単語間のつながりと共起語ネットワーク

関連性の高い単語は、一緒に出現することが多いため、それらの単語の共起関係を調べることで、単語間の関係性を理解することができる。共起分析では単語同士の Simpson 係数という指標を用いて単語同士の共起度合いを比較し、共起関係にある単語と単語を線で結



んで描かれる共起語ネットワークが利用される．このような共起語の分析を通じて，単語同士の意味的な特徴を理解することができる．本研究では，各クラスター内の単語にどのような関係があるのかを理解することを目的とする．共起関係を分析するには主に Jaccard 係数，Dice 係数，Simpson 係数が用いられる [19]．

## Jaccard 係数

ある集合  $A$  とある集合  $B$  について Jaccard 係数  $J(A, B)$  は，以下の式で定義される．

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.19)$$

Jaccard 係数は 2 つのデータセット間の類似度を測る手法である．2 つの集合に含まれる要素のうち共通要素が占める割合を表しており，完全に一致するときに 1，共通する要素がないときに 0 となり，係数は 0 から 1 の間の値となる．テキストマイニングにおいては，文章と文章の類似度を表す指標となる．具体的には 2 つの語少なくともどちらかが含まれる文章を数えて，2 つの語両方が含まれる文章の割合を計算する．割合が大きければ，2 つの語は今回のデータセットの中において「近い」と判断することができる．この Jaccard 係数が大きいほど 2 つの集合の類似度は高いといえる．

### <Jaccard 係数の欠点>

Jaccard 係数では分母に 2 つの集合の和集合を採用することで値を標準化し，他の集合同士の類似度に対する絶対評価を可能にしている．しかし，Jaccard 係数は 2 つの集合の差集合の要素数に大きく依存するため，差集合の要素数が多いほど Jaccard 係数は小さくなる．これは，人の目から判断した際の「共通要素が多いほど類似度が高い」という感覚と異なっている．

そこで，差集合の要素数の影響を抑え，共通要素の要素数の影響に重みをおく Dice 係数が提案された．

## Dice 係数

ある集合  $A$  とある集合  $B$  について Dice 係数  $DSC(A, B)$  は，以下の式で定義される．

$$DSC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (3.20)$$

Dice 定数の定義式は，Jaccard 係数の定義式の分母  $|A| \cup |B|$  を  $(|A| + |B|)/2$  と変換することで得られる．よって Dice 係数は 2 つの集合の平均要素数と共通要素数の割合を表しており，Jaccard 係数と同様に 0 から 1 の間の値となることがわかる．また，こちらも Jaccard 係数と同様に Dice 係数が大きいほど 2 つの集合の類似度は高いといえる．分母を「和集合の要素数」から「2 集合の平均要素数」とすることで，一方の集合だけ要素数が膨大である場合などに類似度が著しく下がる問題を防ぎ，共通要素数を重視した類似度を計算している．

### <Dice 係数の欠点>

上記でも説明した通り、Dice 係数の定義式は、Jaccard 係数の定義式の分母を「和集合の要素数」から「2 集合の平均要素数」とすることで、差集合の要素数が膨大になった場合に類似度への影響を緩和している。しかし、緩和しているとはいっても、2 集合の要素数に大きな差があり差集合の要素数が膨大になった場合 (例えば、一方の集合が別の集合を内包している等の場合) に、Dice 係数は低下してしまう。

そこで、差集合の要素数の影響を極限まで抑えた Simpson 係数が提案された。

### Simpson 係数

ある集合 A とある集合 B について Simpson 係数  $overlap(A, B)$  は、以下の式で定義される。

$$overlap(A, B) = \frac{|A \cap B|}{\min\{|A|, |B|\}} \quad (3.21)$$

上記の定義より、Simpson 係数は 2 つの集合のうち要素数が少ない方の要素数と共通要素数の割合を表しており、Jaccard 係数や Dice 係数と同様に 0 から 1 の間の値となることがわかる。また、Simpson 係数が大きいほど 2 つの集合の類似度は高い (よく似ている) といえる。Dice 係数の定期式は、Jaccard 係数の定義式の分母  $|A| \cup |B|$  を  $(|A| + |B|)/2$  と変換することで得られた。これに対して Simpson 係数の定義式は、Dice 係数の定義式の分母を「2 集合の平均要素数」から「2 集合のうち少ない方の要素数」とすることで、Dice 係数よりも差集合の要素数による影響を下げ、相対的に共通要素数を重視した類似度計算を実現している。

### <Simpson 係数の欠点>

上記でも説明した通り、Simpson 係数の定義式は、Dice 係数の定義式の分母を「2 集合の平均要素数」から「2 集合のうち少ない方の要素数」とすることで、Dice 係数よりも差集合の要素数による影響を下げ、相対的に共通要素数を重視した類似度計算を実現している。しかし、Simpson 係数では要素数が少ない方の要素数を分母としているため、一方の集合の要素数が少ない場合に、差集合の要素数がどれだけ多くても類似度がほぼ 1 となってしまう。この問題を解決するためには、2 つの集合の要素数に条件 (閾値を設定する、2 集合間の要素数の差が範囲内である等) を付加するとよい。

共起語ネットワークを 3D グラフと 2D グラフによって可視化を行う。3D グラフと 2D グラフにはそれぞれメリットデメリットが存在する。

### 3D グラフのメリット、デメリット

- 単語の共起関係を 3 次元で表現できるため 2D グラフに比べて表現できる情報量が多い。
- 情報量の多さや 3 次元空間であることから、視認性が 2D グラフに比べて悪い

これらのことを踏まえ、3D グラフと 2D グラフの両方で共起語ネットワークを表示できるようにした。3D グラフの作成には Three.js, 2D グラフの作成には Graphvis を用いた。

### Three.js

Three.js はウェブブラウザ上で 3 次元コンピュータグラフィックスを描画するための JavaScript ライブラリである。HTML5 の規格に従っており、プラグイン不要で利用することができる。また、WebGL という 3D グラフィックス API をラッピングしており、簡素なコードで 3DCG を描画することができる。3 次元コンピュータグラフィックスとは、3 次元の立体的な仮想物体を、コンピュータで演算することで平面上に奥行きや質感のある画像を表す手法である。従来は大型計算機が必要であったが、プロセッサの性能向上と GPU の一般化により、物性シミュレーションや 3D ゲームなど、さまざまな分野で利用されている [20]。描画には ”3D Force-Directed Graph” というモジュールを用いており、Json ファイルでデータを与えることで、有向グラフを作成することができる。



# 提案手法

## § 4.1 Google Patentsからのデータ収集の高速化と分類

本研究では、Google Patents から特許情報をスクレイピングすることで収集する。まず特許番号を取得し、それらの番号を用いて特許が表示されているページにアクセスし、そこから特許本文をテキストとして取得する。ユーザーが指定したキーワードの or 検索を Google Patents で行う。Google Patents において or 検索を行うにはワードとワードの間にスペースを開ける必要がある。Google Patents では一度に 1000 件までしか表示することができない。そのため、それぞれが 1000 件を超えないように年代を 1 年ごと区切ってスクレイピングを行う。特許の出願日とその年の 1 月 1 日から 12 月 31 日である特許を取得した。取得したデータを図 4.1 に示す。

本研究の提案手法は、大別すると以下のような工程からなる。

1. 利用者の入力したワードにおける GooglePatents での検索結果を取得する。
2. 取得した特許データの本文に対して Sentence-BERT を用いてベクトル化を行う。
3. 出力されたベクトルに対して次元圧縮を行う。
4. 次元圧縮を行ったデータに対してシルエット分析を行いクラスタリングする。
5. それぞれのクラスターについて、K-means で求めた重心からのユークリッド距離の近い 10 個のデータを用いて各クラスターのタイトルを作成する。
6. ユーザーが指定したクラスターに対して、Simpson 係数を用いて共起関係を導出する。
7. 求めた Simpson 係数をもとに 3D グラフおよび 2D グラフを作成する。

### システムのフロントページおよび対象の選択

提案手法においてユーザサイドに提示されるフロントページを図 4.2 に示す。図 4.2 に示した通り、システムのはじめにユーザにはキーワードの入力画面が表示される。キーワードの入力については一つの単語であればそのまま入力し、複数単語入力したい場合は単語と単語との間にスペースを空けて入力することで入力することができる。また、取得するデータの年数を指定することができる。初期設定では直近 6 年間のデータを収集するようになっているが、キーワードの内容によっては 6 年では十分な量のデータを取得できない場合があるため、ユーザー側で取得する年数を指定できるようにしている。このページにおいてユーザは自身が対象としたいキーワードおよび取得するデータの年数を指定する。

ここで、スクレイピングを行う際に時間がかかってしまう問題を解決するために、python のモジュール threads を用いてマルチスレッドによるスクレイピングを行う。コンピュータの

特許本文のテキスト	特許番号
<p>本発明は、外灯機器が切れたときの不点原因箇所の探査に使用する不点探査装置及び、</p> <p>本発明は、押出成形体の製造方法に関し、さらに詳しくは高剛性であり、屈曲金型に、</p> <p>本発明は、複数のビットにより構成されるビット列を暗号化する暗号化装置に関する、</p> <p>本発明は、既設の鉄塔を支える基礎を改修する工法、及び改修する構造、及びそれに、</p> <p>本発明の実施形態は、送電用鉄塔などの送電系統において使用される塔上開閉装置の、</p>	<p>patent/JP5955646B2/ja</p> <p>patent/JP2012126139A/ja</p> <p>patent/JP2013167729A/ja</p> <p>patent/JP5002735B1/ja</p> <p>patent/JP2013198381A/ja</p>
⋮	リンク
<p>本発明は、電力需要者や電力供給者等が電力の消費や発電によって創出された二酸化、</p> <p>特許法第30条第2項適用 令和4年9月13日に、富山県立富山工業高等学校（富山県富山、</p> <p>本発明は、電力需要者や電力供給者等が電力の消費や発電によって創出された二酸化、</p> <p>本発明は、電力需要者や電力供給者等が電力の消費や発電によって創出された二酸化、</p> <p>本発明は、基準価格算出装置及び基準価格算出方法に関する、</p>	<p>patent/JP7246659B1/ja</p> <p>patent/JP7326641B1/ja</p> <p>patent/JP7336816B1/ja</p> <p>patent/JP7369494B1/ja</p> <p>patent/JP7410349B1/ja</p>

図 4.1: テキストデータのフォーマット

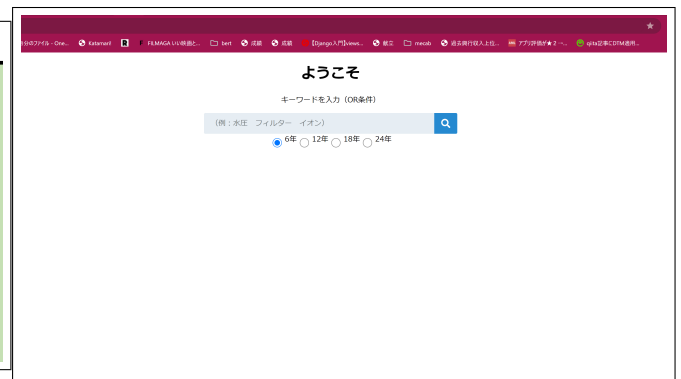


図 4.2: システムのフロントページ

性能によって並列にする数を増やすとかかる時間が長くなる．そのため，並列にする数を6つにして実行を行った．

## threads

スレッドベースの並列タスク実行を助けるモジュールである．threadsを利用することで、一つのpython プログラムの中で複数の処理を同時に実行するマルチスレッド処理を実現できる．threads モジュールには Threads クラスが定義されており、この Threads をスーパークラスとして新しいスレッドを定義する．run () メソッドの中にそのスレッドが実行する処理を書き、start () メソッドを呼び出すことでスレッドが起動し、並列で処理が進む．

join () メソッドを使用すれば、スレッドの終了を待つ制御することもできる．Lock や Semaphore, Event などの動機機構も利用でき、スレッド間でデータを安全に共有することも可能である．ファイル IO やネットワークアクセス時の待ち時間を隠蔽したり、応答性を向上させたり、並列処理による速度向上が図れるなど、threads は即効性の高い並列プログラミングを実現できる．

提案手法では、対象期間を1年ごとに分割し、それぞれの期間を個別のスレッドに割り当てて並列処理を行う．具体的には、6つのスレッドを作成し、各スレッドが1年間のデータをスクレイピングする．つまりスレッド1は1年分、スレッド2は次の1年分となる．そして、各スレッド内で1年ごとにデータ収集を行う．こうすることで期間ごとの並列処理が可能となり、スクレイピングの速度や効率を向上させることができる．最後に、すべてのスレッドが実行完了後、収集したデータを統合することで、対象期間全体のデータセットを取得する．

## データの分類

収集したテキストデータをもとに Sentence-BERT を用いてそれぞれをベクトルに表現する．この時、Sentence-BERT によって出力されるベクトルを pickle を用いて保存しておく．それらのベクトルを UMAP を用いて 15 次元および 2 次元のベクトルに圧縮する．15 次元のベクトルはクラスタリングを行う際と、クラスターの解釈を行う際に用いる．2 次元のベクトルはクラスタリングを行ったデータをプロットする際に用いる．プロットする際に 2 次

元ベクトルを用いる理由は、3次元ベクトルやそれ以上の次元数のベクトルと比較して、2次元のベクトル空間上にプロットされた各データ点間の距離感や密集具合を人間の知覚として把握しやすいためである。また、データ間の類似性を可視化する上でも、2次元空間上では各クラスター内でのデータ点のまとまり方を把握しやすく、データセット全体の構造を俯瞰しやすい。

## § 4.2 クラスターの解釈と共起語ネットワーク

クラスターの解釈を行うために各クラスターの重要語を表示する。ここで、各クラスターのすべての点を対象に重要語を計算しようとする、データの数によっては、莫大な処理時間になる可能性がある。そのため、計算コストを抑えつつ各クラスターの特徴を表すデータを効率的に取得する必要がある。

そこでまず、各クラスター内で最も代表制の高いデータを簡易的に抽出することを試みる。具体的には、K-means アルゴリズムによって求められた各クラスターごとの重心に最も近いデータをユークリッド距離を利用して近い順にソートし上位から10個ずつ取得する。これにより各クラスターの典型的な特徴を示すと考えられるデータを効率よく抽出できる。

その後、この抽出したデータに対して各クラスターごとに重要語や特徴語を計算する。これにより各クラスターの概要や内容の傾向を効率かつ低コストで把握することができる。このような処理フローを設定することで、大規模データに対しても実現できない時間でクラスターの解釈や分析を行うことができる。

### ユークリッド距離

ユークリッド距離は、座標空間において2点間の直線距離を表す指標である。2点をそれぞれ  $(x_1, y_1, z_1, \dots)$  および  $(x_2, y_2, z_2, \dots)$  としたときの座標間のユークリッド距離は以下の式のように求められる。

<n 次元空間の場合>

$$d = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2} \quad (4.1)$$

得られたデータに対しては、専門用語や複合語を考慮した重要語の計算を行う。termextract を用いて、データ内に含まれる重要なキーワードや専門用語を抽出する。最終的に、各クラスターにおいて重要度が高い単語を3つ選出し、それを解釈として表示する。これによってクラスターが表す内容やグループの性質を的確に把握することができる。

### システムのグラフ表示およびクラスターの選択

matplotlib によって作成された散布図を画像データとして保存し、その画像データをhtml 上に表示する。また、各クラスターの内容を表示し、ユーザーが任意のクラスターを指定できるようにする。

### データの前処理

A	B	C	D	E	F	G	H	I	J	K	L	M
系統連系	-1	-1	1000	名詞	固有名称	*	*	*	*	系統連系	*	*
水素電池	-1	-1	1000	名詞	固有名称	*	*	*	*	水素電池	*	*
交流電力	-1	-1	1000	名詞	固有名称	*	*	*	*	交流電力	*	*
水素ガス	-1	-1	1000	名詞	固有名称	*	*	*	*	水素ガス	*	*
発電電力	-1	-1	1000	名詞	固有名称	*	*	*	*	発電電力	*	*
変圧器	-1	-1	1000	名詞	固有名称	*	*	*	*	変圧器	*	*
負荷運転	-1	-1	1000	名詞	固有名称	*	*	*	*	負荷運転	*	*
輸出力	-1	-1	1000	名詞	固有名称	*	*	*	*	輸出力	*	*
負荷率	-1	-1	1000	名詞	固有名称	*	*	*	*	負荷率	*	*
太陽光パネル	-1	-1	1000	名詞	固有名称	*	*	*	*	太陽光パネル	*	*
太陽光発電カーブ	-1	-1	1000	名詞	固有名称	*	*	*	*	太陽光発電	*	*
許容負荷	-1	-1	1000	名詞	固有名称	*	*	*	*	許容負荷	*	*
PCC出力	-1	-1	1000	名詞	固有名称	*	*	*	*	PCC出力	*	*
許容範囲	-1	-1	1000	名詞	固有名称	*	*	*	*	許容範囲	*	*
変圧器バンク	-1	-1	1000	名詞	固有名称	*	*	*	*	変圧器バンク	*	*
電力系統	-1	-1	1000	名詞	固有名称	*	*	*	*	電力系統	*	*
連系ステーション	-1	-1	1000	名詞	固有名称	*	*	*	*	連系ステーション	*	*
連系ユニット	-1	-1	1000	名詞	固有名称	*	*	*	*	連系ユニット	*	*

出力結果

図 4.3: ユーザー辞書のフォーマット (csv)

図 4.4: 出力結果

共起語ネットワークを作成する際に、文章を分かち書きする必要がある。この時、特許には多数の専門用語や複合語が含まれるため、それらを抽出したうえで分かち書きを行う。termextract では専門用語の抽出を行うことはできるが、それらを用いて分かち書きを行うことはできない。そこで、今回用いた分かち書きのモジュールである Janome にユーザー辞書として専門用語や複合語を登録する。

## ユーザー辞書のフォーマット

Janome は独自の単語や品詞情報を追加することができる。この機能を用いることで、特定の文脈や専門用語に適した分かち書きを行うことができる。csv 形式でユーザー辞書を与える必要がある。

### < ユーザー辞書の形式 >

ユーザー辞書はカンマ区切り CSV ファイルで、「表層形、左文脈 ID、右文脈 ID、コスト、品詞、品詞細分類 1、品詞細分類 2、品詞細分類 3、活用型、活用形、原型、音読み、発音」という形式で与える必要がある。今回は表層系に抽出された専門用語や複合語を入力し、左文脈 ID および右文脈 ID は -1 に指定する。コストはすべて 1000 とし、品詞には名詞、品詞細分類 1 には固有名詞を設定する。品詞細分類 2、品詞細分類 3、活用形、活用型、音読み、発音は未設定とし原型においては表層系と同じ文字列を入力する。実際に作成された CSV ファイルを図 4.3 に示す。

この時、作成される辞書の量が多くなると Janome の分かち書きが正しく動作しないことがある。それらを解決するために辞書の量をあらかじめ削減する。削減する方法は事前に求めた専門用語や複合語の重要度が低いものを優先的に削除していく。

分かち書きを行ったのち、共起語ネットワークを作成する。クラスターごとに共起語を分析する。共起語を分析する際、一般的な用語が多く含まれてしまうことがあるため、重要度が高い単語が優先的に含まれるようにする。事前に計算した単語の重要度を用いて、共起語の中に重要ではない単語が含まれているものを除外する。

## 共起ネットワーク



本研究では、単語間の共起関係を分析するために Simpson 係数を用いる。Simpson 係数は Jaccard 係数や Dice 係数と比較して、差集合の要素数による影響をより小さく抑えることができる。ただし、Simpson 係数は一方の集合が他方の真部分集合である場合に 1 となる。そこで、今回、Simpson 係数が 1 となる場合には、それらがもともと一つの単語であったとみなして分析対象から除外している。また、片方の集合の要素数が極端に少ないと係数値が大きくなる傾向があることから、お互いの集合数に 5000 以上の開きがある場合も分析対象から除いている。さらに、共起関係を求める際に、一般的な用語が多く出現する傾向があるため上記で求めた、単語の重要度を用いて重要度が高いものを優先に分析を行う。

## § 4.3 システム化と IPL(Intellectual Property Landscape) への活用

### 提案手法全体の流れ

4 章で示した各手法を統合した課題解決のための提案手法全体の流れの説明を行う。また、提案システム全体のフロー図を図 4.6 に示す。

#### Step 1: キーワードの入力・取得年数の選択

フロントページにてユーザーからのキーワードの入力を取得する。一つの単語だけでなく複数の単語でも検索できるようにすることで広い範囲の検索を可能にする。またここで取得したい年数を指定することができる。初期設定は 6 年間となっており、2017 年から 2023 年までのデータを取得することができ、他にも 12 年、18 年、24 年と選択することができる。

このようにユーザーが選択できるようにすることで、データの取得が足りない、または、多すぎるといったことを回避することができる。また取得するデーターが多いと、スクレイピングに時間がかかってしまうため、それらの回避も行うことができる。ユーザーの用途や目的によって年数を選択することができる。

入力されたキーワードをもとにスクレイピングを行ったデータは次のフローへ送信される。

#### Step 2: 特許の俯瞰とクラスターの選択

Step1 で取得したデーターをもとにしたクラスタリングの結果をプロットする。これらのプロットの結果を見ることで特許全体を俯瞰することができる。またそれらのなかでの分野のまとまりについても見る事ができる。各クラスターについてはそれぞれ違う色の点で描画しており、それらのまとまりをより視覚的にわかりやすいようにしている。

また、クラスターの解釈を図の中に組み込んでしまうと図と文字が重なって水々しくなるため、クラスターの解釈については図の外に記述してある。ユーザーは図の中ク

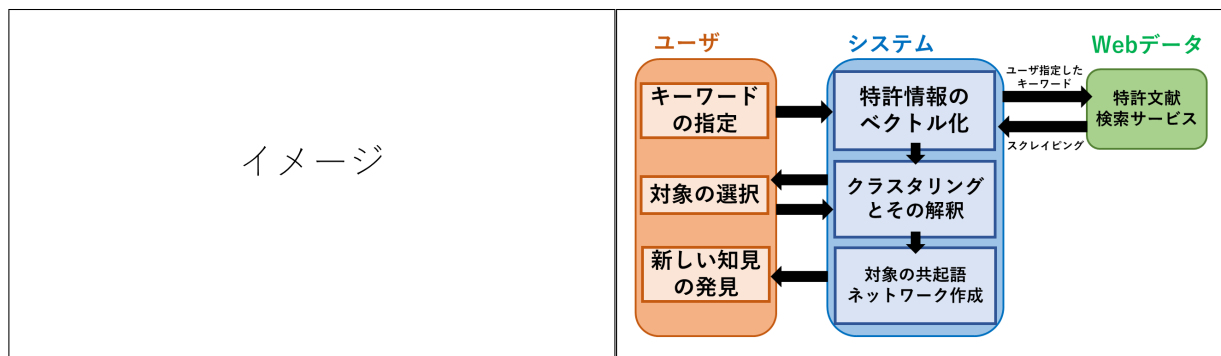


図 4.5:

図 4.6: 提案システム

クラスターの番号と解釈におけるクラスターの番号を照らし合わせることでそれぞれのクラスターの解釈を確認することができる。

以上のことにより、ユーザーは自分が知りたい技術分野や、特許の散らばり具合から、密になっている部分や疎になっている部分に対して分析の対象を選択することができる。

### Step 3: 選択されたクラスターにおける共起語ネットワークの作成

Step2にて選択されたクラスターにおける共起語ネットワークを作成する。Simpson係数を用いて共起関係の分析を行う。そこで計算された係数値をもとに3Dグラフおよび2Dグラフによる可視化を行う。

また、選択されたクラスターに含まれている特許の特許番号を一覧に表示する。さらにその特許番号をクリックすることで、元の特許におけるGooglePatentsでのページが表示されるようになっている。このことで、実際の特許にもアクセスすることが可能となる。

以上の操作の結果もとめられた共起元の単語と共起先の単語およびそれらのSimpson係数をデータフレームに保存しpickleデータの形式で保存する。

### Step 3: 共起語ネットワークの可視化

Step3で作成された共起元の単語と共起先の単語およびSimpson係数の値を用いて言共起語ネットワークを2Dおよび3Dグラフによって可視化する。

# 数値実験並びに考察

## § 5.1 数値実験の概要

## § 5.2 実験結果と考察

おわりに



# 謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部電子・情報工学科情報基盤工学講座の奥原浩之教授，António Oliveira Nzinga René 講師に深甚な謝意を表します．最後になりましたが，多大な協力をしていただいた研究室の同輩諸氏に感謝致します．

2024 年 2 月

平井 遥斗



## 参考文献

- [1] NEC ソリューションイノベータ, ”VUCA とは？意味や読み方、VUCA 時代の組織作りのポイントを解説”, 閲覧日 2024-02-04,  
[https://www.nec-solutioninnovators.co.jp/sp/contents/column/20230623\\_vuca.html](https://www.nec-solutioninnovators.co.jp/sp/contents/column/20230623_vuca.html).
- [2] 株式会社三菱総合研究所, ”代 4 次産業革命における産業構造分析と IoT・AI 等の発展に係る現状及び課題解決に関する調査研究”, 閲覧日 2024-02-04,  
[https://www.soumu.go.jp/johotsusintokei/linkdata/h29\\_03\\_houkoku.pdf](https://www.soumu.go.jp/johotsusintokei/linkdata/h29_03_houkoku.pdf).
- [3] 特許庁, ”広報誌「とっきょ」”, 閲覧日 2024-02-04,  
<https://www.jpo.go.jp/news/koho/kohoshi/>.
- [4] 特許庁, ”経営戦略に資する知財情報分析・活用に関する調査報告書”, 閲覧日 2024-02-04,  
<https://www.jpo.go.jp/support/general/document/chizaijobobunseki-report/chizai-jobobunseki-report.pdf>.
- [5] 東京知的財産総合センター, ”中小企業経営者のための知的財産戦略マニュアル”, 閲覧日 2024-02-04,  
[https://www.tokyo-kosha.or.jp/chizai/manual/senryaku/rmepal000001vypy-att/senryaku\\_all.vol.9.pdf](https://www.tokyo-kosha.or.jp/chizai/manual/senryaku/rmepal000001vypy-att/senryaku_all.vol.9.pdf).
- [6] 特許庁, ”経営戦略を成功に導く知財戦略”, 閲覧日 2024-02-04,  
[https://www.jpo.go.jp/support/example/document/chizai\\_senryaku\\_2020/all.pdf](https://www.jpo.go.jp/support/example/document/chizai_senryaku_2020/all.pdf).
- [7] 特許庁, ”「経営戦略に資する知財情報分析・活用に関する調査研究」について”, 閲覧日 2024-02-04,  
<https://www.jpo.go.jp/support/general/chizai-jobobunseki-report.html>.
- [8] 金融ナビ, ”経営戦略の策定に役立つフレームワーク 7 つ | 経営戦略の代表例も解説”, 閲覧日 2024-02-04,  
[https://financenavi.jp/basic-knowledge/management\\_strategy\\_framework/#tag1](https://financenavi.jp/basic-knowledge/management_strategy_framework/#tag1).
- [9] gikyo.jp, ”Perl による自然言語処理入門”, 閲覧日 2024-02-04,  
<https://gihyo.jp/dev/serial/01/perl-hackers-hub/0031011>.
- [10] 特許庁, ”2019 年度 知的財産権制度入門”, 閲覧日 2024-02-04,  
[https://www.jpo.go.jp/news/shinchaku/event/seminer/text/document/2019\\_syosinsya/1\\_3.pdf](https://www.jpo.go.jp/news/shinchaku/event/seminer/text/document/2019_syosinsya/1_3.pdf).
- [11] 株式会社 日立ソリューションズ・クリエイト, ”テキストマイニングとは？手法や活用法を解説”, 閲覧日 2024-02-04,  
<https://www.hitachi-solutions-create.co.jp/column/technology/text-mining.html>.
- [12] 正林国際特許商標事務所, ”既存技術をほかの用途へ転用する, あるいはビジネス上の課題を解決する既存技術を模索するための IP ランドスケープの活用”, 閲覧日 2024-02-04,  
[https://www.wipo.int/edocs/plrdocs/en/plr\\_2019\\_shobayashi\\_other.pdf](https://www.wipo.int/edocs/plrdocs/en/plr_2019_shobayashi_other.pdf).



- [13] AGIRobots Blog, ”【Transformer の基礎】Multi-Head Attention の仕組み”, 閲覧日 2024-02-04,  
<https://developers.agirobots.com/jp/multi-head-attention/>.
- [14] Nils Reimers, Iryna Gurevych. ”Sentence-BERT : Sentence Embedding using Siamese BERT-Networks”, ArXiv e-prints, 1908. 10084, 2019
- [15] McInnes, L., Healy, J., Melville, J. ”UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction”, ArXiv e-prints, 1802. 03426, 2018
- [16] Hatena Blog, ”UMAP の仕組み-低次元化の理屈を理解してみる”, 閲覧日 2024-02-04,  
<https://kntty.hateblo.jp/entry/2020/12/14/070022>.
- [17] 倉橋 和子, ”分割・併合機能を有する K-Means アルゴリズムによるクラスタリング”, 奈良女子大学学位論文 2007
- [18] Technical Note, ”シルエット分析”, 閲覧日 2024-02-04,  
<https://hkawabata.github.io/technical-note/note/ML/Evaluation/silhouette-analysis.html>.
- [19] MIERUCA AI MEDIA, ”【技術解説】集合の類似度”, 閲覧日 2024-02-04,  
[https://mieruca-ai.com/ai/jaccard\\_dice\\_simpson/](https://mieruca-ai.com/ai/jaccard_dice_simpson/).
- [20] アンドエンジニア, ”Three.js とは？概要やできることを JavaScript 関連術を含めて解説”, 閲覧日 2024-02-04,  
<https://and-engineer.com/articles/ZOWitBIAACMAFtEj>.

