卒業論文

特許情報の質的評価指標と ベクトル化を統合したIPランドスケープ 支援システムの開発

Development of an IP Landscape Support System
Integrating Qualitative Evaluation
Metrics and Vectorization of Patent Information

富山県立大学 工学部 情報システム工学科
2220051 氷見 夏輝
指導教員 António Oliveira Nzinga René 准教授
提出年月: 令和8年2月

目 次

図一覧

表一覧

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
収集される特許データ群	$\mathcal{D}$
個々の特許	d_i
分析対象となる特許の総数	N
特許 d_i における請求項数	c_i
特許 d_i における明細書の文字数	l_i
全特許における請求項数の集合	$\mathcal{C}$
全特許における文字数の集合	$\mathcal{L}$
正規化された請求項数	$\tilde{c}_i$
正規化された文字数	$\tilde{l}_i$
算出された特許 d_i の重要度スコア	S_i
請求項数に対する重み係数	w_c
記述文字数に対する重み係数	w_l
変動係数	α
バイアス項	β
クラスタリングにおける非類似度の総和	J
クラスタ数	k
j 番目のクラスタ	C_j
クラスタ C_j の Medoid	μ_j
データ点間の距離	$d(x, y)$
全データ点の集合	X
初期 Medoid 集合	M
データ点 x_i が割り当てられたクラスタ番号	$label(x_i)$
ある点 x_m を仮の Medoid とした場合のクラスタ内総距離	$Cost(x_m)$
更新された新しい Medoid	x_{new} / μ_j^{new}
データ点 i のシルエット係数	$s(i)$
データ点と同じクラスタ内の他の全点との平均距離	$a(i)$
データ点と最も近い隣接クラスタ内の全点との平均距離	$b(i)$
単語 A を含む文書集合	D_A
単語 B を含む文書集合	D_B
集合 D_A の要素数	$\|D_A\|$

第1章

はじめに

§ 1.1　本研究の背景

現代の産業界において，グローバル化の加速やデジタルトランスフォーメーション（DX）の進展は，企業を取り巻く経営環境を劇的に変化させている．このような不確実性が高く，技術革新のスピードが速い時代において，企業が持続的な競争優位性を確保し続けるためには，自社の保有する技術力や知的財産（IP）を経営戦略の中核に据える「IP ランドスケープ」の実施が不可欠となっている．特許情報は，単なる独占排他権を示す権利書としての側面に留まらず，その技術の権利範囲，詳細な実施形態，発明の背景にある課題解決の意図など，技術経営に必要な情報が凝縮されたデータベースである．したがって，膨大な特許情報を適切に分析し，競合他社の開発動向を把握したり，自社技術の新たな用途を探索したりすることは，経営判断における羅針盤としての役割を果たす．経営層や事業責任者は，これらの情報を活用して市場の機会と脅威を早期に察知し，迅速な意思決定を行うことが求められている [1].

しかしながら，世界的な技術開発競争の激化に伴い，特許出願件数は年々増加の一途をたどっている．世界知的所有権機関（WIPO）の統計によれば，年間の特許出願件数は数百万件の規模に達しており，一つの技術分野に限っても数千から数万件の特許が存在することも珍しくない．もはや人間の専門家がそれらを目視ですべて精査し，関連性を読み解くことは物理的に不可能となっている [2]. この「情報の爆発」という課題に対し，近年では自然言語処理（NLP）技術や機械学習を用いた特許分析の自動化が盛んに研究されている．特に, Bidirectional Encoder Representations from Transformers (BERT) に代表される深層学習モデルを用いて特許のテキスト情報を高次元ベクトルに変換し，それを次元圧縮して2次元マップ上に散布図として可視化する手法は，技術の全体像（ランドスケープ）を俯瞰するための有効な手段として定着しつつある．これにより，人間には知覚できない高次元の意味空間において，類似した技術がどのようにクラスタを形成しているか，あるいはどの分野が競合のいない空白地帯となっているかを視覚的に把握することが可能となった [3].

一方で，既存のベクトル化を用いた可視化手法には，実務的な観点から見て重大な課題が残されている．それは，生成されたマップ上において，すべての特許が「等しい大きさの点」として扱われてしまうという問題である．実際の特許データには，その技術分野のパラダイムシフトを引き起こすような「基本特許」から，既存技術のわずかな改良に過ぎない「周辺特許」，あるいは防衛目的で出願された重要度の低い特許まで，質的な価値には雲泥の差が存在する．しかし，従来の類似度のみに基づいたベクトル化手法では，内容が似ていれば重要度に関わらず近くに配置されるため，クラスタの中心に位置する特許が必

ずしも見るべき価値のある重要な特許であるとは限らない．その結果，分析者はマップ上のクラスタに含まれる数百，数千もの点を一つひとつクリックして内容を確認しなければならず，膨大な「ノイズ」の中から真に注目すべき「重要特許」を見つけ出す作業には，依然として多大な労力と時間を要しているのが現状である．すなわち，特許情報の「量」に対する処理技術は進歩したが，可視化の段階における「質」の評価情報の欠落が，IP ランドスケープの効率化を阻害する大きな障壁となっているのである．経営に資する知見を得るためには，単に技術の地図を描くだけでなく，その地図上で「どこが重要拠点なのか」を即座に識別できる新たな仕組みが必要とされている [4] [5] [6].

§ 1.2　本研究の目的

本研究の目的は，前節で述べた「特許の質的評価の欠落」という課題を解決するために，膨大な特許情報の中から，技術的な内容だけでなく，特許としての「質」や「重要度」を統合的に評価し，分析者が直感的に重要特許を把握できる，新たな IP ランドスケープ支援システムを構築することである．本研究では，単に類似した特許をグルーピングして可視化するだけでなく，その中から「読むべき特許」を即座に提示可能なシステムの実現を目指す．これにより，特許分析にかかる人的コストを大幅に削減し，戦略立案という高度な知的作業にリソースを集中できる環境を提供する．

第一の目的は，特許の重要度を定量的に算出する新たなスコアリングモデルの導入である．従来，特許の価値評価には被引用数などの指標が用いられることが多いが，出願直後の特許や公開されたばかりの特許では引用データが蓄積されていないという問題がある．そこで本研究では，特許文献そのものの構造に着目する．具体的には，特許の権利範囲の広さを示唆する「請求項の数」と，技術開示の詳細さや発明の複雑さを示す「明細書の記述量」という二つの客観的な指標を用いる．これらは特許が出願された時点で確定している情報であり，外部要因に左右されにくい．本研究では，これらを正規化し適切な重み付けを行うことで，出願時期に関わらず，権利として強力であり，かつ技術的に充実した内容を持つ特許を「重要特許」として定義し，スコア化を行う数理モデルを構築する．

第二の目的は，算出された重要度を視覚的に直感可能な形で可視化に反映させることである．従来のシステムでは均一であった散布図上の点のサイズに対し，本研究では算出したスコアをマッピングさせる手法を提案する．これにより，重要な特許は「大きな点」として，そうでない特許は「小さな点」として描画されることになる．分析者は，可視化されたマップを見た瞬間に，各技術クラスタの中で支配的な地位にある特許や，技術的に密度の高い特許を視覚的に識別することが可能となる．これは，IP ランドスケープにおける探索プロセスを,「網羅的な確認作業」から「重要拠点を起点とした効率的な調査」へと変革させるものである．大きな点を中心に周辺技術を確認することで，技術トレンドの核心を短時間で把握できるようになる．

第三の目的は，システムの実用性を高めるためのデータ収集プロセスの高速化である．IP ランドスケープは，一度の分析で終わるものではなく，分析者がキーワードを変え，条件を変えて仮説検証を繰り返す対話的なプロセスである．そのため，支援システムには高い応答性が求められる．しかし，従来のスクレイピング技術を用いた特許データの収集は，Web ページへのアクセス待機時間などにより著しく時間を要し，分析の思考を分断する要因と

なっていた．そこで本研究では，Python の標準ライブラリ等を用いたマルチスレッドによる並列処理を実装し，データ収集から分析結果の提示までの時間を大幅に短縮することを目指す．これにより，ユーザーのストレスを軽減し，よりアジャイルな分析業務を支援する．

以上の取り組みを通じて，本システムは，利用者が入力したキーワードに関連する特許群について,「どのような技術分布になっているか（量的な把握）」と「その中でどれが重要な特許か（質的な把握）」を同時に，かつ迅速に提供する．これにより，技術探索や競合分析，新規用途探索といった知的財産活動の効率を飛躍的に向上させ，経営戦略に資する有益な知見の創出を強力に支援することを最終的な目的とする．

§ 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 本研究で用いる IP ランドスケープや特許情報の定義，自然言語処理技術である BERT，およびクラスタリング手法である K-Medoids 法などの要素技術について解説する．

第 3 章 特許の重要度を算出するためのスコアリング式の定義や，K-Medoids 法のアルゴリズム，共起語ネットワーク構築に用いる類似度係数といった数理モデルについて詳述する．

第 4 章 提案手法について説明する．

第 5 章 開発したシステムを用いた評価実験の結果を示し，スコアリング指標の有効性や並列処理による性能向上，および K-Medoids 法によるクラスタリングの妥当性について考察する．

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．

第2章

知的財産戦略と特許情報の評価

§ 2.1　知的財産用語とIPランドスケープ

現代の企業経営において，知的財産は，単なる法的な保護対象としての「権利」から，競争優位を生み出す源泉としての「経営資源」へとその位置づけを大きく変容させている．かつての知財活動は，自社の発明を特許化し，他社による模倣を排除するという「守り」の側面や，他社の権利を侵害しないための調査といった「リスク管理」の側面が主眼であった．しかし，グローバル化の進展や技術革新のスピード加速，オープンイノベーションの広がりといった経営環境の激変に伴い，知財を能動的に活用して事業戦略をリードする「攻め」の知財戦略が求められるようになった. 知的財産戦略とは，企業が保有する特許，商標，意匠，著作権，ノウハウといった無形資産を経営戦略の中核に据え，事業目標の達成と企業価値の最大化を図るための戦略的活動である．具体的には，以下の 3 つの視点から戦略が構築されることが多い.

オープンイノベーション創出への貢献

- 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

  自前主義からの脱却を図り，外部の技術やアイデアを積極的に取り込むことで，研究開発のスピードアップとリスク分散を図る戦略である．ここでは，自社のコア技術を明確化した上で，不足している技術を補完するための提携パートナー探索や，自社の休眠特許のライセンスアウトなどが含まれる．

- 事業競争力の強化

  自社の製品・サービスにおける技術的優位性を，特許網によって強固に保護し，参入障壁を構築する戦略である．単一の特許ではなく，関連する周辺技術も含めて網羅的に権利化することで，競合他社の回避設計を困難にし，市場における独占的な地位を確保することを目指す．また，M&A（企業の合併・買収）においては，対象企業の技術力評価において知財情報が決定的な判断材料となる．

- 組織・基盤の強化

  ブランド価値の向上や，デジタルトランスフォーメーション（DX）による事業基盤の強化である．特に近年では，SDGs（持続可能な開発目標）への対応として，環境技術に関する知財の公開や活用が，企業の社会的評価を高める要因となっている．

（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駅財産戦略

  グローバル事業展開の形態として，輸出，ライセンシング，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 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．

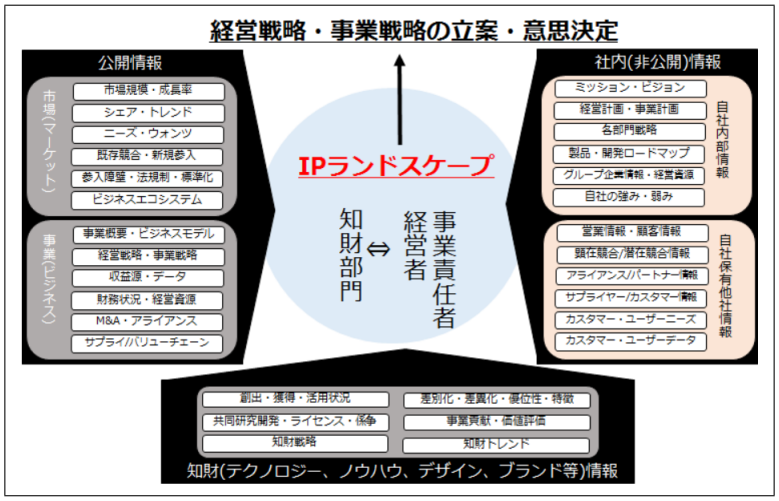

図 2.1: IP ランドスケープの概要

IP ランドスケープの定義と役割

こうした戦略転換を具現化する手法として注目されているのが「IP ランドスケープ」である．IP ランドスケープとは，自社の経営戦略や事業戦略を立案する際，経営情報と知財情報を統合的に分析し，その結果を経営者や事業責任者と共有して，戦略の策定・実行・検証に活かす一連のプロセスを指す. 図 2.1 に IP ランドスケープの概要を示す．「ランドスケープ（風景・景観）」という言葉が示す通り，この手法の目的は，市場における自社と競合他社の立ち位置，技術開発の潮流，新たなビジネスチャンスの芽などを「俯瞰図」として可視化することにある．従来の特許調査が「点」の確認（特定の特許の権利範囲確認など）であったのに対し，IP ランドスケープは「面」や「立体」として市場全体を捉えるアプローチである．IP ランドスケープの実施により，以下のような経営課題への解が得られることが期待される.

- 新規事業の探索

  自社の保有技術を転用可能な異業種・新市場の発見.

- アライアンスパートナーの選定

  技術的補完関係にある企業や，特定の技術を持つスタートアップの特定.

- 競合分析

  競合他社の出願動向から，その企業が将来注力しようとしている技術分野や事業撤退の兆候を早期に察知する.

- ホワイトスペースの発見

  競合が存在せず，かつ市場性が見込める未開拓領域の特定.

本研究では，これらの分析を支援するために，膨大な特許データを自然言語処理と機械学習によって解析し，分析者が直感的に技術動向と重要特許を把握できるシステムを構築する.

IP ランドスケープの主要な情報源となるのが「特許情報」である．特許公報は，発明者や出願人といった書誌情報，技術内容を詳細に記した技術情報，そして独占排他権の範囲を定めた権利情報が一体となった，極めて密度の高い技術文献である．特許分析においては，主に以下の要素が注目される．

- 書誌情報

  出願人，発明者，出願日，公開日，特許分類などが含まれる．これらを時系列で分析することで，特定企業の活動量や，技術分野ごとのライフサイクルを把握することができる．また，共同出願人の分析から，企業間の提携関係を読み解くことも可能である．

- 請求項

  特許権として保護を受けたい範囲を技術的に定義した文章である．「請求項 1」などの独立請求項と，それを限定する従属請求項から構成される．請求項の記載は，権利の広さを決定づけるため，法的解釈において最も重要な部分である．分析的視点では，請求項の数が多い特許は，多面的な実施形態をカバーしているか，あるいは権利範囲を階層的に強固に構築しようとしていることを示唆するため，重要な特許である可能性が高い．本研究では，この「請求項数」を特許の重要度を測る定量的指標の一つとして採用する．

- 明細書

  発明の名称，技術分野，背景技術，先行技術文献，発明が解決しようとする課題，課題を解決するための手段，発明の効果，発明を実施するための形態などが詳細に記述された部分である．ここでは，発明に至った経緯や，従来の技術の問題点，そして本発明がどのようにその問題を解決したかが論理的に説明されている．技術用語の出現頻度や共起関係をテキストマイニングすることで，その特許がどのような技術要素から構成されているか，どのような課題解決を志向しているかを抽出することができる．本研究では，明細書の「記述量」を，技術開示の詳細さや発明の複雑さ・充実度を示す指標として利用する．記述が詳細であることは，発明者がその技術の実施可能性を担保するために多大な情報を開示していることを意味し，技術的価値の高さと相関があると考えられるためである．

§ 2.2　自然言語処理とベクトル化

テキストマイニングと自然言語処理の進化

特許情報の爆発的な増加に伴い，人手による全件読解は不可能となった．そこで，コンピュータを用いて大量のテキストデータから有用な知見を抽出するテキストマイニング技術が不可欠となっている．テキストマイニングの中核をなすのが自然言語処理である．NLP の歴史は古く，初期のルールベースの手法から，統計的確率モデルを用いた手法，そして近年の深層学習を用いた手法へと劇的な進化を遂げている．かつて主流であった「Bag of Words (BoW)」や「TF-IDF」といった手法は，単語の出現頻度に基づいて文書をベクトル化していたが，単語の順序や文脈を考慮できないという欠点があった．例えば,「犬が猫を追いか

けた」と「猫が犬を追いかけた」は，BoW では全く同じベクトルになってしまう．この問題を解決したのが，2013 年に登場した「Word2Vec」である．Word2Vec は，単語の意味を分散表現として学習し，単語間の意味的な近さや演算を可能にした．しかし，Word2Vec は各単語に対して固定のベクトルを割り当てるため，多義語の区別が文脈に応じてできないという限界があった．

BERT

文脈を考慮した動的な単語表現を実現し，NLP 界にブレイクスルーをもたらしたのが，2018 年に Google が発表した「BERT」である．BERT は「Transformer」と呼ばれるニューラルネットワークアーキテクチャの Encoder 部分を多層に積み重ねた構造をしている．BERT の最大の特徴は,「双方向」の学習にある．従来の言語モデルが一方向に文脈を読んでいたのに対し，BERT は「Masked Language Model (MLM)」と「Next Sentence Prediction (NSP)」という 2 つの事前学習タスクを通じて，文章中のある単語を，その前後の文脈全体から予測するように学習される．これにより，同じ単語であっても，文脈が異なれば異なるベクトル表現を獲得することが可能となった．特許文書は，専門用語が多く，かつ係り受け構造が複雑な長文であるため，文脈依存性を高く考慮できる BERT は極めて親和性が高い．

Transformer と Attention 機構

BERT の根幹をなす Transformer は，Attention 機構のみで構成されたモデルである [8]. Attention とは，ある単語を処理する際に，文章中の「他のどの単語に注目すべきか」という重みを計算する仕組みである．具体的には，Multi-Head Attention と呼ばれる機構を用い，複数の異なる「注目点」を並列に学習する．以下の式（2.1）のようになる．

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

ここで，Q（Query）,K（Key）,V（Value）は入力ベクトルから生成される行列である．この計算により，例えば代名詞「それ」が指す具体的な名詞や，動詞と目的語の関係などを，長距離の依存関係であっても捉えることができる．また，RNN のように逐次処理を行わないため，並列計算が可能であり，学習効率が高いという利点もある．なお，Transformer は単語の順序情報を扱えないため，Positional Encoding という位置情報を埋め込むベクトルを入力に加算することで，語順の概念を取り入れている．

Sentence-BERT による文章ベクトル化

BERT は単語レベルのベクトル化や，特定のタスクにおいては最強の性能を誇るが，文章全体の意味ベクトルを生成し，文章同士の類似度を計算するタスクにおいては，計算コストと精度の面で課題があった．BERT の出力層の平均を取るなどの単純な方法では，必ずしも高品質な文章ベクトルが得られないことが知られている．図 2.2 に BERT のイメージ図を示す．そこで本研究では，BERT を改良した Sentence-BERT (SBERT) を採用する．Sentence-BERT は，Siamese Network（シャムネットワーク） 構造を用いている．これは，2 つの BERT モデルにペアとなる文章を入力し，それぞれの出力ベクトルに対して平均プー

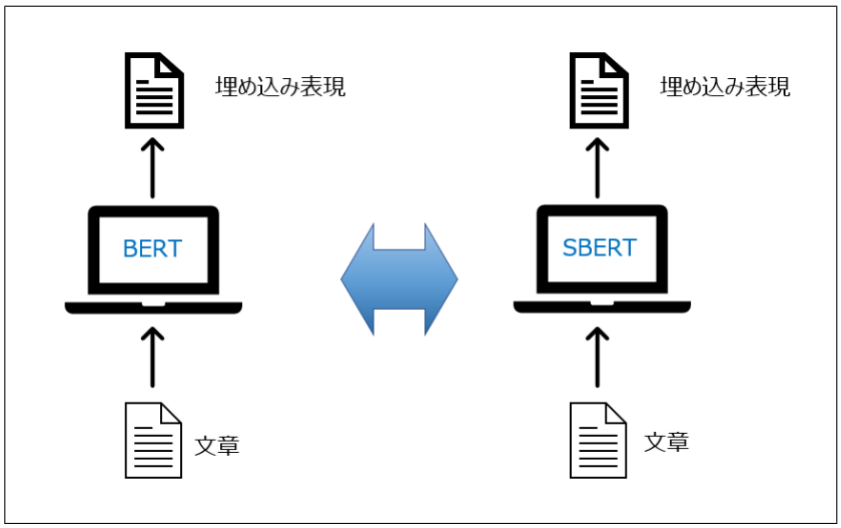

図 2.2: BERT のイメージ図

リングを行い，そのベクトル間の距離に基づいて学習を行う手法である [9]．これにより，意味的に近い文章は近くに，遠い文章は遠くに配置されるようなベクトル空間が形成される．特許分析においては，特許文書全体を一つのベクトルとして表現し，特許間の類似性を定量的に比較する必要があるため，Sentence-BERT は最適な選択肢であると言える．本研究では，Hugging Face で公開されている日本語学習済みモデル（sonoisa/sentence-bert-base-ja-mean-tokens）を使用する．[10]

形態素解析と専門用語抽出

英語と異なり，日本語の文章は単語が空白で区切られていない．そのため，BERT に入力する前処理として，文章を単語単位に分割する「分かち書き」処理が必要となる．本研究では，形態素解析エンジンとして MeCab および Python ラッパーの Janome を使用する．特許文書の特異性は，一般的な辞書には存在しない「複合語」や「未知の専門用語」が極めて多い点にある．例えば「半導体発光素子製造装置」といった複合語が,「半導体」「発光」「素子」「製造」「装置」とバラバラに分解されてしまうと，本来の「一つの技術概念」としての意味が失われるリスクがある [11]．この問題に対処するため，本研究では東京大学の松本研究室が開発した TermExtract を用いる．TermExtract は，専門用語の出現パターンや連接頻度に基づいて，テキストから専門用語を自動抽出するライブラリである．本システムでは，分析対象となる特許群から TermExtract を用いて重要複合語を抽出し，それをユーザー辞書として Janome に動的に登録する処理を実装している．これにより，特許固有の専門用語を正しく認識した上でのベクトル化と共起語分析が可能となり，分析精度を大幅に向上させている [12]．

§ 2.3　クラスタリングと次元圧縮

次元の呪いと次元圧縮の必要性

Sentence-BERT によって生成される特許のベクトルは，通常 768 次元という高次元のデータである．高次元空間においては，データ点同士の距離が均一化してしまい，距離に基づいた類似度判定が困難になる現象（次元の呪い）が発生しやすい．また，人間は 3 次元までしか視覚的に認識できないため，768 次元のデータをそのまま可視化して分析することは不可能である．そこで，データの持つ本質的な構造を保ったまま，2 次元や 3 次元に情報を圧縮する「次元圧縮」技術が必要となる．代表的な手法には，線形圧縮の主成分分析や，非線形圧縮の t-SNE がある．PCA は大域的な構造の維持には優れるが，複雑な非線形構造を捉えきれない．t-SNE は局所的な構造の維持に優れ，可視化においては高い性能を示すが，大域的な位置関係の保存性に難があり，また計算コストが高いという課題があった．

Uniform Manifold Approximation and Projection

本研究では，これらの課題を解決する最新の次元圧縮手法である UMAP を採用する．UMAP は，リーマン幾何学と代数的トポロジーの理論に基づいた手法である [13]. UMAP のアルゴリズムは大きく 2 つの段階からなる．まず，高次元空間において，各データ点の近傍探索（k-nearest neighbors）を行い，データ間の接続関係を表す重み付きグラフ（Fuzzy Simplicial Set）を構築する．次に，低次元空間においても同様のグラフ構造を構築し，2 つのグラフの差異が最小になるように，低次元空間でのデータ点配置を最適化する [14]. UMAP は，t-SNE と比較して以下の利点を持つ．

- 計算速度

  数万件以上のデータに対しても高速に動作する．

- 大域的構造の保存

  局所的なクラスタ構造だけでなく，クラスタ同士の距離関係といった大域的な構造も比較的よく保存する．これにより，IP ランドスケープにおいて「技術分野同士の関連性」をマップ上で読み取ることが可能となる．

- スケーラビリティ

  新たなデータ点に対しても，再学習なしで埋め込みを行うことが可能（本研究では使用していないが，将来的な拡張性として重要）．

クラスタリング手法の比較：K-Means 法と K-Medoids 法

次元圧縮によって可視化された特許群を，技術的な意味のあるグループ（クラスタ）に分類するために，クラスタリングを行う．クラスタリング手法には，デンドログラムを作成する「階層型」と，あらかじめクラスタ数を決めて分割する「非階層型」がある．数千件規模の特許データを扱う場合，計算コストの観点から非階層型が適しており，代表的な手法として K-Means 法 がある [17]. 先行研究では K-Means 法が用いられていた．K-Means 法は，全データの平均値をクラスタの中心として定義し，各点から最も近い重心のクラスタに割り当てる手法である．アルゴリズムが単純で高速であるが，以下の欠点がある．

- 外れ値への弱さ

  重心計算に平均を用いるため，極端な外れ値（ノイズ）が存在すると重心がそちらに引っ張られ，クラスタリング結果が歪む．

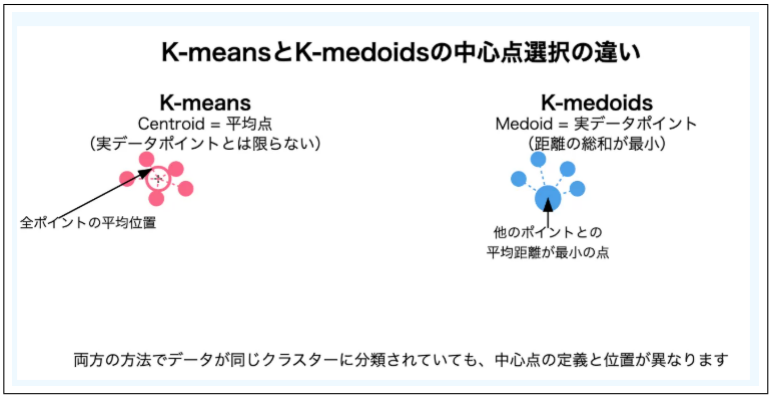

図 2.3: クリスタリング手法の違い

- 非実在の代表点

  重心はあくまで計算上の座標であり，実在する特許データではない．そのため,「このクラスタの代表的な特許はどれか」という問いに対して，重心に最も近い特許を別途探索する必要がある．

K-Medoids 法の採用

本研究では，K-Means 法の課題を克服するため，K-Medoids 法を採用する.K-Medoids 法は，クラスタの代表点として，重心ではなく，クラスタ内に存在する実際のデータ点を使用する手法である．具体的なアルゴリズムとしては，各データ点と Medoid との非類似度の総和が最小になるように，Medoid を探索・更新していく．図 2.3 に K-Means 法と K-Medoids 法の違いを示す．K-Medoids 法を採用する利点は以下の通りである．

1. **頑健性**

   平均値を使わないため，特異な技術やノイズデータの影響を受けにくく，より実態に即したグルーピングが可能である．特許データは，他に類を見ない独創的な発明が含まれることが多いため，この特性は重要である．

2. **解釈性**

   各クラスタの代表点（Medoid）は, 必ず実在する特定の特許文献となる. そのため, システム上で「クラスタ 0 の代表特許」としてその文献を提示することで，ユーザーはそのクラスタがどのような技術分野なのかを，具体的な特許を読んで直感的に理解することができる. これは対話的な分析システムにおいて，ユーザーエクスペリエンスを大きく向上させる要素である．

3. **任意の距離尺度の利用**　K-Means 法は通常ユークリッド距離を前提とするが,K-Medoids 法は距離行列さえ定義できれば, コサイン類似度やジャカード係数など，任意の非類似度尺度を適用可能である．

第3章

提案手法における評価指標とアルゴリズム

§ 3.1　特許重要度スコアの算出モデル

重要度指標の選定理由

特許情報の可視化において，従来の 2 次元散布図は，高次元のベクトル空間における「意味的な近さ」を表現することには成功していた．しかし，マップ上のすべてのプロット（特許）が同一のサイズで描画されるため，ユーザーは視覚的に特許の質的な重みづけを行うことができなかった．特許の価値評価指標としては，一般的に「被引用数」や「パテントファミリー数」,「経過情報（権利維持期間）」などが用いられることが多い．しかし，これらの指標は「時間の経過」に依存する．被引用数は特許公開から数年が経過し，後続の出願が増えて初めて蓄積される指標であり，出願されたばかりの最新技術を分析する IP ランドスケープにおいては速報性に欠けるという致命的な欠点がある．そこで本研究では，特許出願の時点で確定しており，かつ特許文献自体の構造から客観的に導出可能な以下の 2 つの指標を採用し，独自のスコアリングモデルを構築した．

1. **請求項数**

   請求項は，特許権の権利範囲を法的に画定する唯一の記述部分である．特許法において，請求項は「特許を受けようとする発明」を明確に記したものであり，侵害訴訟における権利範囲の解釈は請求項の記載に基づいて行われる．請求項数が多いということは，以下のいずれか，あるいは両方の意味を持つ．

   - 多面的な権利保護

     発明を「物の発明」「方法の発明」「生産方法の発明」など，多角的なカテゴリーで権利化しようとしている．

   - 階層的な防衛網

     最も広い概念である独立請求項に対し，それを限定・具体化した従属請求項を多数配置することで，仮に独立請求項が無効とされても，従属請求項で権利を維持できるよう「強い権利網」を構築している．知財実務において，基本特許や事業の中核をなす重要特許は，第三者による回避設計を防ぐため，必然的に請求項数が多くなる傾向がある．したがって，請求項数は特許の「権利としての強靭さ」や「技術的カバレッジの広さ」を測る代理変数として極めて有効である．

2. **明細書の記述量**

明細書は，発明の技術的内容を社会に開示する代償として独占権が付与されるという特許制度の根幹をなす部分である．特許法には「実施可能要件」が定められており，明細書には「その発明の属する技術の分野における通常の知識を有する者がその実施をすることができる程度に明確かつ十分に」記載しなければならない．画期的な発明や，技術的に高度で複雑なシステム，あるいは多くの応用例を持つ発明は，この要件を満たすために必然的に記述量が増加する．逆に，アイデアのみの先行防衛的な出願や，権利範囲が狭い改良発明は，記述が簡潔になる傾向がある．また，文字数が多いということは，背景技術，課題，解決手段，効果，図面の説明などが詳細に記されていることを意味し，テキストマイニングの観点からも豊富な情報量を含んでいると言える．したがって，明細書の文字数は，その特許に込められた「技術情報の密度」や「発明の充実度」を示す指標として採用した．

正規化とスコア算出式

収集される特許データ群を $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ とする．ここで N は分析対象となる特許の総数である．各特許 d_i は，メタデータとして請求項数 $c_i \in \mathbb{N}$ と明細書の文字数 $l_i \in \mathbb{N}$ を持つ．これらの値は，特許の分野や出願年によってスケールが大きく異なる．例えば，請求項数は通常 1 から 50 程度の範囲に分布するが，中には 100 を超えるものもある．一方，文字数は数千文字から数万文字のオーダーである．これら単位の異なる変数を単一のスコアに統合し，かつ可視化ライブラリの描画パラメータに適応させるためには，適切なスケーリング処理が不可欠である．本研究では，外れ値の影響を考慮しつつ，分布の範囲を $[0, 1]$ に揃える Min-Max 正規化を採用した．

まず，全特許における請求項数の集合を $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$，文字数の集合を $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$ とする．特許 d_i における正規化された請求項数 $\tilde{c}_i$ および正規化された文字数 $\tilde{l}_i$ は以下の式 (3.1)(3.2) で定義される．

$$\tilde{c}_i = \frac{c_i - \min(\mathcal{C})}{\max(\mathcal{C}) - \min(\mathcal{C})} \tag{3.1}$$

$$\tilde{l}_i = \frac{l_i - \min(\mathcal{L})}{\max(\mathcal{L}) - \min(\mathcal{L})} \tag{3.2}$$

ここで，$\min(\mathcal{C})$ および $\max(\mathcal{C})$ はそれぞれ収集された特許群の中での請求項数の最小値と最大値である．この処理により，すべての特許について $0 \leq \tilde{c}_i, \tilde{l}_i \leq 1$ の範囲に収まる無次元化された値が得られる．

次に，これらを加重平均して最終的な重要度スコア S_i を算出する．本研究では，IP ランドスケープにおいて「権利範囲の広さ」を「記述の長さ」よりも重視すべきであるという戦略的観点に基づき，重み係数を決定した．記述が長くても権利範囲が狭ければ事業的な脅威は小さいが，記述が短くても権利範囲が広ければ基本特許となり得るからである．この考えに基づき，請求項数に重み $w_c = 0.7$，記述文字数に重み $w_l = 0.3$ を設定した．さらに，Web ブラウザ上の可視化ライブラリでの描画において，ノードのサイズとして直接利用可能な値に変換するため，以下の式 (3.3) で線形変換を行う．

$$S_i = \left(\tilde{c}_i \times w_c + \tilde{l}_i \times w_l\right) \times \alpha + \beta \tag{3.3}$$

ここで，α は変動係数，β はバイアス項である．本システムの実装では，$\alpha = 180, \beta = 20$ と設定している．定数項 $\beta = 20$ は，スコアが最小（正規化値が 0）の場合でも，マップ上で視認可能な最小サイズ（20px）を保証するためのバイアス項である．これがないと，重要度の低い特許が点として見えなくなってしまう．係数 $\alpha = 180$ は，スコアに応じて最大サイズを決定するスケーリング係数である．スコアが最大（正規化値が 1）の場合，サイズは $20 + 180 = 200$px となる．これにより，最小の特許と最大の特許の間には 10 倍のサイズ差が生まれ，視覚的に「圧倒的な重要度の差」を表現することが可能となる．

このアルゴリズムの実装により，権利範囲が広く，かつ技術詳細が開示されている特許ほど，マップ上で巨大なノードとして描画され，分析者の注意を自然と惹きつけるユーザインターフェースを実現している．

§ 3.2　K-Medoids 法によるクラスタリングアルゴリズム

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．図 3.1 に UMAP による次元圧縮の概念図を示す．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，計算がようであるが，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である．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 UMAP を用いる．

アルゴリズムの概要と採用理由

本システムでは，次元圧縮後の特許ベクトル群を技術的な意味のあるグループに分類するために，K-Medoids 法 (Partitioning Around Medoids: PAM) を採用している．一般的なクラスタリング手法である K-Means 法は，クラスタの中心として「重心」を用いる．重心はクラスタ内全点の座標の平均値であり，計算コストが低いという利点がある．しかし，特許データ分析においては以下の 2 つの大きな欠点がある．

1. **非実在性**

   重心は計算上の仮想的な点であり，実在する特許文献ではない．ユーザーが「このクラスタは具体的にどんな技術か？」を知りたい場合，重心の座標に近い特許を別途探索する必要があり，直感的な解釈が難しい．

2. **外れ値への脆弱性**

   平均値は外れ値の影響を強く受ける．特許データには，既存の技術体系に属さない独創的な発明（外れ値）が含まれることが多く，K-Means 法ではこれらに引っ張られてクラスタの中心がずれ，分類精度が低下する恐れがある．

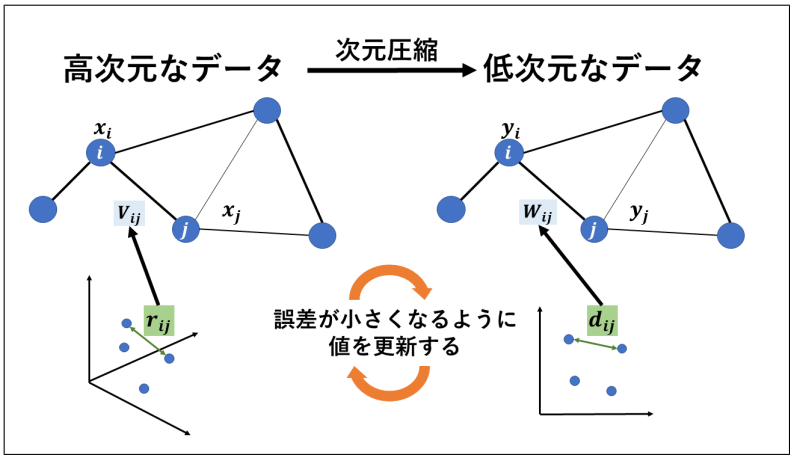

図 3.1: UMAP による次元圧縮

これに対し，K-Medoids 法はクラスタの代表点として，必ずクラスタ内に存在する実在のデータ点を選択する．これにより，外れ値に対してロバストであり，かつ「このクラスタの代表特許はこれである」と具体的な文献を提示できるため，分析結果の解釈性が飛躍的に向上する．

実装アルゴリズムの詳細ステップ

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, \mu_j) \tag{3.4}$$

ここで，式（3.4）において k はクラスタ数，C_j は j 番目のクラスタ，μ_j はそのクラスタの Medoid である．本研究の実装では，距離 d として，UMAP によって圧縮されたベクトル空間におけるユークリッド距離を使用している．具体的なアルゴリズムは以下のステップで実行される．

1. **初期化**

   全データ点 $X = \{x_1, x_2, \ldots, x_N\}$ の中から，重複なしでランダムに k 個のデータ点を選び，初期 Medoid 集合 $M = \{\mu_1, \mu_2, \ldots, \mu_k\}$ とする．

2. **割り当て**

   各データ点 x_i について，現在の Medoid 集合 M の中で最も距離が近い Medoid μ_{best} を探索し，そのデータ点を対応するクラスタ C_{best} に割り当てる．以下の式（3.5）のようになる．

$$\text{label}(x_i) = \underset{j \in \{1,\ldots,k\}}{\text{argmin}}\ d(x_i, \mu_j) \tag{3.5}$$

3. **更新**

   各クラスタ C_j について，新たな Medoid を探索する．クラスタ内のすべての点 $x_m \in C_j$ を候補とし，その点 x_m を仮の Medoid とした場合の「クラスタ内総距離（Cost）」を以下の式（3.6）で計算する．

$$\mathrm{Cost}(x_m) = \sum_{x_p \in C_j} d(x_p, x_m) \tag{3.6}$$

このコスト $\mathrm{Cost}(x_m)$ が最小となる点 x_{new} を，クラスタ C_j の新しい Medoid μ_j^{new} として更新する．これにより，クラスタ内の「中心的な特許」が選ばれる．

4. **収束判定**

更新前後の Medoid 集合を比較する．すべての Medoid が変化していなければ，アルゴリズムは収束したとみなして終了する．変化があればステップ 2 に戻り，再割り当てと更新を繰り返す．ただし，無限ループを防止するため，最大反復回数を設けている．

シルエット分析によるクラスタ数の自動決定

クラスタリングにおける最大の課題は，最適なクラスタ数 k を人間が事前に決定しなければならない点にある．適切な k は分析対象のキーワードや技術分野によって変動するため，固定値（例えば常に $k = 10$）ではデータの分布構造を正しく反映できない．そこで本システムでは，シルエット分析 (Silhouette Analysis) を導入し，データの分布に基づいて最適な k を自動決定するロジックを実装した．[18]

シルエット係数 $s(i)$ は，あるデータ点 i が「自身のクラスタにいかに密接しているか（凝集度）p」と「隣接する他クラスタといかに離れているか（乖離度）」のバランスを評価する指標であり，以下の式（3.7）で定義される．

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3.7}$$

ここで，

$a(i)$: データ点 i と同じクラスタ内の他の全点との平均距離（凝集度．小さいほど良い）．

$b(i)$: データ点 i と最も近い（隣接する）他のクラスタ内の全点との平均距離（乖離度．大きいほど良い）．

$s(i)$ は -1 から 1 の値をとり，1 に近いほどそのデータ点が適切なクラスタに分類されていることを示し，0 付近であればクラスタ境界上にあり，マイナスの場合は誤ったクラスタに分類されている可能性が高いことを示す．本システムでは，クラスタ数 k を 3 から 20 の範囲で変化させながら K-Medoids 法を試行し，それぞれの k における全データのシルエット係数の平均値を計算する．そして，この平均値が最大となる k を「最適クラスタ数」として採用し，最終的なクラスタリング結果として提示する．これにより，ユーザーの主観に頼ることなく，データの客観的な構造に基づいた最適な技術分類が可能となった．

先行研究との比較検討

本研究は，先行研究「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」を基礎としつつ，アルゴリズムの高度化，可視化アプローチの革新，およびシステムの実用性向上において，質的・量的に大きな拡張を行ったものである．以下に，具体的な比較検討を詳細に述べる．

アルゴリズム面での比較：重心ベースから実在点ベースへ

先行研究では，クラスタリング手法として K-Means 法 が採用されていた．K-Means 法はアルゴリズムが単純で計算が高速であるという利点がある．しかし，前述の通り，クラスタの「重心」は計算上の仮想的な座標点であり，実在する特許文献とは一致しない．平井卒論のシステムでは，クラスタの特徴を把握するために「クラスタ内の重要語」を表示する機能はあったが,「このクラスタを代表する特許はどれか？」というユーザーの根源的な問いに答える機能は弱かった．重心に最も近い特許を近似的に探索することは可能だが,外れ値の影響で重心が実際のデータ密集地帯からずれていた場合，代表性が低い特許が提示されるリスクがあった.

これに対し本研究では，K-Medoids 法 を導入した．K-Medoids 法は，クラスタの中心として必ず「実在する特許」を選択する.

- 解釈性の向上

  システムはクラスタの代表として，具体的な特許番号，タイトル，要約を即座に提示できる．ユーザーは代表特許を一読するだけで，そのクラスタの技術特性を直感的に把握できる．これは「説明可能な AI」の観点からも重要な進歩である.

- ロバスト性の向上

  平均値ではなく中央値的なアプローチをとるため，特許データ特有の「外れ値（極めて特異な技術用語を含む特許)」の影響を極小化できる．多様な技術が混在する大規模特許データセットにおいて，より純度の高い（意味的にまとまりのある）クラスタリングを実現した.

可視化アプローチの比較：位置情報のみから重要度情報の統合へ

先行研究における可視化は，特許ごとのベクトル類似度を 2 次元平面上の「位置」として表現することに留まっていた．マップ上ではすべての特許が 「等しい大きさの点」 として描画されており，ユーザーは点の密集度合いからトレンドを推測することはできても，個々の特許の質的な重要性を判断することは不可能であった．結果として，その技術分野のパラダイムシフトを引き起こした「基本特許」も，微細な改良に過ぎない「周辺特許」も，同じ「点」として埋没してしまっていた．分析者は，重要な特許を見つけるために，多数の点を手当たり次第にクリックして内容を確認する必要があり，探索効率に課題があった.

本研究では，この課題に対する根本的な解決策として,「重要度スコアに基づくノードサイズの可変表示」 を実装した.

- 質的情報の可視化

  第 3.1 節で定義した「請求項数」と「記述量」に基づく独自スコアをノードの半径にマッピングすることで，重要な特許を「巨大な点」として描画した.

- 探索プロセスの革新
  ユーザーはマップを一目見るだけで,「どこに有力な特許が存在するか」を瞬時に認識できる.「大きな点を中心に周辺を見る」という，重要拠点を起点とした効率的な探索フローを提供することで，IP ランドスケープにおける分析時間を劇的に短縮し，意思決定の質を向上させた．これは，特許情報の「クラスタリング」に留まっていた先行研究を,「評価」の領域まで拡張した点において，大きな新規性を有する．

システム実装面での比較：単一処理から並列処理へ

先行研究のシステムでは，Web スクレイピングによるデータ収集が順次処理で行われており，数千件のデータ取得に数十分単位の時間を要することもあった．IP ランドスケープは，分析者がキーワードを変え，条件を変えて仮説検証を繰り返す「対話的プロセス」であり，待ち時間の長さは分析の思考を分断する致命的なボトルネックであった．

本研究では，この実用面での課題を解決するため，Python の threading モジュールを用いた マルチスレッド並列処理 をアーキテクチャレベルで実装した．具体的には，データ取得期間を 1 年単位で分割し，複数のスレッドで同時に Google Patents へのアクセスとデータ解析を行う．これにより，データ収集時間を理論値に近い倍率で短縮することに成功した．この高速化により，ユーザーはストレスなく試行錯誤を繰り返すことが可能となり，本システムは実験的なプロトタイプから，実務に耐えうる分析ツールへと昇華したと言える．

以上の比較から，本研究は先行研究の基礎的な枠組み（ベクトル化と可視化）を継承しつつも,「重要度の可視化（スコアリング）」「分類のロバスト性（K-Medoids）」「処理の高速化（並列処理）」という 3 つの軸において，明確な技術的進歩と実用的価値を付加したものであると結論付けられる．

§ 3.3　共起語抽出における類似度係数

共起語ネットワークの目的と意義

特許情報を可視化した散布図は技術の全体像を把握するのに役立つが，各クラスタが具体的に「どのような技術テーマ」で構成されているかを理解するには，テキスト内容の分析が必要である．本研究では，クラスタ内の特許群から抽出された重要語間の「共起」関係を分析し，ネットワークグラフとして可視化する機能を提供する．共起とは，2 つの単語が同一の文書内で一緒に出現する現象を指す．強く共起する単語ペアは，技術的な結びつきが強いことを意味する．これをネットワーク化することで，技術要素同士の関係性を構造的に理解することが可能となる [19].

類似度係数の比較と選定

共起の強さを測る指標として，Jaccard 係数，Dice 係数，Simpson 係数などが知られている．それぞれの特性を比較し，本研究に最適な指標を選定する．ここで，単語 A を含む文書集合を D_A，単語 B を含む文書集合を D_B とし，$|D_A|$ を集合の要素数とする．

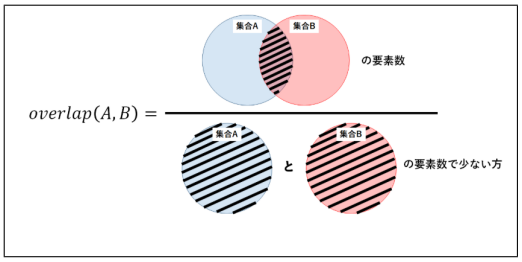

図 3.2: Simpson 係数の概念図

1. **Jaccard 係数**
   最も一般的な類似度指標だが，出現頻度に大きな差がある場合に係数が極端に小さくなる．特許文書では,「装置」「方法」のような頻出語と，希少な専門用語の関係を見たい場合が多いため，Jaccard 係数では重要な関係を見逃すリスクがある．以下の式（3.8）で定義される．

$$J(A,B) = \frac{|D_A \cap D_B|}{|D_A \cup D_B|} \tag{3.8}$$

2. **Dice 係数**
   調和平均を用いた指標．Jaccard よりは頻度差の影響を受けにくいが，依然として包含関係の検出には弱い．以下の式（3.9）で定義される．

$$D(A,B) = \frac{2|D_A \cap D_B|}{|D_A| + |D_B|} \tag{3.9}$$

3. **Simpson 係数 (Overlap Coefficient)**
   分母に小さいほうの集合のサイズを用いる．この定義により，単語 A の出現文書がすべて単語 B の出現文書に含まれている場合（$D_A \subset D_B$），$|D_A| < |D_B|$ であれば係数は 1.0 となる．これは，出現頻度の低い専門用語が，出現頻度の高い一般用語と共起している関係を抽出するのに極めて有効である．以下の式（3.10）で定義される．

$$S(A,B) = \frac{|D_A \cap D_B|}{\min(|D_A|, |D_B|)} \tag{3.10}$$

本研究における採用指標とフィルタリングアルゴリズム

本研究では，特許技術の階層構造を可視化することを重視し，Simpson 係数 を主要な指標として採用した. 図 3.2 に Simpson 特許文書においては,「具体的な技術手段」は必ず「その技術が属する広義の分野用語」と共起する傾向があるため Simpson 係数が最も適している．

ただし，Simpson 係数には副作用がある．出現頻度が極端に少ない単語ペアでも，たまたま同じ文書にあれば係数が 1.0 となり，ノイズとして検出されやすい．また，一般的すぎる単語との結合も過剰に検出されやすい．そこで，本システムの実装においては，以下の複合的なフィルタリング処理を実装し，ネットワークの質を担保している．

表 3.1: 類似度係数の比較表

係数名	特徴	特許分析におけるメリット（上段）・デメリット（下段）
Jaccard係数	集合の「和」に対する「共通部分」の割合．最も一般的な類似度指標	直感的で理解しやすい．文書サイズや単語出現数が同程度の場合の類似性判定に適している．
		出現頻度に大きな差がある場合，分母が大きくなり係数が極端に小さくなるため，重要な共起関係を見逃しやすい．
Dice係数	集合の要素数の「算術平均」に対する共通部分の割合	Jaccard係数と似た挙動を示すが，共通部分の重みが大きくなるため，Jaccardよりやや高い値が出る傾向がある．
		Jaccard係数同様，出現頻度の差が激しい単語間の関係を捉えるのが難しく，階層構造の抽出には不向きである．
Simpson係数	共通部分が「小さい方の集合」にどれだけ含まれているかの割合	包含関係の抽出に極めて強い．頻出語と専門用語の結びつきを検出できるため，特許の技術体系の可視化に最適である．
		出現数が極端に少ない単語が偶然一致した場合でも係数が1.0になりやすく，ノイズを拾いやすい．

- 頻度差フィルタ

  単語 A と B の出現回数の差が一定の閾値以上の場合は，関係性が希薄である，あるいは一方が一般的すぎる語であるとみなしてエッジを生成しない．これにより，意味のない「超頻出語」への集中を防ぐ．

- 完全一致除外フィルタ

  Simpson 係数が 1.0 となる（完全包含）ケースのうち，出現頻度が低いものはノイズである可能性が高く，また出現頻度が高いものは単なる表記揺れや類義語である可能性が高い．可視化においては，これらはネットワークを複雑にする要因となるため，本研究では可視化の明瞭性を優先し，あえて係数が 1.0 未満のペアのみを採用するロジックを組み込んでいる．

- 重要度フィルタ

  TermExtract によって算出された「用語重要度」が高い単語ペアを優先的に抽出する．

  これらのフィルタリングを経たエッジのみを用いて，PyVis による 2D ネットワークおよび Three.js による 3D ネットワークを描画することで，ユーザーは技術的な意味のある「濃い」つながりのみを直感的に把握することができる．

第4章

提案手法

本章では，第 3 章で定義した数理モデルとアルゴリズムを，実際の IP ランドスケープ支援システムとしてどのように実装したかについて詳述する．本システムは，ユーザーが入力したキーワードに基づいて特許データを収集・解析し，その結果を対話的なインターフェースで可視化する Web アプリケーションである．開発言語には Python を採用し，Web フレームワークとして Flask，スクレイピングに Selenium，数値計算に NumPy/Pandas，自然言語処理に Transformers/MeCab，可視化に Matplotlib/Seaborn/PyVis を用いるという，データサイエンスと Web 開発の技術を統合したアーキテクチャを構築した．以下に，システムの全体構成，並列処理によるデータ収集の高速化，および重要度を反映した可視化インターフェースの実装詳細について述べる．

§ 4.1　システムの全体構成と並列処理

システムアーキテクチャと Flask フレームワーク

本システムは，Python の軽量 Web アプリケーションフレームワークである Flask を基盤として構築されている．Flask を採用した理由は，そのシンプルさと拡張性の高さにある．IP ランドスケープのような分析ツールでは，データ処理のバックエンドと結果表示のフロントエンドが密接に連携する必要があるが，Flask は Python の豊富なデータ分析ライブラリとの親和性が極めて高く，アルゴリズムの試行錯誤をシステムへ即座に反映できる利点がある．

システムは主に以下の 3 つの層で構成されている．

1. **プレゼンテーション層**

   HTML/CSS/JavaScript により構築され，ユーザーからの検索クエリの受け付けや，分析結果の描画を行う．Flask のテンプレートエンジンである Jinja2 を用いて，Python 側で生成された動的なデータを HTML に埋め込んでいる [20].

2. **アプリケーション層**

   Flask のルーティング機能により，URL リクエストを適切な処理関数に振り分ける．セッション管理機能を用いて，ユーザーが入力したキーワードや，途中経過のデータフレームをページ遷移間で保持する設計となっている．

3. **データ処理層**

実際に特許データの収集，ベクトル化，クラスタリング，スコアリングを行うコアロジック部分である．ここでは，Selenium によるブラウザ操作や，Transformers による BERT モデルの推論が実行される．

スクレイピングにおける課題と Selenium の採用

特許データの収集源として，Google Patents を利用している．Google Patents は Web API を公式には提供していないため，Web スクレイピング技術を用いてデータを取得する必要がある．当初は軽量な requests ライブラリと BeautifulSoup の組み合わせを検討したが，Google Patents の検索結果ページは JavaScript によって動的に生成される要素が多く，静的な HTML 取得だけでは必要な情報を網羅できないという課題があった．そこで本システムでは，Web ブラウザ自動操作ツールである Selenium WebDriver を採用した．Selenium は，Chrome などのブラウザをプログラムから直接制御し，JavaScript の実行完了を待ってから DOM（Document Object Model）を解析できるため，動的な Web サイトに対しても確実なデータ収集が可能である．本実装では webdriver.Chrome を使用し，ヘッドレスモードオプションを付与することで，GUI 画面を表示せずにバックグラウンドで高速にブラウザを動作させる構成とした [21].

Threading モジュールによるマルチスレッド並列処理の実装

本システムの最大の実装上の特徴は，データ収集プロセスの 並列化にある．従来のシングルスレッド処理では，1 件の検索クエリに対して，例えば 2000 年から 2023 年までの 24 年分のデータを順次取得していた．Web スクレイピングは，サーバーへのリクエスト送信，レスポンス待機，DOM のレンダリングといった I/O（入出力）待ち時間が処理時間の大半を占める「I/O バウンド」なタスクである．そのため，順次処理ではこの待ち時間が累積し，数千件のデータ取得に数十分を要することも稀ではなく，対話的な分析システムの応答速度として致命的であった．

この課題を解決するため，Python 標準の threading モジュールを用いたマルチスレッド処理を実装した．具体的な実装ロジックは以下の通りである．

1. **時間軸によるタスク分割**

分析対象期間を 1 年単位のサブタスクに分割した．Google Patents の検索 URL パラメータには before と after があり，これを利用して特定の 1 年間に出願された特許のみを検索する URL を生成する．

2. **スレッドワーカーの定義**

各期間のデータを取得するための専用関数として W1 〜 W12 を定義した．例えば W1(url) 関数は，渡された URL（ある 1 年分）にアクセスし，ページネーションを巡回して特許データをリスト desc1 に格納する役割を持つ．各ワーカー関数内では，独立した WebDriver インスタンスが割り当てられており，互いに干渉することなく並列にブラウザ操作を行うことができる．

3. **スレッドの生成と制御**

   threading.Thread クラスを用いて，各ワーカー関数を独立したスレッドとして生成する．実装コードでは，ユーザーが指定した取得期間に応じて，起動するスレッドの組み合わせを動的に制御している．
4. 例えば，直近6年分が指定された場合，thr19〜thr24 の6つのスレッドを生成し，start() メソッドで一斉に起動する．
5. 24年分全期間が指定された場合は，マシンスペックとネットワーク負荷を考慮し，6スレッドずつのバッチ処理として実行する制御を行っている．
6. **データの同期と統合**

   各スレッドは非同期に動作するため，すべてのデータ収集が完了するのを待つ必要がある．これには join() メソッドを使用する．すべてのスレッドの join() が完了した時点で，各スレッドが個別のリストに格納したデータを，一つの巨大なリスト desc に結合する．

データ取得と前処理のパイプライン

スクレイピングにより取得されたデータは，以下の要素を含むリストとしてメモリ上に保持される．

- 本文テキスト自然言語処理の対象となる明細書や要約のテキスト．
- URL データの出典元リンク．
- 請求項数スコアリング計算に使用.DOM 要素数から算出．
- 文字数スコアリング計算に使用．テキスト長 len(text) で算出．

取得されたテキストデータは，そのままではノイズが含まれているため，正規表現を用いたクリーニング処理が行われる．具体的には，特許特有の記号や，意味を持たない特殊文字，HTML タグの残骸などを除去し，Sentence-BERT に入力可能なクリーンな文字列リストへと変換される．また，並列処理によって収集されたデータは順序が保証されないため，必要に応じて時系列ソートや重複排除の処理が行われるが，本システムではベクトル化による類似度解析が主目的であるため，順序よりもデータの網羅性を優先している．収集された生データは pickle 形式で text_data.pickle として保存され，後続のプロセスで再利用可能な設計となっている．これにより，一度収集したデータに対してパラメータを変えて何度も分析を行う際の時間を大幅に短縮している．

§ 4.2　ハイパーパラメータの選定とテストデータの比較結果

次元圧縮（UMAP）のパラメータ選定

特許テキストのベクトル化には Sentence-BERT を用いたが，クラスタリングおよび可視化の前処理として UMAP による次元圧縮を行った．UMAP には主に n_neighbors（近傍点

数）と min_dist（最小距離）という 2 つの重要なハイパーパラメータが存在する．本研究では，特許データ特有の構造を適切に捉えるため，以下の比較検証を行い，最終的なパラメータを決定した．

n_neighbors

各データ点の周辺構造をどれくらいの範囲まで考慮するかを制御する．

小さい値（例：5〜15）：局所的な構造が強調され，非常に細かいクラスタに分断される傾向があった．特許の場合，同一技術であっても表現の揺らぎがあるため，過度に細分化されると技術体系としてのまとまりが見えにくくなる問題が生じた．

大きい値（例：50〜100）：大域的な構造が保持される一方で，異なる技術分野の境界が曖昧になり，全体がひとつの大きな塊に見えてしまう傾向があった．

最適値（25）：検証の結果，類似した技術を一つのグループとして認識しつつ，異なる技術分野を明確に分離できるバランス点として n_neighbors = 25 を採用した．

min_dist

低次元空間においてデータ点をどれだけ密集させるかを制御する．

小さい値（例：0.01）：クラスタが極端に凝集し，クラスタ内部の構造（特許同士の微細な違い）がつぶれてしまう現象が見られた．

大きい値（例：0.5）：クラスタが拡散しすぎて，クラスタ間の境界が不明瞭になった．

最適値（0.1）：視認性とクラスタの分離性を両立させるため，min_dist = 0.1 を採用した．

metric

テキストデータの類似度計算には，ユークリッド距離よりも角度（方向）に基づく類似度が適しているとされる．比較の結果，metric = 'cosine'（コサイン類似度）を用いた場合が，最も意味的に近い特許を近傍に配置できることを確認したため，これを採用した．

クラスタリング（K-Medoids）のパラメータ決定プロセス

K-Medoids 法における最大のハイパーパラメータは，クラスタ数 k（n_clusters）である．本システムでは，k を固定値とせず，入力データに応じて動的に決定する適応的なアプローチを実装した．

- シルエット分析による自動決定 k の値を 3 から 20 まで変化させながら K-Medoids を試行し，それぞれの結果に対してシルエット係数の平均値を算出した．
- シルエット係数は，クラスタ内の凝集度とクラスタ間の乖離度を評価する指標であり，値が 1 に近いほど良好なクラスタリングであることを示す．

表 4.1: ハイパーパラメータ一覧

パラメータ名	変数名	設定値
近傍点数	n_neighbors	25
最小距離	min_dist	0.1
距離尺度	metric	cosine

- 実装コードでは，silhouette_samples を用いて全データの係数を計算し，その平均値 silhouette_avg を silhouette_df リストに記録している．
- ループ終了後，silhouette_df の中で最大値を示したインデックスに対応する k を最適クラスタ数 class_n として自動採用するロジックとした．これにより，データの分布特性に合わせて，常に最適な粒度での分類が可能となった．

スコアリングモデルの重み係数の調整結果

第 3 章で定義した重要度スコア算出式 $S_i = (\tilde{c}_i \times w_c + \tilde{l}_i \times w_l) \times \alpha + \beta$ における重み係数 w_c, w_l の選定についても，実データを用いた定性的な比較検証を行った．

- ケース A: 記述量重視 $(w_c = 0.3, w_l = 0.7)$: 明細書の長い特許が大きく表示されたが，これらは必ずしも権利範囲が広い基本特許とは限らず，単に実施例を羅列しただけの特許や，冗長な記述の特許が過大評価される傾向が見られた．
- ケース B：均等配分

  $(w_c = 0.5, w_l = 0.5)$: バランスは良いが，IP ランドスケープにおいて最も警戒すべき「強力な権利網を持つ特許」の強調が不十分であった．
- ケース C：請求項重視

  $(w_c = 0.7, w_l = 0.3)$: 請求項数が多い特許が顕著に大きく表示された．実際に内容を確認したところ，これらは基本特許や包括的な権利を主張する特許である割合が高かった．記述量はあくまで「内容の充実度」を示す補助的な指標とし，権利範囲の広さを優先するこの設定が，戦略的な分析において最も有用であると判断し，最終的にこのパラメータセットを採用した．

テストデータによる比較検証

システムの有効性を確認するため，異なる規模や分野のテストデータを用いて予備実験を行った．

- 小規模データ（約 100 件）処理時間は数秒程度と高速だが，データ数が少なすぎて UMAP の次元圧縮が不安定になり，クラスタ構造が不明瞭になるケースがあった．システム実装では，取得データ数が 30 件未満の場合は処理を中断し，再検索を促すガード処理（if desc_len ¡ 30:）を追加した．

- 大規模データ（約 2000 件以上）

  全体の見通しは良くなるが，ブラウザでの描画負荷が高まり，インタラクションが重くなる課題があった．これに対し，3D グラフ（Three.js）では表示するエッジ数を制限（上位 3000 件など）するフィルタリング機能を実装し，パフォーマンスと情報量のバランスを調整した．

これらの検証を通じて決定されたハイパーパラメータと実装上の工夫により，本システムは多様な検索クエリに対してロバストかつ実用的な分析結果を提供できる構成となっている．

§ 4.3　重要度を反映した可視化インターフェース

ユーザーによる分析操作プロセス

本システムは，ユーザーが直感的な操作で IP ランドスケープを実践できるよう，以下の 4 段階のプロセスで分析を行う設計となっている．

- 条件設定とデータ収集

  ユーザーはまずフロントページ（図 4.1）にアクセスし，分析対象となる「技術キーワード（例：ブロックチェーン 決済)」と，調査したい「対象期間（例：6 年)」を入力フォームに設定する.「分析を実行する」ボタンを押下すると，システムはバックグラウンドでマルチスレッドによる並列スクレイピングを開始し，指定された条件の特許データを網羅的に収集する．

- 全体俯瞰

  データ処理が完了すると，分析結果画面へ自動的に遷移する．ここには，特許群の技術的な分布を示す「IP ランドスケープマップ（散布図)」が表示される（図 4.2）．このマップ上では，第 3 章で定義した重要度スコアに基づいて，特許に対応するマーカーのサイズが動的に変化している．ユーザーは，まずマップ上でひと際大きく描画されている「巨大な点」に着目することで，その技術分野における重要特許がどのクラスタに存在するかを瞬時に把握する．

- 詳細分析対象の選定

  マップの下側には，各クラスタの内容を示唆する「自動生成タイトル」が凡例として表示されている (図 4.3)．ユーザーは，マップ上の分布（大きな点の位置）とタイトルを照らし合わせ，自社の戦略にとって重要と思われる技術領域を特定する．そして，そのクラスタ番号を画面下部の選択フォームに入力し，詳細分析を実行する．

- 構造理解

  クラスタを選択すると，詳細分析画面へ遷移する．ここでは，選択されたクラスタに含まれる特許リストと共に，技術用語間の関係性を可視化した「共起語ネットワーク図」が表示される．ユーザーは，インタラクティブなネットワーク図を操作しなが

図 4.1: システムのフロントページ

ら，どのような技術要素が結びついているかを解釈し，最終的に具体的な特許文献の読み込みへと進む．

この一連のフローにより，ユーザーは「全体俯瞰」から「重要箇所の特定」，そして「詳細理解」へと，スムーズかつ論理的に分析を深めることが可能となる．

Matplotlib と Seaborn による動的散布図の実装

本システムのユーザーインターフェースの中核となるのが，特許の技術分布と重要度を同時に可視化する「IP ランドスケープマップ」である．このマップ生成には，Python の描画ライブラリである Matplotlib および Seaborn を使用している．Web アプリケーションとして動的に画像を生成するため，以下の技術的な実装を行っている [22].

- バックエンドレンダリング Web サーバー上で GUI ウィンドウを表示することはできないため，Matplotlib のバックエンドとして Anti-Grain Geometry (Agg) を指定している．これにより，画像データをメモリ上のバッファに直接書き込み，ファイルシステムを経由せずに処理を完結させている．

- Base64 エンコーディング

  生成された画像バッファは base64 エンコードされ，HTML の ¡img¿ タグの src 属性にデータ URI スキームとして直接埋め込まれる．これにより，画像ファイルの管理コストを削減し，ステートレスな画像配信を実現している [23].

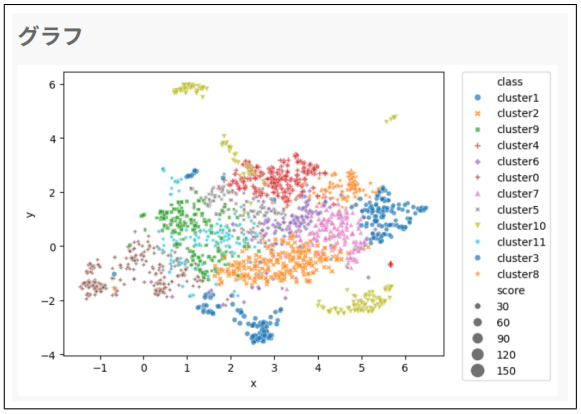

図 4.2: 散布図の様子

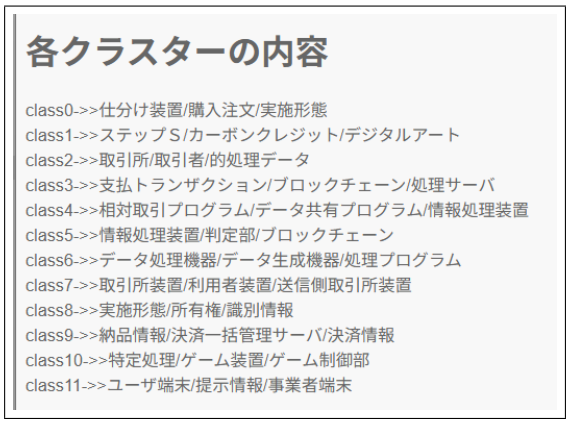

図 4.3: クラスター一覧

- 視覚的エンコーディングの実装詳細 sns.scatterplot 関数を用いて散布図を描画する際，データの持つ多次元情報を以下の視覚属性にマッピングしている．
  - 位置 (X, Y) UMAP によって 2 次元に圧縮されたベクトル座標．意味的に近い特許は近くに配置される．
  - 色 (Hue) K-Medoids によって分類されたクラスタラベル（class）．palette=”tab10” を指定し，明瞭な色分けを行っている．
  - 形状 (Style) 色覚多様性への配慮および識別性の向上のため，クラスタごとにマーカーの形状を変化させている．コード内では class_marker_dict を定義し，20 種類以上のマーカー記号を各クラスタに割り当てている．
  - サイズ (Size) 本研究の核心である「重要度スコア」をマーカーの大きさに反映させている．size=”score” 引数を指定し，sizes=(20, 200) で最小・最大サイズを制御することで，重要度の低い特許（20px）と高い特許（200px）の間に明確な視覚的差異を作り出している [24].

データの前処理

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 [26].

以上のインターフェース実装により，本システムは単なるデータ処理ツールを超えて，ユーザーの知的生産活動を視覚的かつ対話的に支援する「IP ランドスケープ・プラットフォーム」として機能するものである．

第5章

実験結果並びに考察

§ 5.1　実験の概要

本章では，第 4 章で構築した提案システムの有効性を検証するために実施した評価実験の概要とその結果について詳述する．本実験では，先行研究に相当するシステムと提案システムの実行結果を比較し，特に「重要特許の視認性」や「探索効率」といった観点から，ユーザーがどのように知覚するかをアンケート調査により定量的に評価した．また，特定の技術テーマ（水圧，フィルター，イオン）を用いた事例分析を通じて，システムの定性的な有効性を検証した．

本節では，本研究で実施した評価実験の目的，比較対象としたシステムの詳細，実験に用いたデータセットの選定理由，およびアンケート調査の設計について述べる．

まず実験の目的について述べる．IP ランドスケープにおいては，膨大な特許データの中から「事業戦略上，無視できない重要な特許」を迅速に発見し，その技術内容を深く理解することが求められる．しかし，従来の手法に実装されているような単純なベクトル化とクラスタリングのみを用いた手法では，可視化されたマップ上ですべての特許が均一な大きさで描画されるため，分析者はどの特許が重要であるかを視覚的に判別することが困難であった．そこで本実験では，本研究で提案する「重要度スコア（請求項数・記述量）に基づくノードサイズの可変表示」および「K-Medoids 法によるクラスタリング」がこの課題を解決し，ユーザーの探索行動をどのように変容させるかを明らかにすることを目的とした．

次に比較対象システムの設定について述べる．本実験では，同一の検索クエリとデータセットに対し，2 つの異なるアルゴリズムを適用した結果画像を比較対象とした．一つ目は比較手法（従来手法）である．これは先行研究である平井卒論のロジックを踏襲したものであり，Sentence-BERT によるベクトル化後，K-Means 法でクラスタリングを行うものである．可視化においては UMAP による 2 次元散布図を用いるが，すべてのノード（特許）は等しい大きさの点として描画され，色はクラスタごとに区別されるのみである．この手法では，特許の密集度合いからトレンドを把握することは可能であるが，個々の特許の質的価値はクリックして内容を確認するまで不明であるという特徴がある．二つ目は提案手法である．これは本研究で独自に開発・実装した手法であり，請求項数と明細書の文字数から算出したスコアを用いる．クラスタリングには K-Medoids 法を採用し，算出された重要度スコアをノードのサイズ（半径）にマッピングして描画する．重要な特許は巨大な点として，そうでない特許は小さな点として表示されるため，マップを見た瞬間に「見るべき特許」が視覚的に強調され，探索の優先順位が明確になることが期待される．

表 5.1: アンケート内容

重要な特許の視認性（見つけやすさ）	情報の優先順位の明確さ
探索にかかる手間の削減（効率性）	直感的な理解のしやすさ
全体の技術トレンドの把握	視覚的なインパクト
視覚的ストレスの少なさ	外れ値（ノイズ）の識別
探索意欲の向上	総合的な満足度

アンケート調査の設計について述べる．システムのユーザビリティと可視化効果を客観的に評価するため，20 歳から 22 歳の大学生 28 名を被験者としてアンケートを実施した．被験者は学部生であり，特許分析の実務経験はない初学者層である．IP ランドスケープの専門家ではないユーザーを対象とすることで，事前知識に依存しない「直感的な分かりやすさ」や UI/UX を純粋に評価する狙いがある．実施手順としては，まず被験者に「あなたは企業の企画担当者で，この分野の重要な技術（コア技術）を探している」というシナリオを提示した．次に，3 つのキーワードそれぞれについて，従来手法の画像と提案手法の画像を交互に提示し，評価を求めた．評価項目は以下の 10 項目を設定した．第一に「重要な特許の視認性（見つけやすさ）」であり，パッと見て重要なものがどこにあるかわかるかを問うものである．第二に「情報の優先順位の明確さ」であり，どれから先に見るべきかの順序がつけやすいかを問うものである．第三に「探索にかかる手間の削減（効率性）」であり，無駄な確認作業が減りそうだと感じるかを問うものである．第四に「直感的な理解のしやすさ」であり，説明を受けなくても図の意味がわかるかを問うものである．第五に「全体の技術トレンドの把握」であり，どのような技術群があるかというクラスタ構造が理解しやすいかを問うものである．第六に「視覚的なインパクト」であり，図としての説得力や印象の強さを問うものである．第七に「視覚的ストレスの少なさ」であり，情報が過多でなく見ていて疲れないかを問うものである．第八に「外れ値（ノイズ）の識別」であり，重要でないものと重要なものの区別がつくかを問うものである．第九に「探索意欲の向上」であり，もっと詳しく調べてみたいという興味が湧くかを問うものである．第十に「総合的な満足度」であり，システム全体としての評価を問うものである．回答については,「まったく満足していない（そう思わない）」,「あまり満足していない」,「どちらでもない」,「やや満足している」,「非常に満足している（そう思う）」の 5 段階評価のリッカート尺度を採用した．

§ 5.2　実験結果と考察

最後にアンケート調査における結果と考察を行う．

一個目に,「重要な特許の視認性（見つけやすさ）」という質問を行った．結果として,『非常に満足している』が 32.1%,『やや満足している』が 42.9%となり，全体的に好印象な評価を得ることができた．この結果から，提案システムの核となる「重要度に応じたサイズ変更」が，ユーザーの視覚的なターゲット特定能力をどれだけ物理的に向上させたかが分かる．

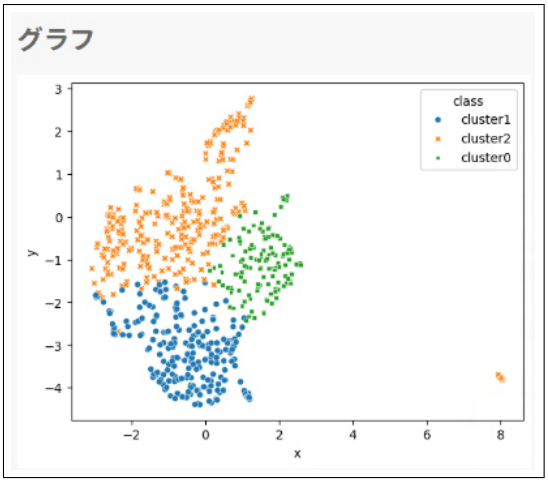

図 5.1: 先行研究出力結果

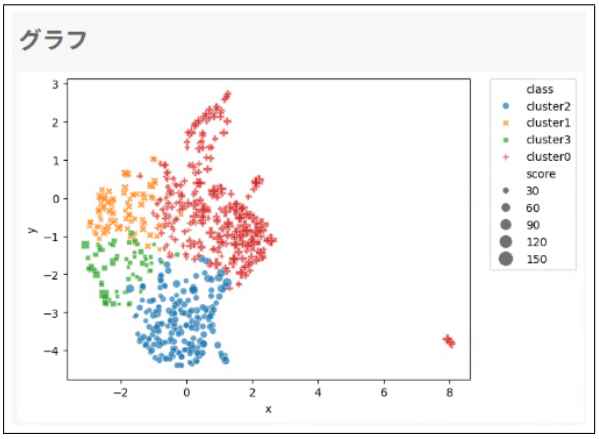

図 5.2: 本研究で出力された結果

二個目に,「情報の優先順位の明確さ」という質問を行った．結果として,『非常に満足している』が 46.4%,『やや満足している』が 35.7%となり，全体的に好印象な評価を得ることができた．この結果から，ノードの大小という視覚的階層構造が，探索における「見るべき順序」を直感的に示唆し，ユーザーの意思決定プロセスをどれだけ円滑にしたかが分かる．

三個目に,「探索にかかる手間の削減（効率性）」という質問を行った．結果として,『非常に満足している』が 28.6%,『やや満足している』が 39.3%となり，全体的に好印象な評価を得ることができた．この結果から，重要特許の視覚的な絞り込み機能が，無作為なクリックによる確認作業を排除し，分析に要する工数をどれだけ物理的に削減させたかが分かる．

四個目に,「直感的な理解のしやすさ」という質問を行った．結果として,『非常に満足している』が 46.4%,『やや満足している』が 28.6%となり，全体的に好印象な評価を得ることができた．この結果から,「大きい＝重要」というシンプルな視覚メタファーが，専門知識を持たないユーザーの学習コストを下げ，システムの受容性をどれだけ向上させたかが分かる．

五個目に,「全体の技術トレンドの把握」という質問を行った．結果として,『非常に満足している』が 17.9%,『やや満足している』が 39.3%である一方,『あまり満足していない』が 14.3%,『全く満足していない』が 10.7%となり，全体的にあまり好印象な結果を得ることができなかった．この結果から，提案手法の可視化においてはトレンドの全体像把握に難しさを感じたユーザーが一定数いたことが考えられる．その理由として，スコアリングによって強調された大きなノードが視覚的に支配的になりすぎたことが挙げられる．大きな点が前面に出ることで，背後にあるクラスタの形状や点の密集度合い（分布の全体像）が遮られ，マクロな視点での構造理解が阻害されたと考得られる．この解決策として，ズームレベルに応じてノードサイズを動的に調整する機能や，重要度表示と分布表示を切り替えるレイヤー機能の導入などが挙げられる．さらに，重なりを回避するレイアウトアルゴリズムを採用することで，個々の重要性と全体の分布を両立させた可視化が可能になると考える．

六個目に,「視覚的なインパクト」という質問を行った．結果として,『非常に満足している』が 46.4%,『やや満足している』が 39.3%となり，全体的に好印象な評価を得ることができた．この結果から，大小のノードがリズミカルに配置されたマップデザインが，分析結果としての説得力を高め，プレゼンテーションツールとしての価値をどれだけ向上させた

表 5.2: アンケート結果

	5	4	3	2	1
重要な特許の視認性	32.1	42.9	3.6	10.7	10.7
情報の優先順位の明確さ	46.4	35.7	3.6	7.1	7.1
探索にかかる手間の削減	28.6	39.3	10.7	7.1	14.3
直感的な理解のしやすさ	46.4	28.6	3.6	10.7	10.7
全体の技術トレンドの把握	17.9	39.3	17.9	14.3	10.7
視覚的なインパクト	46.4	39.3	0	7.1	7.1
視覚的ストレスの少なさ	28.6	35.7	7.1	17.9	10.7
外れ値（ノイズ）の識別	42.9	28.6	10.7	10.7	7.1
探索意欲の向上	14.3	57.1	14.3	3.6	10.7
総合的な満足度	42.9	39.3	0	10.7	7.1

かが分かる．

七個目に,「視覚的ストレスの少なさ」という質問を行った．結果として,『非常に満足している』が 28.6%,『やや満足している』が 35.7%である一方,『あまり満足していない』が 17.9%,『全く満足していない』が 10.7%となり，全体的にあまり好印象な結果を得ることができなかった．この結果から，システムの利用においては視覚的な圧迫感やストレスを感じるユーザーが一定数いたことが考えられる．その理由として，情報の過密化とオーバープロット（重なり）が挙げられる．重要度に応じてサイズを大きくした結果，データが密集している領域ではノード同士が重なり合い，視認性が低下するとともに画面全体が「ごちゃごちゃしている」という印象を与えたと考得られる．この解決策として，透過度（Alpha 値）の更なる調整や，密集度に応じて表示するノード数を間引くフィルタリング機能の実装などが挙げられる．また，クラスタの中心から離れたノードを小さく表示するなどの視覚的整理を行うことで，情報量は維持しつつ心理的ストレスの少ないインターフェースを実現できると考える．

八個目に,「外れ値（ノイズ）の識別」という質問を行った．結果として,『非常に満足している』が 42.9%,『やや満足している』が 28.6%となり，全体的に好印象な評価を得ることができた．この結果から，重要度の低い特許を最小サイズで背景化する処理が，視覚的なフィルタリングとして機能し，ノイズの選別能力をどれだけ物理的に向上させたかが分かる．

九個目に,「探索意欲の向上」という質問を行った．結果として,『非常に満足している』が 14.3%,『やや満足している』が 57.1%となり，全体的に好印象な評価を得ることができた．この結果から，重要特許が地図上のランドマークのように可視化されたことが，ユーザーの知的好奇心を刺激し，能動的な探索行動をどれだけ促進させたかが分かる．

十個目に,「総合的な満足度」という質問を行った．結果として,『非常に満足している』が 42.9%,『やや満足している』が 39.3%となり，全体的に好印象な評価を得ることができた．この結果から，特許情報の「質」と「量」を統合的に可視化するという本研究のアプローチが，IP ランドスケープ支援におけるユーザー体験（UX）をどれだけ総合的に向上さ

せたかが分かる.

本研究で提案した「重要度スコアに基づく可視化」は, 特許情報の探索効率を飛躍的に向上させることがアンケート結果より明らかとなった. 特に,「視認性」や「優先順位の明確さ」において従来手法を大きく上回る評価を得たことは, 初学者でも直感的に重要特許を識別できるシステムの実用性を示している. 一方で,「全体の技術トレンドの把握」や「視覚的ストレス」に関しては課題が残った. これは, スコアリングによる強調表示が, 情報の過密化やオーバープロットを引き起こし, マクロな視点での構造理解を阻害したためと考えられる. 今後は, ズーム機能の強化や, 重要度と分布表示を動的に切り替えるレイヤー機能の実装など, 情報の解像度を制御するインターフェースの改善が, より高度な IP ランドスケープ支援には不可欠であると結論付ける.

第6章

おわりに

本研究では，デジタルトランスフォーメーション（DX）の加速やグローバル市場における競争激化を背景に，企業経営の意思決定を支える「IP ランドスケープ」を高度化するためのシステム開発を行った．特許情報は，技術的な権利範囲や詳細な実施形態を含む経営戦略上の極めて重要なリソースである．しかし，世界的な特許出願件数の爆発的な増加により，その膨大なデータを人手で精査・分析することは物理的な限界を迎えている．先行研究をはじめとする既存の分析システムでは，自然言語処理技術を用いた特許のベクトル化やクラスタリングにより，情報の「整理」や「分類」は実現されたものの，可視化されたマップ上ですべての特許が均一な大きさの点として描画されるという課題が残されていた．このため，真に注目すべき「重要特許」が有象無象のノイズの中に埋没してしまい，分析者は有益な情報を探索するために多大な労力を費やす必要があった．

本研究では，この課題に対する根本的な解決策として，特許情報の「量」だけでなく「質」の評価を可視化に統合する新たなアプローチを提案し，以下の 3 つの主要な成果を得た．第一に，特許の権利範囲を画定する「請求項数」と，技術開示の詳細さを示す「明細書の記述量」に基づく独自の重要度スコアリングモデルを構築したことである．このスコアを可視化マップ上のノードサイズに動的に反映させることで，権利範囲が広く技術密度が高い特許を「巨大な点」として視覚的に強調することに成功した．評価実験の結果，この手法は従来手法と比較して「重要特許の視認性」を劇的に向上させ，専門知識を持たない初学者であっても直感的にコア技術を特定できることが実証された．第二に，クラスタリング手法として K-Medoids 法を採用したことである．従来の K-Means 法が計算上の重心を扱うのに対し，本手法では実在する特許を代表点（Medoid）とすることで，外れ値への頑健性を高めるとともに，クラスタの意味解釈を容易にした．これにより，ブラックボックスになりがちな機械学習の結果に対し，透明性と納得感を与えることができた．第三に，マルチスレッドによる並列処理技術の実装である．ボトルネックとなっていたスクレイピング処理を並列化することでデータ収集時間を大幅に短縮し，実務においてユーザーが思考を中断することなく仮説検証サイクルを回せる，実用的な応答速度を達成した．

一方で，本システムの実用化と更なる高度化に向けては，解決すべき課題も明らかとなった．最も大きな課題は，情報の強調表示に伴う視覚的な過密化（オーバープロット）の解消である．重要度に応じてノードを拡大した結果，データ密集部ではノード同士が重なり合い，マクロな視点でのトレンド把握や背後にある情報の視認性が阻害される傾向が見られた．これに対しては，Google Maps のようにズームレベルに応じて表示情報の粒度を制御する Semantic Zooming の実装や，重なりを回避しつつ位置関係を保持する高度なレイアウトアルゴリズムの導入が必要である．また，スコアリング指標の多様化も重要な検討

事項である．現在は速報性を重視して特許文献の内部情報のみを用いているが，特許の経済的価値をより正確に測るためには，被引用数やパテントファミリー数，権利維持期間といった外部指標を API 連携等で取り込み，ユーザーの分析目的に応じて重み付けをカスタマイズできる機能が求められる．さらに，システム基盤に関しては，より大規模なデータセットや高度な自然言語処理に対応するため，GPU の活用や分散処理環境への移行による処理能力の向上が不可欠である．

結論として，本研究は特許情報の「質」を可視化するという新たな視点を提示し，IP ランドスケープにおける探索効率と意思決定の質を飛躍的に向上させる可能性を示した．本システムが，膨大な技術情報の海から新たな知見を創出する羅針盤となり，企業のイノベーション活動に貢献することを強く期待する．

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情報工学部データサイエンス学科システム数理学講座の António Oliveira Nzinga René 准教授，　新潟国際情報大学の奥原浩之教授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．

2026 年 2 月

氷見　夏輝

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 2026-01-20, https://www.nec-solutioninnovators.co.jp/sp/contents/column/20230623_vuca.html.

[2] 株式会社三菱総合研究所, “第 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 2026-01-20, https://www.soumu.go.jp/johotsusintokei/linkdata/h29_03_houkoku.pdf.

[3] WPIO, “世界知的財産指標報告書”, 閲覧日 2026-01-20, https://www.wipo.int/pressroom/ja/articles/2023/article_0013.html.

[4] 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 2026-01-20, https://www.tokyo-kosha.or.jp/chizai/manual/senryaku/rmepal000001vypy-att/senryaku_all_vol.9.pdf.

[5] 特許庁, “経営戦略を成功に導く知財戦略”, 閲覧日 2026-01-20, https://www.jpo.go.jp/support/example/document/chizai_senryaku_2020/all.pdf.

[6] 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 2026-01-20, https://financenavi.jp/basic-knowledge/management_strategy_framework/#tag1.

[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 2026-01-20, https://www.jpo.go.jp/support/general/chizai-jobobunseki-report.html.

[8] AGIRobots Blog, “【Transformer の基礎】Multi-Head Attention の仕組み”, 閲覧日 2026-01-20, https://developers.agirobots.com/jp/multi-head-attention/.

[9] N.Reimers，I.Gurevych．“Sentence-BERT：Sentence Embedding using Siamese BERT-Networks”, *ArXiv e-prints*, 1908. 10084, 2019

[10] data-analytics.fun,“【論文解説】Sentence-BERT を理解する”, 閲覧日 2026-01-20, https://data-analytics.fun/2020/08/04/understanding-sentence-bert/.

[11] Hugging Face, “sonoisa/sentence-bert-base-ja-mean-tokens”, 閲覧日 2026-1-20, https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens.

[12] 東京大学 松本研究室, “専門用語自動抽出システム TermExtract”, 閲覧日 2026-1-20,http://gensen.dl.itc.u-tokyo.ac.jp/.

[13] L.Mclnnes， J.Healy，J.Melville, “UMAP：Uniform Manifold Approximation and Projection for Dimension Reduction”, *ArXiv e-prints*, 1802. 03426, 2018

[14] Reinforz Insight, “UMAP の深堀：パラメータ解説から最新の動向まで”, 閲覧日 2026-01-20,
https://reinforz.co.jp/bizmedia/11257/.

[15] 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.

[16] L.Kaufman, P.J.Rousseeuw, “Finding Groups in Data: An Introduction to Cluster Analysis”, *John Wiley Sons*, 1990,
https://doi.org/10.1002/9780470316801.

[17] e-Gov 法令検索, “特許法”, 閲覧日 2026-1-20,
https://www.google.com/search?q=https://elaws.e-gov.go.jp/document/3Flawid/3D334AC000000

[18] Technical Note, “シルエット分析”, 閲覧日 2026-01-20,
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 2026-01-20,
https://mieruca-ai.com/ai/jaccard_dice_simpson/.

[20] Flask Documentation, “Flask: A microframework for Python”, 閲覧日 2026-1-20,
https://flask.palletsprojects.com/.

[21] Selenium Documentation, “Selenium with Python”, 閲覧日 2026-1-20,
https://selenium-python.readthedocs.io/.

[22] Seaborn Documentation, “seaborn.scatterplot”, 閲覧日 2026-1-20,
https://seaborn.pydata.org/generated/seaborn.scatterplot.html.

[23] 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 2026-01-20,
https://and-engineer.com/articles/ZOWitBIAACMAFtEj.

[24] Three.js Documentation, “Three.js – JavaScript 3D Library”, 閲覧日 2026-1-20,
https://threejs.org/.

[25] 鈴木 純, “pyvis でネットワークグラフをインタラクティブな html に出力してみた”, 閲覧日 2026-01-20,
https://dev.classmethod.jp/articles/python-pyvis-interactive-network-graph-html-output/.

[26] PyVis Documentation, “Interactive network visualization in Python”, 閲覧日 2026-1-20,
https://pyvis.readthedocs.io/.