# JAPAS: 日本語特許サポート要件抽出のためのベンチマーク

帖佐 克己　杉浦 亮介
NTT, Inc.
{katsuki.chousa,ryosuke.sugiura}@ntt.com

## 概要

特許文献の効率的な分析は、技術開発や知的財産保護において不可欠である。特許分析における重要なタスクの一つに、明細書が請求の範囲を十分に説明しているか（サポート要件）の検証がある。この検証は高度なドメイン知識と多大な労力を要するため自動化が望まれているが、(1) 公開されたベンチマークの不在、(2) 既存研究が語彙マッチングに依存し意味的等価性を捉えられない、という 2 つの課題が存在した。本研究では、これらを解決するために、日本語特許を対象に 2,000 件以上の手動アノテーションを付与したデータセット「JAPAS」を構築し、テキスト埋め込みや大規模言語モデルを用いたベースラインを確立した。

## 1　はじめに

特許文書は最新技術に関する情報源であり、その分析は技術開発および知的財産保護にとって不可欠である。特許出願書類は、特許請求の範囲（請求項）と発明の詳細な説明（明細書）から構成される。請求項は権利保護の法的範囲を定義し、明細書は発明に関する詳細かつ網羅的な情報を提供する。国内外の特許法において[1)]、請求の範囲は明細書によって十分に説明されていなければならない（サポート要件）と規定されている。すなわち、明細書は、当業者が請求項に記載された発明とその動作を理解できるような記述を含んでいる必要がある。この要件を満たさない場合、特許出願が拒絶されたり、後に無効とされたりする可能性がある。図 1 にサポート関係の例を示す。このことから、すべての請求項が明細書によって適切にサポートされているかどうかを検証する必要があるが、この検証には高度なドメイン知識と多大な労力を要するため、作業の自動化が求められている。

1）　例えば、日本特許法第 36 条、米国特許法第 112 条、欧州特許条約第 84 条、特許協力条約（PCT）第 6 条

この課題に対し、これまでいくつかの自動サポート関係抽出手法が提案されてきた [1, 2]。しかし、これらの手法は語彙の一致度合いに依存しており、請求項と明細書で異なる語彙が使用されている場合、サポート関係を見逃してしまう可能性があった。また、より根本的な問題として、この分野の研究を推進するための公開ベンチマークが存在しないという課題が存在した。

そこで本研究では、特許庁（JPO）の出願文書に対し、専門家の監修のもと 2,056 件のサポート関係を手動でアノテーションしたデータセット「JAPAS」を構築する。また、今後の研究のための強力なベースラインを確立するために、テキスト埋め込みや大規模言語モデル（LLM）を用いた意味的類似度を利用した新たな抽出モデルを提案する。実験の結果、JAPAS でファインチューニングした LLM は、語彙ベースのベースラインと比較して F1 スコアで+0.25 ポイントの向上を達成した。この結果は、JAPAS がこの複雑なタスクに取り組む上で効果的かつ高品質な基盤であることを示唆している。

## 2　関連研究

これまで、いくつかの大規模な特許コーパスが公開されている [3, 4, 5, 6, 7]。これらは特許内の各テキストセグメント（請求項、明細書など）へのアノテーションを含んでいるものの、サポート関係のようなセクション間の関係に関する情報は欠如していた。

サポート関係抽出の先行研究は、主に請求項と明細書段落のアライメントに焦点を当ててきた。Murata ら [1] は `diff` コマンドを用いたグローバルアライメントを、Shinmori ら [2] は談話構造に基づくローカルアライメントをそれぞれ提案した。しかし、これらの手法は語彙マッチングに依存しており、言い換えを含む表現に対しては脆弱である。さらに、既存研究は定量的評価を欠いているか、非公開データセットを使用しており、公開ベンチマーク

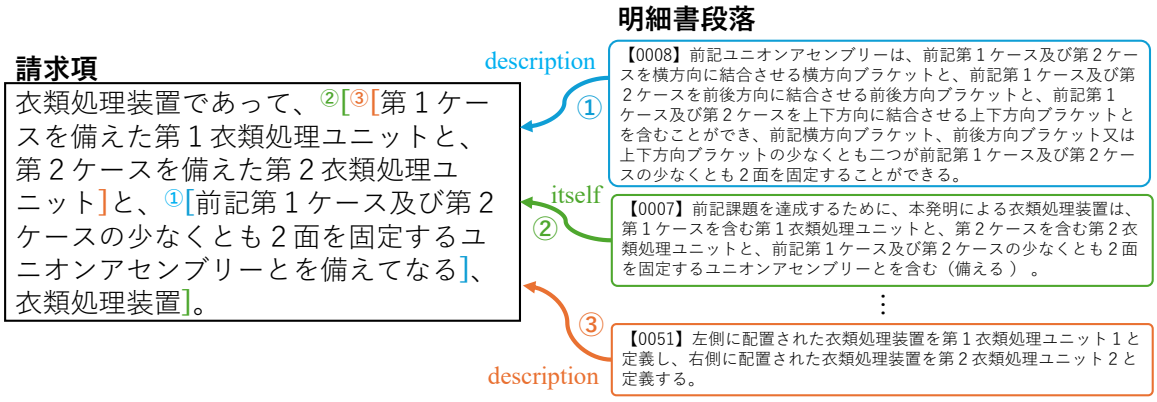


**図 1**　特許文書におけるサポート関係の例。請求項内の特定の語句（スパン）が、明細書の特定の段落によってサポートされている。

の不在が本タスクの進展を妨げてきた。

近年の NLP 分野では、語彙マッチングの限界を克服するために、テキスト埋め込みや LLM による意味的類似性が活用されている。E5 [8] や Ruri [9] などのモデルが高い性能を示しているが、サポート関係抽出における有効性は未検証である。したがって、ベンチマークを確立し、これらの意味的類似度に基づくアプローチを評価することは、本分野の発展において重要なステップである。

# 3　データセット構築

## 3.1　リソース

データセットのリソースには特許庁が提供する公開特許公報を用い、2019 年に公開された PCT 出願からランダムに 62 件を選出した。次に、各特許について、提供された XML ファイルから「発明の詳細な説明」と「特許請求の範囲」のセクションを抽出した。予算の制約上、アノテーションの対象は各特許の最初の 4 つの請求項に限定した。これは、前半の請求項が通常、最も広範で基本的な発明概念を定義しているためである。

## 3.2　アノテーションスキーム

本研究では、請求項内のテキストスパンと明細書段落との間のサポート関係を定義し、アノテーションを行うことでデータセットを構築した。アノテーションのガイドラインは特許専門家との議論を通じて決定した。各インスタンスは以下の要素を持つ。

- **Claim Span**: 請求項内の連続するテキストスパン。通常は述語を含む節単位。
- **Description ID**: サポートする明細書の段落番号。
- **Relation Type**: 以下の 3 分類。
  - `itself`: 請求項の文言がそのまま、あるいは軽微な言い換えで記載されている。
  - `description`: 定義や詳細な説明がなされている。
  - `example`: 実施例や変形例が示されている。
- **Confidence**: アノテータの確信度。80%以上の場合は `high`、50-79%の場合は `medium`、それ以下の場合は `low` とした。

なお、単一の Claim Span が複数の説明段落によってサポートされる場合があり、その場合は一対多の対応関係となる。

## 3.3　アノテーション手順

アノテーションは特許文書の読解経験が豊富な作業者によって実施した。はじめに、作業者は請求項を精読し、裏付けが必要な節（clause）レベルのテキストスパンを特定する。次に、特定されたスパンをクエリとして明細書全体を探索し、サポートとなる段落を特定する。最後に、特定された請求項スパンと明細書段落のペアに対して、3 つの関係タイプ（Relation Type）のいずれかを付与し、その際の確信度（Confidence）を記録する。ほぼ全ての請求項スパンが少なくとも 1 つの明細書段落に紐づくまで、このプロセスを繰り返した。アノテーション完了後、データセットを概ね 4:1:1 の割合で訓練・開発・評価セットに分割した。

表 1 と表 2 に、構築されたデータセットの統計と、データセット内の各ラベルの割合を示す。学習セットには 1,335 件のアノテーションが含まれており、LLM をファインチューニングするには十分な量を有している。また、請求項あたりのサポート関

表 1　データセットの統計

| 項目 | 訓練 | 開発 | 評価 |
|---|---|---|---|
| 特許数 | 40 | 11 | 11 |
| 請求項数 / 特許 | 4 | 4 | 4 |
| 明細書段落 / 特許 | 108.6 | 102.8 | 111.4 |
| サポート数 | 1335 | 347 | 406 |
| サポート数 / 請求項 | 8.34 | 7.89 | 9.23 |

表 2　関係タイプと確信度の分布（%）

| Relation | Prop. | Confidence | Prop. |
|---|---|---|---|
| itself | 10.4 | high | 54.9 |
| description | 66.4 | medium | 30.2 |
| example | 23.2 | low | 14.9 |

係数は平均して 8 件程度となった。一般的な特許の請求項が平均 11 件であり [10]、明細書段落がおおよそ 100 程度であることを考えると、このアノテーション数は妥当であると考えられる。

# 4　提案手法

請求項は、発明の異なる要素を記述する複数の節から構成されており、これらの要素に対する証拠は、明細書中の様々な段落に散在していることが多い。したがって、本タスクを、明細書の各段落をそれがサポートする特定の請求項スパンに対応付ける問題として定式化した。この問題に取り組むために、事前学習済みモデルを活用した、テキスト埋め込みベースの手法と LLM ベースの手法を提案し評価する。

## 4.1　埋め込みベース手法

本研究では、請求項スパンと明細書段落の埋め込みベクトル間のコサイン類似度を計算し、閾値を超えた場合にサポート関係ありと判定する方法を提案する。請求項のスパンの単位としては、請求項全体（`SentEmb`）と文節（`ClauseEmb`）[2] の 2 種類を検証した。さらに文節単位の場合、文脈を考慮せずに対象スパンだけをエンコードする方法（`Context-free`）、文脈を考慮するために全体をエンコードしてからスパンだけをプーリングする方法（`Contextual`）の 2 つを比較した。

## 4.2　LLM ベース手法

本タスクで大規模言語モデル（LLM）を活用するために、構造化されたプロンプトテンプレートを使用する。プロンプトは、システム指示（System Instructions）とユーザー入力（User Input）から構成される。システム指示ではタスクの概要と必要な出力フォーマットを指定し、ユーザー入力では請求項と明細書段落を提供する。プロンプトは予備実験に基づいて日本語で作成された。

LLM には、抽出された請求項スパンのテキスト、サポートタイプ、および確信度を含む JSON 形式の文字列を生成するように指示した。なお、モデルは文字やトークンのオフセットではなく、スパンの実際のテキストを生成する。また、サポート関係が見つからない場合、モデルには空のリストを返すように指示した。

本手法の評価にあたり、0-shot および 3-shots での予測と、JAPAS によってファインチューニングされたモデルでの予測の 3 つの設定を比較検討した。データセットのサイズが比較的小さいため、モデルの学習の際には Low-Rank Adaptation (LoRA) [11] を適用する。

# 5　実験

本論文では、最初に提案手法を主要タスクであるサポート関係抽出において評価した。次に、最も性能の高かったモデルについて、関係タイプおよび確信度の分類能力を評価した。

## 5.1　サポート関係抽出

### 5.1.1　実験設定

埋め込みベースの手法には、日本語に特化したモデルである `Ruri-v3-310m` [9] を使用した。プーリングにはモデルのデフォルト設定である `mean pooling` を使用した。閾値は、開発セットで請求項単位での評価指標が最大となる値を使用した。

LLM には Qwen3-14B[3] を使用した。ファインチューニングには `Unsloth`[4] を、推論には `vLLM` [12] を使用した。

表層レベルの情報のみを使用するベースラインとして、文レベルの TF-IDF ベクトル間のコサイン類似度を使用した。TF-IDF ベクトルは各特許について個別に計算し、その特許のすべての請求項と明細書を単一の文書として扱って語彙を構築した。先行研究では他の表層ベースの手法も提案されているが、実装が公開されておらず再現が困難であるた

2）　文節はテキストを読点で区切ることで獲得した。

3）　https://huggingface.co/Qwen/Qwen3-14B
4）　https://unsloth.ai/

表 3　サポート関係抽出の精度比較

| Systems | Claim-level Prec. / Rec. / F1 | Span-level F1 inc. NS | Span-level F1 exc. NS |
|---|---|---|---|
| TF-IDF | .39 / .19 / .25 | .91 | .16 |
| テキスト埋め込み - Ruri | | | |
| 請求項全体 | .41 / .35 / .38 | .91 | .26 |
| 文脈なし文節 | .30 / .37 / .33 | .87 | .24 |
| 文脈あり文節 | .45 / .34 / .39 | .91 | .25 |
| LLM - Qwen3-14B | | | |
| 0-shot | .16 / .53 / .26 | .74 | .24 |
| 3-shots | .16 / .53 / .26 | .74 | .24 |
| fine-tuning | .51 / .48 / .50 | .91 | .37 |

め、本研究では TF-IDF を採用した。

手法の性能は、請求項レベル（Claim-level）とスパンレベル（Span-level）の 2 つの粒度で評価した。請求項レベルの評価は請求項に対する正しいサポート段落を特定するモデルの基本的な能力を評価するものであり、ここでは系列アライメントタスクで広く採用されている F1 スコア [13] を採用した。スパンレベルの評価は抽出されたテキストスパンの正確さをトークン単位で測る指標であり、SQuAD v2 [14] などのスパン抽出タスクで使用される、マイクロ平均トークンレベル F1 スコアを使用した [5]。スパンレベルの評価では、正解が「サポートなし（No Support）」となる事例を評価対象に含める場合（inc. NS）と除外する場合（exc. NS）の双方について評価を行った。これにより、非サポート関係を正しく棄却する能力と、正例における抽出精度のトレードオフを包括的に評価した。

#### 5.1.2　結果

実験結果を表 3 に示す。最も高い性能を示したのは Qwen3-14B Fine-tuning であり、Claim-level F1 で 0.50 を達成した。これは TF-IDF やテキスト埋め込み手法を大きく上回る。また、Span-level F1 においても inc. NS を高く保ちつつ、exc. NS でも最も高い精度を達成した。これは、ファインチューニングによって、モデルが特許特有の表現とサポート関係を学習したことを示唆している。一方、0-shot や 4-shots の LLM は適合率が低く、過剰検出する傾向が見られた。

5）トークナイザには fugashi [15] を使用し、実装は日本語 SQuAD の評価スクリプトに準拠した。Stability-AI/lm-evaluation-harness: https://github.com/Stability-AI/lm-evaluation-harness/blob/jp-stable/lm_eval/jasquad/evaluate.py

表 4　関係タイプ分類の精度

| ラベル | 正解 / 合計 | 正解率 (%) |
|---|---|---|
| itself | 22 / 24 | 91.7 |
| description | 212 / 278 | 76.3 |
| example | 66 / 104 | 63.5 |
| **合計** | **300 / 406** | **73.9** |

### 5.2　サポート関係タイプ分類

次に、モデルの関係タイプの分類性能を評価するため、サポート関係が正しく特定されていると仮定した Oracle 設定の下での正解率を評価した。具体的には、テストセットから得られた 406 件の請求項スパンと明細書段落のペアを用いて、fine-tuning した Qwen3-14B に関係タイプのみを予測させた。正解のペアは、forced decoding を使用してモデルに与えた。評価指標には正解率を使用した。

結果を表 4 に示す。ラベル別での内訳では、語彙的な重複度が高い itself が 91.7%と最も高い正解率を達成した。対照的に、発明の実施形態や変形例を記述する example は分類が最も困難なカテゴリであり、正解率は 63.5%であった。この難しさは example のトピックやスタイルの多様性、および訓練データの少なさに起因している可能性がある。さらに、description と example の違いはしばしば抽象度の問題であり、このようなタスク固有の曖昧さが原因であることも考えられる。

## 6　おわりに

本研究では、特許の請求項と明細書との間のサポート関係を抽出するタスクに取り組み、サポート関係を手動でアノテーションしたデータセット「JAPAS」を構築した。また、強力なベースラインを確立するために、テキスト埋め込みや LLM を用いた意味的類似性を活用する新しい手法を提案・評価した。実験の結果、JAPAS でファインチューニングされた LLM が請求項レベルの F1 スコアで 0.50 ポイントを達成し、ベースラインに対して+0.25 ポイントの改善を示した。この結果は、JAPAS がこの複雑なタスクに取り組む上で効果的かつ高品質な基盤であることを裏付けると同時に、このタスクが実現可能でありながらも依然として困難であることを示唆している。

# 参考文献


[1] Masaki Murata and Hitoshi Isahara. Using the diff command in patent documents. In **Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering**, 2002.

[2] Akihiro Shinmori, Manabu Okumura, and Y Marukawa. Aligning patent claims with detailed descriptions for readability. In **Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization**, 2004.

[3] Makoto Iwayama, Atsushi Fujii, Noriko Kando, and Akihiko Takano. Overview of patent retrieval task at NTCIR-3. In **Proceedings of the ACL-2003 Workshop on Patent Corpus Processing**, pp. 24–32. Association for Computational Linguistics, July 2003.

[4] Shaobo Li, Jie Hu, Yuxin Cui, and Jianjun Hu. Deeppatent: patent classification with convolutional neural networks and word embedding. **Scientometrics**, Vol. 117, No. 2, pp. 721–744, 2018.

[5] Eva Sharma, Chen Li, and Lu Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics.

[6] Mirac Suzgun, Luke Melas-Kyriazi, Suproteem K Sarkar, Scott Kominers, and Stuart Shieber. The harvard USPTO patent dataset: A large-scale, well-structured, and multipurpose corpus of patent applications. In **Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track**, 2023.

[7] Florina Piroi, Mihai Lupu, Allan Hanbury, and Veronika Zenz. Clef-ip 2011: Retrieval in the intellectual property domain. In **CLEF (notebook papers/labs/workshop)**, 2011.

[8] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024.

[9] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese general text embeddings, 2024.

[10] IP5 Offices. IP5 Statistics Report 2020 Edition, 2020. Accessed: 2025-09-24.

[11] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In **International Conference on Learning Representations**, 2022.

[12] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In **Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles**, 2023.

[13] Brian Thompson and Philipp Koehn. Vecalign: Improved sentence alignment in linear time and space. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 1342–1348, Hong Kong, China, November 2019. Association for Computational Linguistics.

[14] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors, **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[15] Paul McCann. fugashi, a tool for tokenizing Japanese in python. In Eunjeong L. Park, Masato Hagiwara, Dmitrijs Milajevs, Nelson F. Liu, Geeticka Chauhan, and Liling Tan, editors, **Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)**, pp. 44–51, Online, November 2020. Association for Computational Linguistics.