情報処理学会論文誌　デジタルプラクティス　Vol.4　No.3　58–68 (July 2023)

特許文献による BERT 事前学習モデルと特許調査業務への応用

秋山 賢二[1,a)]　斎藤 隆文[1]

受付日 2022年2月21日, 採録日 2022年11月10日

概要：数多くの文献から目的にあった文献を効率よく仕分けすることは，様々な分野で求められている．近年の特許検索データベースは，クエリ文書との類似性により検索された特許文書をランキング表示することで，仕分けをサポートする機能を提供しているケースもある．しかし，特許の侵害回避調査では，調査対象製品との関係性で特許文書を仕分けする必要がある．製品に関する知識のほとんどは開発者の頭の中にあるため，仕分け作業はもっぱら人手に頼っていた．本研究では，あらかじめ指定した検索条件で文献を定期的に収集してチェックする SDI 調査における過去の結果データを訓練データとして使うことで，製品との関連性で特許文書を機械学習で仕分けすることを提案する．また，機械学習の言語処理モデルとしては，2018 年に Google から発表された BERT が各種の言語処理タスクにおいて最も高い性能を達成しているので有力である．現状の日本語 BERT モデルは日本語 Wikipedia を使って事前学習した大型のモデルで，多くの計算機資源を必要とするため企業の一般的な PC では利用が難しい．そこで，日本語版の特許専用モデルを作成して，特許に関するタスクでは，より小型の BERT モデルでも現行の大型一般モデルと同等の性能を発揮することを確認したので，その結果を報告する．

キーワード：特許調査，自然言語処理，BERT，文書ランキング

BERT Pre-trained Model with Patent Documents and Application for Patent Survey

Kenji Akiyama[1,a)]　Takafumi Saito[1]

Received: February 21, 2022, Accepted: November 10, 2022

Abstract: Efficient selection of suitable documents from large number of documents is required in various fields. Some recent patent search databases provide a function of ranking and displaying the retrieved patent documents by the similarity with the query texts to support the selecting works. However, in the patent infringement avoidance survey, it is necessary to select the patent documents according to the relationship with a product under survey. The knowledge of products is mostly in the memory of the person in charge of development. Therefore, the selecting works relied exclusively on human. In this study, we propose to use the past result data of the SDI survey as training data for machine learning, and to sorts the patent documents according to the relationship with the product. (SDI is to collect and check the document regularly under the pre-specified search conditions.) BERT is a promising language processing model for machine learning, which is announced by Google in 2018 and has achieved the highest performance in various language processing tasks. The current Japanese BERT models are pre-trained using Japanese Wikipedia and have large size of parameters, which requires too much computer resources to execute on a general PC in a company. Therefore, we created a Japanese version of the patent-specific BERT model and confirmed that even a smaller parameter size of BERT model exhibits the same performance as a current Japanese BERT model in the patent ranking task. We report the effectiveness of the proposal and the BERT performances.

Keywords: patent survey, natural language processing, BERT, document ranking

1. はじめに

毎年数多く公開される特許文献や研究論文から，自らの業務や研究テーマに即した文献を効率よく仕分けする作業

[1] 東京農工大学
Tokyo University of Agriculture and Technology, Koganei, Tokyo 184–8588, Japan
[a)] ken2akiyama@gmail.com

は様々な場面で求められる．特に製品開発で他社特許を強く意識しなければいけない企業や，特許に関する調査を行う会社ではニーズが高い．人手による仕分け作業には多くの工数を要し，増大する文献に対応するのが難しくなっている．その解決策の 1 つとして，特許検索システムに自然言語処理技術を取り込み，検索された文献に調査対象技術の文書上の類似性によるランキングを行うシステムも実現されている [1], [2].

自然言語処理技術としては 2018 年に発表された BERT（Bidirectional Encoder Representations from Transformers）モデルは，様々な評価タスクで従来の性能を凌駕する性能を達成し [3]，Web 検索システムや翻訳システムなどに応用が広がっている [4]．さらに Google からは米国特許で訓練された特許専用モデルもリリースされた [5]．一方で BERT は計算コストが大きいという課題があり，専用のグラフィックボードや多くのメモリを搭載した計算機が必要となる．そのためモデルの軽量化 [6] の検討も行われている．

本稿では BERT の特許調査業務への応用について論じるので，最初に**図 1** の特許調査の作業フローを使って調査の特質を説明する．企業が行う特許調査には「技術動向調査」「出願前調査」「侵害回避調査」などがある．技術動向調査は調査対象の技術を元にクエリを作成し特許データベースから特許公報の集合を検索した後，人が調査対象の技術と判断できるものを数百件仕分けして，その統計的な分析を行う調査である．出願前調査は特許庁における先行技術調査とおなじで，新しく出願を考えている（出願された）文書から関連する特許文献を収集し，新規性／進歩性の判断基準となる数件を人手で仕分けする調査である．このときに特許検索された公報集合を更に調査対象の技術を表した文書との比較で機械学習がランキングすることで，人の仕分け作業をサポートすることも考えられている．

一方，侵害回避調査では調査する製品が他社特許を侵害していないことを確認する調査である．特許検索して公報集合を作成するところまでは同じ手順であるが，その後の仕分けの方法が異なり，公報集合と製品搭載技術を比較して仕分けする必要がある．ここで課題となるのは，製品に関する知識は製品仕様書や取扱説明書に記載されているかもしれないが，特許文献と類似性を比較できるような文書として存在しないことである．また製品知識の多くは開発者の頭の中にだけ存在する情報で，特許文献を解釈しながら判断する必要がある．知財専門雑誌の中で迫川も「侵害回避調査は文書の中身を解釈しないと判断できないという難しさを持っている」と指摘している [7].

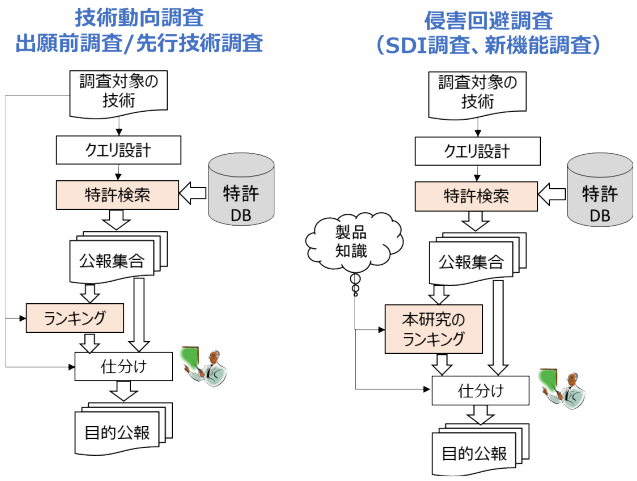

図 1　特許調査の作業フロー

Fig. 1　Workflows of patent surveys.

侵害回避調査には 2 とおりの実施方法がある．ひとつは新機能調査といって，新機能を導入するときに当該技術が他社の特許技術を侵害していないことを確認するためのものである．もうひとつはキーワードや技術分野などの条件をあらかじめ指定しておき，その条件に該当する特許情報を定期的に検索して自社と関係しそうな特許を仕分けし，必要なデータを収集・管理する調査である．この手法は一般的に Selective Dissemination of Information（SDI）と呼ばれて，特許文献以外の科学技術論文でも一般的に行われている．

侵害回避調査では人手による仕分け作業に工数がかかるという課題がある．製品開発には有限な費用と工数の範囲で行う必要があるので，漏れのない調査を目指しながら妥当な範囲に絞って調査実務を行っているのが現状である．この課題に対応すべく，本研究では SDI 調査を対象として，過去に仕分けした情報の蓄積を使って製品との関連性で特許公報集合をランキングする，という方法を提案する．この仕分け情報には製品との関係を人が判断した情報が含まれているためである．これにより侵害回避調査の効率化を図ることを目的とする．提案の有効性を示すために，SDI 調査の仕分け情報を使ったランキングが正しく行えるかを BERT モデルで検証する．また何年データを蓄積していると有効なランキングができるかを検証する．

更に，Wikipedia 等で事前学習した一般 BERT モデルは計算コストが高いという課題があるので，特許文献で事前学習した特許専用の小型のモデルを作成し，一般 BERT モデルとの代替可能性ついても確認を行う．

本稿では次の順序で説明する．2 章では今回の提案と関連した研究と特許検索サービスについて概観する．3 章では本研究で使用する BERT モデルの基本的な構成と特質を説明し，日本語特許専用モデルの作成方法について述べる．4 章ではランキングを評価するためのタスクの説明を行うとともに，5 章では評価指標としてより分かりやすい REI という指標を導入したので，その考え方とメリットを説明する．6 章では実験結果を示し，7 章でその結果の考察を行い，8 章で提案手法の有効性と将来の応用についてまとめる．

2.　関連研究

文書のランキングに関しては，古川らは入力文書と検索対象特許公報の文書ベクトルのコサイン類似度によってランキングする方法を提案している [8]．一方 Tian らは BERT のコンテキスト依存の類似性計算の利点を使った科学技術論文の検索について提案している [9]．

またこのような機械学習を使った類似度計算を実際の特許検索システムに適用した事例もある．たとえば，amplified ai Inc．が提供するサービスでは，調べたい技術を説明した文章や指定した登録番号の特許との類似度の高さで特許公報をソートする機能を提供している [1]．

パテント・インテグレーション株式会社では入力した文書から類似の特許公報を検索する概念検索を可能としており，クエリ設計も自動化する一方で，検索された特許公報に対してランキングを行うことで人手の仕分け作業が効率化できる機能を有している [2]．以上のランキング機能はいずれも出願前調査などでクエリ作成の前提となる文書との類似度を計算するもので，製品知識を使ってランキングするものではない．

一方で Panasonic が提供する PatentSQUARE では「AI 検索機能」という機能を提供しており，社内分類情報を付与した情報から検索された特許と社内分類との関係の有無を判断するサービスを提供している [10]．この機能に社内分類情報として過去の仕分け情報を入力すれば，本研究で行うランキングに近い機能が達成できる．AI 検索機能の性能については特に情報が開示されていないが，本研究はこれと同等の機能を BERT モデルで確認することに相当する．さらに本研究では Web サービスで使用している大規模な計算機を使用せず，一般的な PC でも実行できる特許専用モデルの検証も行う．

3.　BERT と分野専用事前学習モデル

3.1　BERT のしくみ

BERT の詳細な動作に関しては Alammar の解説 [13] に委ねるが，ランキングを行う上で必要な構成と性能に影響するパラメータについて**図 2** を使って説明する．

BERT は従来の自然言語処理モデルである再帰型ニューラルネットワーク（Recurrent Neural Network：RNN）[11] のようなトークンのシーケンス入力を再帰的に計算して文書全体の情報をベクトルで表現する方法と異なり，アテンション（注意機構）により計算する Transformers [12] を使った複数のエンコーダ層で構成されることが特徴である．BERT のアテンションの計算は入力されるトークン間のすべての関係を計算するためパラメータ数が RNN に比べて格段に大きくなり複雑なモデル表現が可能な一方で，計算機資源を多く必要とする．

BERT のもう 1 つの特徴は，個別タスクの学習前に大規模なコーパスを使って事前学習モデルを作成することである．事前学習モデルには図 2 の緑色部分が含まれ，大規模コーパスを使ってエンコーダのパラメータを学習する．この学習はコーパス文書のトークンの一部をマスクして，それを予測する訓練と，一連の文が次に現れる文か否かを判定する訓練を行うことで，トークン間の関係を学習する．この訓練データはコーパス文書があれば自動で生成できるので大規模な学習が可能となる．従来の事前学習モデルは大規模な計算資源を持った組織が作成・公開して，個別タスクに使うユーザが利用できるようになっている．

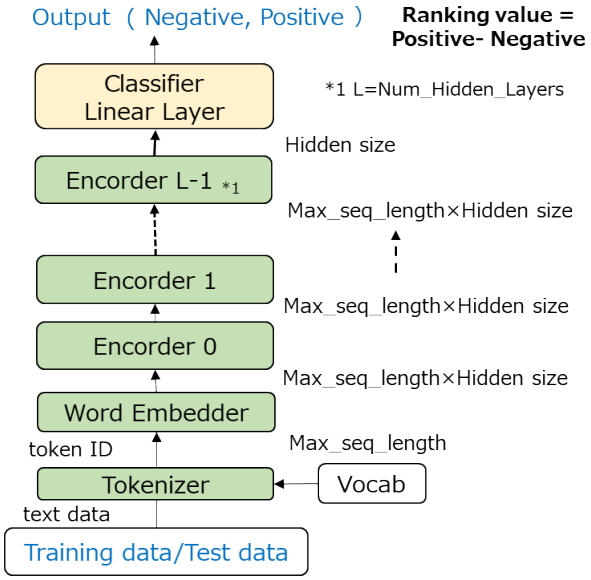

図 2　BERT モデルの構成
Fig. 2　Structure of BERT model.

事前学習モデルをベースに分類タスクのような個別学習を行うときには，黄色の分類器を付加して分類タスクの訓練データを入力して分類器のパラメータを調整するとともに，エンコーダのパラメータも微調整する．そのため個別学習は Finetuning と呼ばれている．あらかじめ事前学習することで少ない教師有りデータでも，個別の分類タスクにおいて高い性能を上げることが可能となる．

具体的な個別学習の手順を説明する．本研究で取り上げる「製品との関連性で特許公報集合をランキングする」というタスクでは，過去に選別を行って関係あり（正例）関係なし（負例）のラベルがついた文書を訓練データとして入力する．

形態素分解（Tokenizer）では「Vocab」という辞書ファイルにもとづきテキストがトークン単位に分解され，文書の先頭に [CLS]，区切りに [SEP] という特殊トークンを付加した後にトークン ID に変換される．このとき辞書にない単語は［UNK］というトークンに置き換えられる．また辞書に登録されていない単語でも，Byte Pair Encoding（BPE）を使うことで，複数のサブワードに分解する手法を使う場合もある [14]．たとえば“エアコンディショ

ナ”という単語を，“エア”と“コンディショナ”という2つのサブワードに分割することで，辞書登録数を大幅に増やさずに [UNK] トークンを減らすことができる．

単語埋め込み層（Word Embedder）ではトークン単位で辞書に登録されたトークン ID が Hidden size の埋め込みベクトルに変換されエンコーダ層（Encoder 0）に入力される．辞書には Vocab_size のトークンが登録されていて，1 つのトークンが Hidden_size の次元のベクトルに変換されるので，この次元が高いほど多様なトークンの表現が埋め込まれることになる．エンコーダに入力できるトークン数（文の長さに相当）は最大シーケンス長（Max_seq_length）であらかじめモデル設計時に決められているので，それを超える入力は切り捨てられる．

エンコーダ層は複数（Num_hidden_layers）のエンコーダによって構成される．このエンコーダ層の中でアテンション計算が行われ，入力されたトークン間の関連性が埋め込みベクトルによって計算される．さらに 1 つのエンコーダ層の中で並列して複数のアテンション計算が行われており，並列数を Num_attention_heads で表す．

エンコーダの最終層からは先頭の [CLS] に対応する Hidden_size の埋め込みベクトルだけが取りだされ，個別タスクの分類器（Classifier Linear Layer）に入力される．今回検証するタスクは 2 値分類タスクになるので，分類器は正例および負例の予測確率に対応した値（Positive, Negative）を出力する．個別学習時にはこれらの出力と教師データとの差を損失関数として計算し，分類層のパラメータにフィードバックするとともに，エンコーダ層のパラメータも微調整する．

予測計算時にはラベルのないデータが個別学習後のモデルに入力され，分類層の出力（Positive，Negative）を比較して入力文書に対する分類を決める．また Positive と Negative の値の差を評価値として，入力された文書のランキングを行うことができる．

Google から発表された BERT のオリジナル事前学習モデルは英語の Wikipedea と BookCorpas [15] を使ってモデルが作成された．その中で代表的な BERT-uncased というモデルのパラメータを**表 1** に示す．ここで uncased は入力される文字を大文字と小文字の区別をせず，すべて小文字に置き換えて処理するモデルである．モデルサイズによって Tiny から Large までのモデルサイズが定義されている．また辞書は各モデル共通で，BPE を使った 30522 個のトークン辞書でモデルが作成されている．

3.2　英語版特許専用モデル

特許文献を使った事前学習モデルは Google から 2020 年に公開された [5]．このモデルは米国の特許公報のみを使って事前学習されており，BERT-uncased で使われた辞書に対して，特許文献に特有な単語約 1 万件を追加している [16]．このモデルを Patent Model とよびパラメータを表 1 の下段に示す．本研究では日本語版の特許専用モデルの作成に先だって，特許専用モデルの性能をオリジナルモデルと比較する検証を行った．特許関連タスクとして特許分類コードを推定するタスクを定義する．

米国に特許出願された発明文書は，発明の要約を示す Abstract と詳細な発明内容を説明する Description と，特許の権利範囲を確定させる Claim から構成されている．通常 Claim は複数含まれているが第一請求項の権利範囲が広く，発明の特徴を最も良く表す内容が記載されている．また出願され公開される発明には Corporative Patent Classification（CPC）というコードが，欧州特許庁と米国特許庁で付与される．**表 2** に CPC コードの例を示す．

この分類コードは 4 桁のアルファベットおよび数字と「/」で区切られた 2 つの数字で構成され，25 万の技術を表現している．この分類コードは階層的な構造を持っており，表 2 の例では H01B がケーブル等に関係する技術で，1/02 はケーブルが金属または合金から構成されること，下の階層の 1/023 はアルミ合金であることを表している．前記 4 桁の文字の先頭文字は A〜H および Y の 9 種で，**表 3** に示すような技術範囲を表していて，例の H01B は電気の分野であることを示している．特許文献には複数の CPC が付与されているが，先頭の CPC はその特許の最も特徴的な技術範囲を表している．今回はモデル間の比較を

表 1　代表的な BERT モデルのパラメータと性能
Table 1　Parameters and performances of typical BERT models.

Model Name	Model size	Corpus	Vocab Size	Num hidden layers	Num attentin heads	Hidden size
BERT_uncased_tiny	Tiny	Wikipedia+ Bookcopus	30522	2	2	128
BERT_uncased_mini	Mini			4	4	256
BERT_uncased_small	Small			4	8	512
BERT_uncased_medium	Medium			8	8	512
BERT_uncased_base	Base			12	12	768
BERT_uncased_large	Large			24	24	1024
Patent Model	Large	US Patent	39859	24	24	1024

表 2　CPC コード例
Table 2　CPC code Examples.

H01B	CABLES; CONDUCTORS; INSULATORS; SELECTION OF MATERIALS FOR THEIR CONDUCTIVE, INSULATING OR DIELECTRIC PROPERTIES
H01B 1/02	. mainly consisting of metals or alloys
H01B 1/023	. . {Alloys based on aluminium}
H01B 1/026	. . {Alloys based on copper}

表 3　CPC コードセクションレベル技術内容
Table 3　Technical descriptions in section level CPC codes.

Code	Description
A	HUMAN NECESSITIES
B	PERFORMING OPERATIONS; TRANSPORTING
C	CHEMISTRY; METALLURGY
D	TEXTILES; PAPER
E	FIXED CONSTRUCTIONS
F	MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING
G	PHYSICS
H	ELECTRICITY
Y	GENERAL TAGGING OF NEW TECHNOLOGICAL DEVELOPMENTS etc.

することが目的なので，最も簡単な CPC 上位 1 桁を要約から推定するタスク（CPC1_abst）と，第一請求項から推定するタスク（CPC1_fclaim）を行った．

また，一般文書のタスクとしては良く知られている General Language Understanding Evaluation（GLUE）の中から 2 つのタスクを実行した [17]．MRPC（Microsoft Research Paraphrase Corpus）はオンラインニュースソースから集められた 2 つの文書が意味的に一致しているかを判定するタスクである．SST-2（The Stanford Sentiment Treebank）は映画レビューが肯定的か否定的かを判定するタスクである．事前の評価として，これら 2 つのタスクを使って評価する．

事前検証結果を**表 4** に示す．CPC1_abst および CPC1_fclaim ともにモデルサイズが大きくなるに従って精度が高くなることが確認された．Bert_uncased_large と同じモデルサイズの PATENT Model は CPC1_abst で最も高い 0.798 の精度を達成している．しかし，一般文書のタスクではモデルサイズが大きくなるに従って性能が向上していることは確認できるが，PATENT Model は SST-2 で 0.885 の精度，MRPC で 0.874 の F1 値であり，Base サイズにも満たない性能となることが確認された．以上のことから特許専用モデルは特許に関するタスクには良い性能を発揮するが，一般文書のタスクでは有効でないことが確認できた．特許専用の辞書を持っていることで未知語が減ることと，特許文書に固有の文書の特徴を事前学習モデルが学習していることが要因と推測している．

なお，BERT は多くの計算機資源を必要とし，モデルサイズが Large のモデルは付録 A.2 で示した実験環境では動作しない．そのため Web 上の計算機資源である Google Colaboratory [25] を使い，ランタイムの仕様をハイメモリというメモリ容量を大きく確保する設定を行って実施した．Base モデルでも一般 PC では一度に処理するデータ量（Batch size）を 8 に制限しないと動作しないので，すべての個別学習時の Batch size を 8 とした．

3.3　日本語版 BERT

BERT の日本語版モデルとしては情報通信研究機構（以後 NICT）から日本語 Wikipedia をコーパスとした事前学習したモデルが公開されている [18]．このモデルの諸元を**表 5** の上部に示す．形態素分解の際に BPE を使用した約 3 万語の辞書によるモデル（NICT_Base_32KBPE）と，BPE を使用せずにおよそ 10 万語の辞書を使ったモデル（NICT_Base_100K）が提供されている．

その他京都大学 [19] や東北大学 [20] からも，同様な日本語 Wikipedia をコーパスとしたモデルが公開されている．いずれもオリジナルモデルの Base および Large のモデルサイズを基準にして作成されたモデルである．これらを利用して日本語タスクに適用することが可能な環境が整ってきている．しかしながら，日本語の特許文献を使った事前学習モデルは現状発表されていない．英語版で特許専用モデルの優位性は事前検証できたので，日本語版でも特許文献をコーパスとしたモデルが特許文献を対象とした個別タスクに対してより適していることが予想される．

3.4　日本語版特許専用モデルの作成

本研究では日本語特許文献をコーパスにした BERT 事前学習モデルを作成し，NICT が作成した一般モデルと比較する．NICT と同等なモデルサイズではモデルのパラメータが 1 億 1 千に達するため通常の PC では資源不足から計算できない．NICT モデルの作成にはグラフィックボード（NVIDIA 社製 Tesra V100）を 32 枚搭載したコンピュータで 7〜9 日を要した，と NICT の Web ページの中 [18] で報告されている．本研究では，通常のグラフィックボードを搭載した一般的な PC でも計算できるよう，Tiny と Mini という小さなパラメータサイズのモデルを作成した．Max_seq_length は特許の要約がおおよそ収まるサイズとして 256 を選択し，Batch size も大幅に減らして 32 とした．使用した特許公報も 2012 年から 2017 年に公開された約 180 万件の公開公報を使ったもので，限られたデータで事前学習モデルを作成した．これらのモデルのパラメータ値を表 5 の下部に示す．

コーパスとしては要約，請求項，および両方を使ったモデルを作成した．また特許公報で使用頻度の高いトークンを登録した専用辞書を作成し，BPE を使うモデルと使わないモデルを作成した．ただし，登録単語数は NICT の

表 4　特許専用モデルの性能比較
Table 4　Performance comparison for patent specific model.

Model name	Task type	Patent task		General Task	
	Task name	CPC1_abst	CPC1_fclaim	SST-2	MRPC
	Metric	Acc.	Acc.	Acc.	F1
BERT_uncased_tiny		0.715	0.716	0.807	0.807
BERT_uncased_mini		0.733	0.733	0.866	0.820
BERT_uncased_small		0.742	0.736	0.875	0.841
BERT_uncased_medium		0.747	0.736	0.896	0.876
BERT_uncased_base		0.756	0.752	0.917	0.885
BERT_uncased_large		0.769	0.763	0.937	0.891
PATENT Model		0.798	0.794	0.885	0.874

表 5　実験に使用した日本語 BERT 事前学習モデル
Table 5　Japanese BERT pre-trained models for the experiments.

Model_Name	Corpus	Vocab_size	BPE	Cal_time	Model_size	Max_seq_lengrh ×Training steps	Batch _size
NICT-Base_100K	日本語 Wikipedia	100016	×	9 days	Base	128×1M 512×100k	4096
NICT-Base_32KBPE		32016	○	7 days			
Pabcl-Tiny_100K	特許要約＋請求項	100016	×	11h55min	Tiny	256× 1M	32
Pabcl-Tiny-32KBPE		32016	○	7h40min			
Pabcl-Mini_100K		100016	×	27h05min	Mini		
Pabcl-Mini_32KBPE		32016	○	21h10min			
Pabst-Tiny_100K	特許要約	100016	×	10h51min	Tiny		
Pabst-Tiny_32KBPE		32016	○	7h33min			
Pabst-Mini_100K		100016	×	27h10min	Mini		
Pabst-Mini_32KBPE		32016	○	21h01min			
Pclaim-Tiny_100K	特許請求項	100016	×	11h57min	Tiny		
Pclaim-Tiny_32KBPE		32016	○	7h35min			
Pclaim_Mini_100K		100016	×	27h05min	Mini		
Pclaim_Mini_32KBPE		32016	○	21h04min			

モデルと同じで，BPE 有りで 100016 トークン，BPE 無しで 32016 トークンとした．モデルの具体的な事前学習方法を付録 A.1 に示す．また事前学習には付録 A.2 で示した PC を使い Tiny で 7〜12 時間，Mini で 21〜27 時間を要した．

特許専用の事前学習モデルを作れば，同じモデルサイズならば性能が高くなることが 3.2 節の英語版モデルで確認されている．従って，企業が所有している一般的な PC を使って企業ごとに持っている過去データで個別学習を行いランキングが実行できるように，小さなモデルで検証を行った．

4. 特許ランキング手法

特許に関係するタスクとしては，国立情報学研究所（NII）が主催する NTCIR（NII Testbeds and Community for Information access Research）から特許分類タスク [21] が提供されているが，本研究では，特許調査の実務に即した独自の特許ランキングタスクを設定した．

本研究に協力をいただいた企業では，SDI 調査として毎月新たに登録される特許公報から検索式に基づいて特許を抽出し，それを専門の調査会社が振り分ける作業をしていた．この中で 2013 年から 2016 年までの 4 年間に行った調査データを使用させていただいた．**表 6** にデータの概要を示す．年ごとに比率の変動はあるが，全体で 23573 件の検索結果から 2675 件，比率では 11.3% の特許を人手で仕分けしている．この作業が機械学習で代替えできるか否かを確認する．

4.1 事前学習モデルとランキング性能

事前学習モデルごとの性能を比較するため，特許ランキングタスクのデータをすべて使うランキング性能検証タスクを**表 7** のように定義する．ランダムに並べた公報を 9 対 1 の割合で分割してそれぞれ訓練データと試験データとする．また個別学習および予測に特許公報の要約文を使うケースと，請求項を使うケースの 2 種のデータを作成する．ただしすべての請求項を使用するとトークン数が大きくなり過ぎるので，第一請求項のみを使用する．事前学習モデルとしては表 5 に示した 14 種の日本語事前学習モデルを使い，要約または請求項でランキングを行う．

実験は，訓練データを使って事前訓練済みの BERT モデルの個別学習を行い，訓練後のモデルを使って試験データに対して予測を行う．このとき分類器から得られる評価値の Positive と Negative の差でランキングを行う．個別学習は 1 回から 5 回のエポック数で繰り返し訓練を行い，エポックごとに試験データの予測を行う．実験結果には 5 回のエポック数の中で試験データの予測性能が最もよいものを結果として示す．評価指標として REI，AUC，NDCG の値を示すが，評価指標の詳細については 5 章で説明する．

表 6　特許公報ランキングタスクのデータ
Table 6　Patent publication ranking task data.

登録年	検索結果	選別数	破棄数
2013	6912	631	6281
2014	5829	733	5096
2015	5179	747	4432
2016	5653	564	5089
合計	**23573**	**2675**	**20898**
比率	100.0%	11.3%	88.7%

表 7　ランキング性能検証タスクのデータ
Table 7　Ranking performance evaluation task data.

Task_Name	Train data			Test Data		
	データ年度	使用文書	公報数	データ年度	使用文書	公報数
ET_ALL Abstruct	2013-2016	要約	21216	2013-2016	要約	2357
ET_ALL 1st Claim	2013-2016	第一請求項	21216	2013-2016	第一請求項	2357

4.2 訓練データ量の検証

4.1 節のランキング評価は，4 年間のデータを訓練データと試験データとに分割してランキングを行っているが，実務上は過去何年かの実績データをもとに新たに公開された特許公報をランキングすることが行われる．そこで前年までに得られた仕分け結果を訓練データとし，当年に公開された公報をランキングする年度評価タスクとして再定義する．この評価タスクのデータ構成を**表 8** に示す．年度評価タスクの事前学習モデルには NICT-Base の BPE 無し／有りの 2 種と，特許の要約と請求項を使って事前訓練を行った Pabcl-Mini の BPE 無し／有りの 2 種を使う．そして要約を使って個別学習と予測を行う評価実験を実施する．

5. ランキング性能の評価

文書集合に対して選択／破棄を予測する問題は 2 値分類問題として扱うことができる．しかし，今回定義したタスクデータで選択されたものを正例，破棄されたものを負例とすると，正例と負例の割合はおよそ 1 対 9 と不均衡なので，精度で評価しても分かりやすい指標が得られない．本研究では分類タスクで得られた評価値を元にランキングを行い，3 つの指標で結果を示す．第 1 の指標は ROC（Receiver Operation Characteristic）カーブの下の領域の大きさで評価する AUC（Area Under Curve）[22]，第 2 の指標はランキング評価で良く使われる Discounted Cumulative Gain を規格化した NDCG（Normalized Discounted Cu-

表 8　年度評価タスクのデータ
Table 8　Yearly evaluation task data.

Task_Name	Train data			Test Data		
	データ年度	使用文書	公報数	データ年度	使用文書	公報数
ET_2014 Abstruct	2013	要約	5328	2014	要約	5829
ET_2105 Abstruct	2013-2014	要約	11157	2015	要約	5179
ET_2016 Abstruct	2013-2015	要約	16336	2016	要約	5653

mulative Gain))[23] を使用する．そして第 3 の指標として仕分け作業効率の改善度を表すためのランキング評価指標（Ranking Evaluation Index：REI）を独自に導入する．これは ROC と同様な手法で計算される指標であるが，REI は無作為なランキングがされていれば 0，完璧なランキングができれば 1 となる指標であり，平均的な改善率を示す指標として理解しやすい．従って REI を使って説明し他 2 つの指標も結果に併記した．各ランキング指標の特質を以下に示す．

5.1　AUC

2 値分類の識別能力を評価する判別の特性は受信者動作特性（ROC）で表すことができる [22]．**図 3**(a) は 2 値判定の閾値を変えたときの偽陽性数と真陽性数をグラフ化したもので ROC 曲線と呼ぶ．グラフがより上方向を通過すると陽性と陰性のデータをきれいに切り分けられたことになる．従ってグラフの下の領域の面積（AUC）を理想的な場合を基準に規格化することでランキング評価指標として使うことができる．

5.2　NDCG

Web サイト検索のランキング評価指標として広く用いられるのは NDCG である．ランキングされた適合度を順位に応じた割引率で累積した値を DCG（Discounted Cumulative Gain）としてランキング評価に使う手法で K. Javelin らによって提案された [23]．DCG は次の式で表される．

$$DCG(\boldsymbol{G}) = G_1 + \sum_{i=2}^{k} \frac{G_i}{log_2 i}$$

ここで $\boldsymbol{G}$ はゲインベクトルで G_i は i 番目のランキングに対する適合度を表している．DCG は k 番目までの要素まで累積計算しており，適合度の高い要素が上位に出現するほど大きくなる指標である．理想的なランキング $\boldsymbol{G}_{ideal}$ に対して推定したランキングが $\boldsymbol{G}_{pred}$ となった場合，各ゲインベクトルの DCG の比率をランキング評価値 NDCG とする．

$$NDCG = \frac{DCG(\boldsymbol{G}_{pred})}{DCG(\boldsymbol{G}_{ideal})}$$

この指標は 1 から 0 の値を取り，適合度の高いものを上位にランキングするほど 1 に近い値となる．

5.3　REI

ランキングによる作業効率の改善という観点でランキング性能指標（REI）という評価尺度を導入する．図 3(b) は文献をランキングした順序でチェックした場合に，何件目のチェックで選択すべき文献を何パーセントカバーしたかを示すグラフで，REI 曲線と呼ぶ．文献総数を N，選択すべき文献数を X とする．全く無作為に文献がランキングされたならば，緑色のグラフのようにチェックの進捗に従ってカバー率が比例して上がっていく．一方，理想的なランキングができた場合には，目的の文献は最上位にランクされるので，赤色のグラフのように X 件をチェックした時点ですべてカバー率は 100% となる．実際ランキングの推定ではカバー率曲線は青い曲線のように 2 つの線の間を通過する．

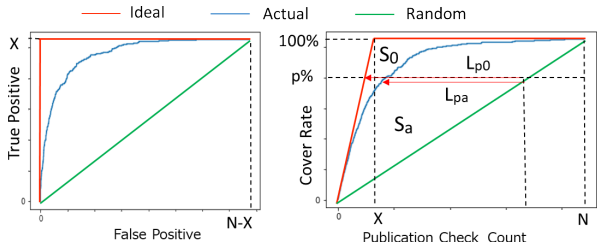

図 3　(a)ROC 曲線　(b)REI 曲線
Fig. 3　ROC/REI curve.

たとえば，選択すべき文献の p% をカバーしたいと考えた場合，対応する横軸に引いたラインで緑色グラフとの交点が無作為に文献を当たったときの作業量，赤色グラフとの交点が理想的なランキングの場合の作業量，青色グラフとの交点が実際の作業量に対応する．図中の L_{pa}/L_{p0} は理想的なランキングと比較して作業改善できた指標となる．この指標は選択すべき文献のカバー率によって変わるが，すべてのカバー率に渡ってこの比率を積算することで，平均的な作業改善を表現できる．そこで緑色グラフと赤色グラフで囲まれる面積 S_0，および緑色グラフと青色グラフで囲まれる面積 S_a から，次の式で REI を定義する．

$$REI = \frac{S_a}{S_0}$$

この指標値は −1 から 1 の値をとり，1 に近いほど理想的なランキングであり，0 に近いほど無作為なランキングになり，理想と真逆のランキングをすると −1 となる．

5.4　ランキング指標の比較

ROC 曲線はチェックした文献の中に含まれる選択すべき文献数を縦軸に，外れた文献数を横軸にプロットするので図 3(a) で示す曲線となる．この曲線から下の面積を計算する指標 AUC は理想的なランキングの場合 1，無作為なランキングの場合 0.5，真逆なランキングになるとに 0 となる．REI と AUC は考え方が異なるが次式で置き換えられる．

$$AUC = 0.5 + REI/2$$

ランキング評価には，無作為なランキングの指標値が 0 となる REI のほうが，理想的なランキングとの近さを直感的に理解しやすいと考える．

前述の NDCG と REI を具体的な事例で比較したものを**表 9** に示す．10 個の文献の中に 3 つの適合度の高い文献があるとすると，理想的なランキングは $\boldsymbol{G}_{ideal}$ となる．こ

表 9　ランキング指標の比較
Table 9　Comparison of ranking indicators.

		REI	AUC	NDCG
G_{ideal}	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0]			
G_{pred1}	[1, 1, 0, 0, 0, 0, 0, 0, 0, 1]	0.333	0.667	0.875
G_{pred2}	[0, 1, 1, 1, 0, 0, 0, 0, 0, 0]	0.714	0.857	0.810

れに対して適合度の高いものを最下位に推定した G_{pred1} と適合度の低いものを最上位に推定した G_{pred2} を仮定する．これらランキングの指標値を表の右側に示す．

NDCG はランキング順位の対数を分母とする重みづけがされるので，適合度の高い文献を下位に推定した G_{pred1} よりも，適合度の低い文献を上位に推定した G_{pred2} の NDCG が悪化する傾向にある．一方，REI は G_{pred2} よりも G_{pred1} のほうが悪くなる．これは適合度の高い文献を下位にしたことが同じ比率で指標に影響を与えるためである．従って，NDCG は上位の精度を重視した指標で，REI は再現率を重視した指標と言える．また，NDCG は全く無作為のランキングに対して 1 より小さいどのような値を取るかは条件によって異なるので，ランキング良否判断のベースラインが分かりにくい．

以上のことから，Web 検索のランキングなどでは上位に適合率の高いものを多く含むことを優先した NDCG が適しているが，特許侵害調査などで関連技術を含む文献を仕分けする場合は，下位のランキングも上位と同等に扱う REI が適していると考えられる．本研究では REI を中心にして説明を行う．

6.　実験結果

6.1　事前学習モデルとランキング性能の結果

ランキング性能検証タスクの実験結果を**表 10** に示す．NICT-Base モデルでは BPE 無しおよび BPE 有りの事前モデルで要約を使って個別学習した場合に，REI でそれぞれ 0.832 と 0.825 の値が得られた．一方で，特許で事前学習したモデルでは，パラメータ数の大きい Mini のほうが Tiny に比べて高い性能が得られることが確認できた．ランキングに特許公報の要約を使った場合の平均が 0.794 に対して，第一請求項の平均は 0.772 なので，要約は 0.022 程度高めに出る傾向となった．特許請求項は，使用する文言が上位概念化された特別な単語を使われる傾向にある．また過去に出現した文言を参照するときには「前記○」，「該○」のような表記をすることや，すべてを一文で表すなど一般文書と比べ特有の表現をしていることから，要約に比べて悪い結果になった可能性がある．

また特許公報のトークン数の影響も考えられる．**図 4** に公報データに含まれるトークン数のヒストグラムを示す．要約は文書全体を 300 文字以内で作成することを求められるので，ほとんど最大シーケンス長の 256 以内に収

表 10　ランキング性能検証タスクの結果
Table 10　Results of ranking performance evaluation tasks.

Model_Name	ET_ALL_Abstruct			ET_ALL_1stClaim		
	REI	AUC	NDCG	REI	AUC	NDCG
NICT-Base _100K	0.832	0.916	0.925	0.824	0.912	0.915
NICT-Base_32K_BPE	0.825	0.913	0.926	0.817	0.908	0.916
Pabcl-Tiny_100K	0.748	0.874	0.865	0.711	0.856	0.836
Pabcl-Tiny_32KBPE	0.768	0.884	0.878	0.722	0.861	0.879
Pabcl-Mini_100K	0.819	0.909	0.913	0.799	0.900	0.913
Pabcl-Mini_32KBPE	0.823	0.912	0.921	0.798	0.899	0.896
Pabst-Tiny_100K	0.760	0.880	0.885	0.727	0.863	0.869
Pabst-Tiny_32KBPE	0.772	0.886	0.859	0.728	0.864	0.865
Pabst-Mini_100K	0.790	0.895	0.898	0.791	0.895	0.915
Pabst-Mini_32KBPE	0.822	0.911	0.918	0.802	0.901	0.924
Pclaim-Tiny_100K	0.753	0.876	0.875	0.750	0.875	0.893
Pclaim-Tiny_32KBPE	0.768	0.884	0.893	0.744	0.872	0.888
Pclaim-Mini_100K	0.813	0.906	0.909	0.798	0.899	0.909
Pclaim-Mini_32KBPE	0.819	0.910	0.908	0.800	0.900	0.897
Avarge	0.794	0.897	0.898	0.772	0.886	0.894

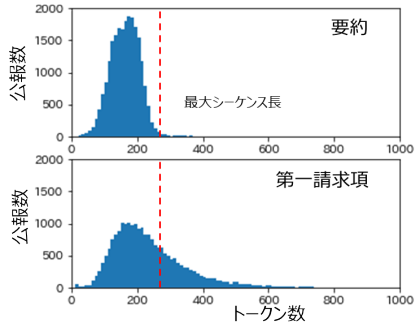

図 4　特許公報データのトークン数
Fig. 4　Token number in patent publication data.

まっており，これを越えるものは全体の 1.5% である．一方請求項にはこのような制限がなく，第一請求項に関しては 36% が最大シーケンス長を越えていて，これが性能に影響した可能性がある．しかし，要約に比べて極端に悪い結果ではなく，クレームを使った分類も可能であることを確認することができた．

事前学習モデルに注目すると，請求項と要約を使ったモデル（Pabcl-**）が最も高い性能を示している．要約だけでは文書の量が少ないが，請求項は平均でも約 13.9 個の請求項をもっているので，これらを個別の文書として事前学習に使うことにより，要約だけの場合より多くの文書で訓練できていることが理由と考えられる．特許を使ったモデルの中では，Pabcl-Mini_32KBPE が最も高い 0.823 の値を出しており，NICT-Base_32KBPE に迫る値となっている．表 1 で示したように，モデルサイズを上げれば高い性能が得られることは分かっているので，特許で作った base サイズのモデルが作成できれば，NICT で得られた成績よりも高い性能を達成することが期待できる．

6.2　訓練データ量の検証結果

年度評価タスクの実験結果を**表 11** に示す．各モデルとも過去 1 年のデータでは 0.7 程度の結果で，過去 3 年程度（約 16 千件）のデータがあれば 6.1 節で示した全データによる評価結果値近づくことが分かった．

表 11　年度評価タスクの結果

Table 11　Results of yearly evaluation tasks.

Model_Name	Task_Name	REI	AUC	NDCG
NICT-Base 100K	ET_2014 Abstruct	0.686	0.843	0.886
	ET_2105 Abstruct	0.749	0.875	0.909
	ET_2016 Abstruct	0.799	0.899	0.903
NICT-Base_32K_BPE	ET_2014 Abstruct	0.718	0.859	0.900
	ET_2105 Abstruct	0.799	0.900	0.936
	ET_2016 Abstruct	0.825	0.912	0.920
Pabcl-Mini_100K	ET_2014 Abstruct	0.717	0.859	0.903
	ET_2105 Abstruct	0.762	0.881	0.907
	ET_2016 Abstruct	0.786	0.893	0.895
Pabcl-Mini_32K_BPE	ET_2014 Abstruct	0.703	0.852	0.901
	ET_2105 Abstruct	0.777	0.889	0.935
	ET_2016 Abstruct	0.804	0.902	0.913

7.　考察

7.1　ランキング性能と改善効果

特許文献から作成した事前モデルにおいて REI で最大 0.823 の成績が得られたが，どのくらい正解文献が上位にランクされたか直感的に理解しにくい．**図 5** にランキング順序における正解の出現率をグラフとして表した．試験データ数 2357 件の中から 263 件の選択されたドキュメントを上位にランキングするタスクなので，理想的なランキングができれば赤線のようにランキング順位 263 位までは正解出現率は 1.0 でその後正解は 0 になる．無作為にランキングされた場合は緑のように一定して 0.11 程度の値を取る．実際のランキング結果を青線で，そのときの REI 曲線を橙色で示す．正解数の 2 倍の 526 位までに 82%，3 倍の 789 位までに 91% の正解が含まれており．関心のある文献をより先に確認できる効果が得られることが視覚的に理解できる．

ランキングによる作業効率改善効果について考察する．SDI 調査では毎回同じ条件で検索するので，仕分けで関係あると判定される公報数はおよそ予想できる．この実験の例では 11% を関係ありとしていたので，ランキング上位から 3 倍の 33% を仕分けすれば関係ありと判定される 91% はカバーできる．このカバー率を許容すれば残り 67% は仕分け作業から外すことが可能となる．データ提供に協力いただいた会社では毎月 500 件程度の仕分け作業を行っていた．検索結果の特許公報から大まかな製品との関係性を見る作業でも 1 件 1 分程度はかかるので，500 件の仕分け作業の 67% が削減できれば，500 分×67% ≒ 5 時間半程度の工数削減が可能となる．

侵害回避調査は漏れを無くした調査が求められるので，カバーできなかった 9% が気になるかもしれない．しかし，実務的には限られた予算と時間で行う必要があり，仕分け作業が時間的に難しければ，クエリ設計に条件を加えて検索の段階で件数を絞らざるを得ない．本実験では約 2 万件から 2 千件余りを仕分けしたデータを使っているが，仕分けしたデータがすべて特許侵害しているわけではない．実際に侵害回避のために設計変更をしたり，見つかった特許の無効化調査を行ったりするケースは，2 千件の中で数件である．従って仕分け段階であらかじめ製品との関係でランキングされていれば，上位の案件に調査時間を割き，時間の許す限り選別作業を行うという選択や，関係性の低い特許公報ばかりが出現すると感じた時点で仕分け作業を終了することも現実的な選択となり，図 5 において 2 千件余りの検索結果をすべて人手で仕分けする作業からは解放される．

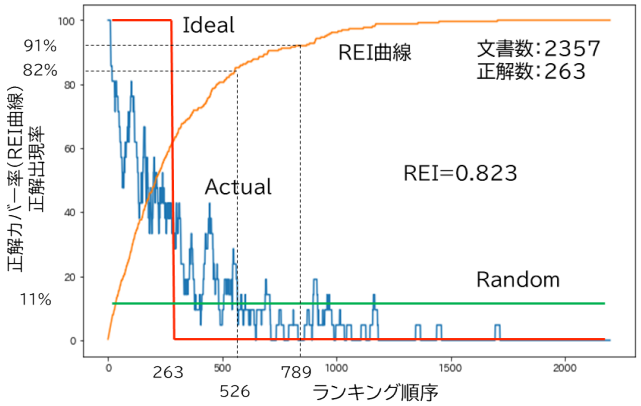

図 5　ランキング順序における正解出現率

Fig. 5　Correct answer rate in ranking order.

表 12　未知語の出現確率

Table 12　Unknown word appearance probability.

Corpus	Vocab_size	BPE	Model_Name	UNK_rate	
				Abstract	Claim
日本語 Wikipedia	100016	×	NICT-*_100K	1.78%	1.78%
	32016	○	NICT-*_32K_BPE	0.25%	0.33%
要約＋請求項	100016	×	Pabcl-*_100K	0.35%	0.32%
	32016	○	Pabcl-*_32KBPE	0.03%	0.01%
要約	100016	×	Pabst-*_100K	0.32%	0.39%
	32016	○	Pabst-*_32KBPE	0.02%	0.01%
請求項	100016	×	Pclaim-*_100K	0.39%	0.31%
	32016	○	Pclaim-*_32KBPE	0.04%	0.01%

7.2　未知語の削減による効果

小さなモデルでも，大規模モデルに近い性能を確認できた．**表 12** に各モデルでタスクのデータを扱ったときの未知語の出現確率を示す．NICT-Base のような一般文書から作成した辞書を使ったモデルでは未知語が 0.25〜1.78% 発生してしまう．一方特許文献を元に単語辞書を作成することで未知語減らすことができている．BPE を使わない場合でも 0.39% 以下，BPE を使うことで 0.04% 以下に落とすことができている．この未知語の削減が小さなモデルでも大規模モデルに迫る性能を上げた理由のひとつと考えられる．

7.3　クエリ検索しないデータに対する応用

6 章の検証では，検索式を使って抽出された公報集合とその仕分け結果をもとにモデルの個別学習を行った．この個別学習されたモデルが，クエリによる検索を行わない特許公報集合の仕分けに役立つかを検証した．

2021 年 11 月の 2 週間で新たに特許登録され公開された

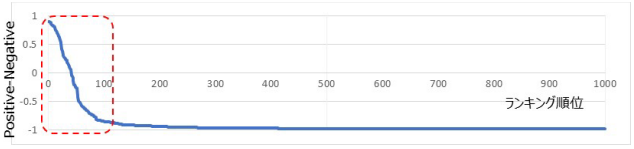

図 6　直近公報のランキングと分類器出力

Fig. 6　Recent publication ranking and classifier output.

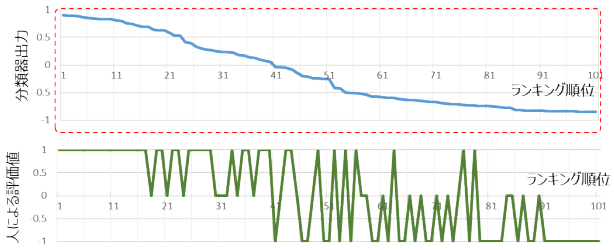

図 7　上位 100 位に対する人による評価値

Fig. 7　Evaluation value by person for the top 100.

5990 件に対して，3.4 節の特許文献から作成した Pab-cl-mini_32KBPE を要約で個別学習したモデルを使ってランキングを実行した．そのときの分類器の出力の差（Positive–Negative）をランキング順位で 1 位から 1000 位までプロットしたものを**図 6** に示す．事前のキーワード検索を掛けていないのでほとんどが無関係の特許公報と想定されるが，Positive–Negative の値も一部の公報についてのみ正の値を出力している．この上位 100 位までの公報について，自社製品との関連性で 3 段階評価（1：関連有り，0：中立，−1：関連なし）を人手で行った結果を**図 7** に示す．上位 100 件の中でも上位側に関連性の高いものが集まっている様子が確認できる．このことから，事前に自社製品との関係性を学習したモデルを作成しておくと，検索式なしで関係性の高い文献を取り出すことができることも確認できた．最初に検索式で選別していない公報集合に対しても製品との関連性でランキングできるので，今まで検索式で切り捨てていた中から新たに注目すべき公報を見つけられる可能性も考えられる．

8.　結論と今後の課題

本研究により，SDI 調査結果データを機械学習の訓練データとして使い，侵害回避調査のために検索した公報集合を調査対象の製品との関連性でランキングを行いうことで，人による仕分けのサポートができることを確認した．本稿の事例では 67% の作業改善が可能となることが分かった．これは特許関連の業務に関わらず，科学技術論文の検索など別の分野でも応用できると考える．検索クエリで表現しきれない人の志向や関心事に関する仕分け実績を使って，より自分の望む情報に早くたどり着くことが期待できる．SDI 調査の結果データを適切に残して，今後の業務改善に役立てたい．

また特許文献を扱うタスクでは，小さなサイズのモデルでも特許文献による事前学習モデルで作成することで，大規模な汎用モデルに迫る性能を示すことが確認できた．これにより，一般の企業の中で仕分け作業の機械化を行うことが可能となる．またパラメータサイズが大きいほど高い性能が得られることは知られている．今後大規模な特許事前学習モデルが作成されれば，ランキング性能の向上が見込まれる．また特許庁では，2016 年から「人工知能技術を活用した特許行政事務の高度化・効率化実証的研究事業」として，特許出願から審査，登録に関わる業務を対象に，人工知能技術の活用について調査を実施しており [24]，特許専用モデルはこのような業務改善にも資することが期待できる．

謝辞　本研究のために特許調査に関するデータを提供いただいた FCNT 株式会社に感謝する．

参考文献

[1] Amplified：自然で直感的な特許調査，〈https://www.amplified.ai/ja/how-it-works〉.
[2] パテント・インテグレーション：機能説明，〈https://patent-i.com/ja/wiki/〉.
[3] Devlin J., Chang M., Lee K. and Toutanova K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of NAACL-HLT 2019*, pp.4171–4186, ACL (2019).
[4] Google: google-research/bert，〈https://github.com/google-research/bert〉.
[5] Google: patents-public-data，〈https://github.com/google/patents-public-data/blob/master/models/BERT〉for Patents.md.
[6] Sanh V., Debut L., Chaumond J. and Wolf T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).
[7] 追川康之：深層学習を利用した自然言語処理の発展と特許調査への応用の現状，Patent, Vol,75, No.2, pp.3–16 (2022).
[8] 古川修平，関　洋平，青野雅樹：特許の無効資料調査のための類似特許検索とリランキング，FIT2008 第 7 回情報科学技術フォーラム，D-027 (2008).
[9] Tian X. and Wang J.: Retrieval of Scientific Documents Based on HFS and BERT, *IEEE Access* 9, pp.8708–8717 (2021).
[10] Panasonic：特許調査支援サービス PatentSQUARE，〈https://www.panasonic.com/jp/business/its/patentsquare.html#ai-search-function〉.
[11] Ilya S., Vinyals O. and Le Q. V.: Sequence to Sequence Learning with Neural Networks, NIPS (2014).
[12] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł. and Polosukhin I.: Attention Is All You Need, 31st Conference on Neural Information Processing Systems NIPS (2017).
[13] Alammar J.: The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)〈https://jalammar.github.io/illustrated-bert/〉.
[14] Sennrich R., Haddow B. and Birch A.: Neural Machine Translation of Rare Words with Subword Units, *Proc. of the 54th Annual Meeting of the ACL* (2016).
[15] Zhu Y., Kiros R., Zemel R., Salakhutdinov R., Urtasun R., Torralba A. and Fidler S.: Aligning Books and Movies: Towards Story-like Visual Explanations by Watching

Movies and Reading Books, ICCV (2015).
[16] Google: How AI, and specifically BERT, helps the patent industry, ⟨https://cloud.google.com/blog/products/ai-machine-learning/how-ai-improves-patent-analysis⟩.
[17] Wang A., Singh A., Michael J., Hill F., Levy O. and Bowman S. R.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, Conference paper at ICLR (2019).
[18] 情報通信研究機構：NICT BERT 日本語 Pre-trained モデル，⟨https://alaginrc.nict.go.jp/nict-bert/index.html⟩.
[19] 京都大学：BERT 日本語 Pretrained モデル，⟨https://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese⟩.
[20] 東北大学：Pretrained Japanese BERT models released / Tohoku NLP Lab, ⟨https://www.nlp.ecei.tohoku.ac.jp/news-release/3284/⟩.
[21] Fujii A., Iwayama M. and Kando N.: Overview of the Patent Retrieval Task at the NTCIR-6 Workshop, Proceedings of NTCIR-6 Workshop (2007).
[22] Streiner D. L. and Cairney J.: What's under the ROC? An introduction to receiver operating characteristics curves, *The Canadian Journal of Psychiatry*, Vol.52, No.2 (2007).
[23] Jarvelin K. and Kekalainen J.: Cumulated gain-based evaluation of IR techniques, *ACM Transactions on Information Systems*, Vol.20, No.4, (2002).
[24] 富永泰規：特許庁業務における人工知能技術の活用，Patent, Vol.75, No.2, pp.43–49 (2022).
[25] Google: Welcome To Colaboratory – Google Research, ⟨https://colab.research.google.com/⟩.
[26] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, ⟨https://taku910.github.io/mecab/⟩.
[27] Rico Sennrich: rsennrich/subword, ⟨https://github.com/rsennrich/subword-nmt⟩.

付録

付録 A.1　特許専用日本語 BERT モデルの事前学習方法

特許専用日本語 BERT モデルの事前学習時の設計条件を示す．github の google-research/bert [4] で公開されているプログラムの tokenization.py に Mecab [26] による日本語形態素解析ができるよう修正を加えた．コーパスは 2012 年から 2017 年に公開された約 180 万件の公開公報の要約と請求項を使い，前処理として「【要約】【課題】制御システムにおいて･･･」のような【　】書きの文字は削除し，半角文字は数字も含めて全角文字に変換した後に形態素分解を行った．辞書作成は github の rsennrich/subword-nmt [27] で公開されている subword_nmt.py を使った．形態素解析されたトークンで出現頻度の上位 10 万件（BPE 無し）と 3 万 2 千件（BPE あり）を選別し NICT モデルの辞書と同じ [UNUSED*] を含む特殊トークン 16 個を加えて登録した．事前学習は pretrain.py を次のパラメータで実行した．

--train_batch_size=32
--max_seq_length=256
--max_predictions_per_seq=20
--num_train_steps=1000000
--num_warmup_steps=10
--learning_rate=2e-5

付録 A.2　PC の概略仕様

実験に使用した PC の概略仕様を下記に示す．
OS：Windows 10 Pro（64bit）
CPU：インテル ® Core™ i9-9900K プロセッサー
GPU：NVIDIA® GeForce RTX™ 2080Ti　11GByte
Memory：32Gbyte
Storage：512GbyteSSD + 2TbyteHDD

秋山 賢二（正会員）

1984 年 東北大学大学院工学研究科博士前期課程修了．同年東芝入社．2011 年富士通転籍．2018 年 FCNT 株式会社（旧富士通コネクテッドテクノロジーズ）転籍．現在知財部門に所属．東京農工大学 生物システム応用科学府博士後期課程在籍．

斎藤 隆文（正会員）

1982 年東京大学工学部計数工学科卒業．1990 年東京大学大学院工学系研究科博士課程修了．工学博士．1987 年 NTT 研究所勤務．1997 年東京農工大学工学部助教授．現在，東京農工大学大学院工学研究院教授．コンピュータグラフィックス，可視化，形状処理などの研究に従事．本学会フェロー，画像電子学会フェロー．