Contrastive Learning を利用した類似特許検索

星野雄毅 [1]　内海祥雅 [2]　中田和秀 [1]
[1] 東京工業大学工学院　[2] 楽天グループ株式会社
hoshino.y.ad@m.titech.ac.jp　nakata.k.ac@m.titech.ac.jp

概要

近年，知的財産の管理は社会にとって大きな役割を担っている．特に，特許は毎年 30 万件を超える出願があり，膨大な量の特許を処理する上で多くの課題が存在する．そこで，本研究では特許を扱う上で非常に重要な類似特許検索タスクについて，Contrastive Learning の応用を考えた．この際重要な教師データ並びに Hard Negative の取得方法について提案を行った．さらに，実際の特許データを用いた数値実験を行い，その効果を検証した．

1　はじめに

1.1　類似特許検索の必要性

特許とは知的財産の一つで，発明を保護するものである．特許は各企業の技術を守るうえで重要な役割を果たしている．また，特許の審査を厳密かつ素早く行うことは特許制度を機能させる上で非常に重要である．したがって，特許処理技術は，企業側と特許庁側の両面で有用である．

特許に関わる様々なタスクの中でも，類似特許検索は企業と特許庁の双方にとって非常に需要の高いタスクである．まず，特許庁にとっては審査を行う上で類似特許検索を必ず行わなければならない．現在，類似特許検索は人手で行われており，特定の単語が含まれているかどうかや，国際特許分類などを用いて絞り込みながら行っている．しかし，適切な類似特許を見つけるのは難しく，審査に膨大な時間がかかる要因の一つとなっている．

また，類似特許検索は各技術を有する企業も頻繁に実施することになる．なぜなら，各企業は自社特許が侵害されていないか監視したり，自社の新技術が他企業の特許を侵害しないか確認する必要があるからである．さらに，新たに開発を行ったり，特許を申請する際にも類似特許を調査することで，他社特許との比較した優位性を明らかにする必要がある．

1.2　IPC とは

特許には，国際特許分類 (以下 IPC) という，国際的に統一された分類が付与される．IPC は技術分野を表す番号といえる．IPC には 2 つの特徴があり，階層構造を持っていることと，一つの特許に複数与えられることが挙げられる．

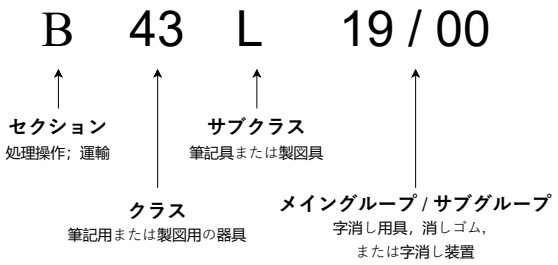

図 1　IPC の例

IPC は「セクション」,「クラス」,「サブクラス」,「メイングループ」,「サブグループ」の 5 つの要素から構成されており, 階層構造を持っている. 例えば図 1 の場合, セクションに当たる「B」は「処理操作」という大まかな分類を表している. 次にクラスの「43」はセクションの「B」と合わせてより細かい「筆記用または製図用の器具」を表し, 最後の「サブグループ」まで合わせると,「字消し用具，消しゴム，または字消し装置」という細かい分類にまで分けられる. このように，階層構造を持っていることで，様々な粒度で分野の絞り込みを行える．IPC は全部で約 7 万通り存在しており，多種多様な分野を表現可能である．

次に，IPC は一つの特許に複数与えられる．例えば，消しゴム付き鉛筆を発明したとすると，鉛筆に関係する IPC と消しゴムに関する IPC の両方が付与されることとなる．

2 関連研究

2.1 Contrastive Learning

Contrastive Learning[1] とは，画像処理から発展してきた表現ベクトルを学習する手法である．Contrastive Learning の最も特徴的な点としては，データ間の関連性から学習を進めることにある．具体的には，データ間について似ているものは近く，異なるものは遠くに学習を進めていく．Contrastive Learning を実行するうえで，通常負例としては，ミニバッチ内の他サンプルを用いることが多い．しかし，場合によっては自明な負例ばかりになってしまい，学習が不十分になることがある．そこで，予測が難しいと想定される負例を Hard Negative として明示的に入力して学習する場合もある．

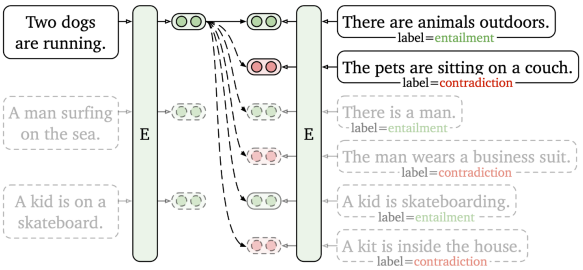

図 2 SimCSE の学習方法．原著 [2] より引用

SimCSE[2] は Contrastive Learning を自然言語に応用する方法である．図 2 のように，意味的な従属関係のある文を近くに，そうでないものを遠くに学習するものである．例えば，「２匹の犬が走っている」(“Two dogs are running.”) という文に対して，意味的な従属関係がある文とは「外に動物がいる」(“There are animals outdoors.”) といった文が対応する．ここで，教師あり SimCSE の特徴として，Hard Negative には意味的に排反なデータ [3] を使用していることが挙げられる．排反な意味のデータとは，「ペットがソファに座っている」(“The pets are siting on a couch.”) というように，同時に起こりえない２つの事象のことである．

2.2 特許への機械学習の応用

特許に対して機械学習技術を応用する研究はいくつか行われている．まず，古典的な機械学習手法を用いて IPC や自作ラベルの予測などが行われている [4][5]．また，深層学習を用いた研究では，BERT を特許データでファインチューニングし，IPC のサブクラスの予測が行われている [6]．さらに，IPC 予測について，入力方法とデコーダに工夫を加えることで精度の改善も行われている [7]．

機械学習手法を用いて類似特許検索するという研究はいくつか行われている．まず，Word2Vec を用いて，特許の類似度が比較された [8]．また，深層学習ベースだと，LSTM ベースのエンコーダを用いた類似特許検索モデルも作成された [9]．さらに，LSTM ベースのエンコーダにグラフ情報などを加え，精度の向上も行われた [10]．しかし，これらはいずれも自作でラベリングした類似特許データに対して学習，評価を行っており，これらの手法は公開されている情報だけでは応用できない．したがって，日本語の特許で公開された情報のみを用いて類似特許検索を行う研究はいまだされていない．

3 提案手法

3.1 引用情報の利用

類似特許検索を行う上では何をもって，類似特許とみなすのかという問題が存在する．この，「類似した特許」として適したデータとしては，無効審判請求及び，異議申し立てのデータが挙げられる．これらは，実際に厳密に時間をかけて審査されたものであり，その際に参照された特許は確実に類似特許とみなすことができる．一方で，これらのデータは学習データとして用いるにはデータ数が十分でないという課題がある．

そこで，本研究では特許の引用情報に注目し，引用されている特許を類似特許とみなすこととした．これは，類似した特許のすべてを網羅しているとは言えないものの，引用された特許は類似した特許であるとみなすことができる．さらに，引用特許はほとんどの特許に対して存在しており，日本語の特許情報からも取得可能である．したがって，引用情報を類似特許の正解データと仮定し，学習及び評価をすることを考える．

ここで，引用は各特許に対して複数存在していることに注意が必要である．そこで，各特許に対して１つの正例をランダムで取得するようにした．アルゴリズムは Algorithm 1 のようになっており，4 行目で引用特許の集合からランダムに選択をしている．ただし，B はミニバッチで学習したい特許の ID 集合，I_b は特許の IDb に対応する特許，$P(i)$ は特許 i に引用されている特許集合である．

Algorithm 1 特許データミニバッチ作成

Require: I, P, B

Ensure: $\text{batch}_i, \text{batch}_p$

1: $\text{batch}_i, \text{batch}_p \leftarrow [], []$
2: **for** $b \in B$ **do**
3: $\quad i \leftarrow I_b$
4: $\quad$ Choose p from $P(i)$ uniformly randomly
5: $\quad \text{batch}_i \leftarrow \text{concatenate}(\text{batch}_i, [i])$
6: $\quad \text{batch}_p \leftarrow \text{concatenate}(\text{batch}_p, [p])$
7: **end for**

Algorithm 2 IPC による HardNegative を加えたサンプリング

Require: I, P, B, N, H

Ensure: $\text{batch}_i, \text{batch}_p, \text{batch}_n$

1: $\text{batch}_i, \text{batch}_p, \text{batch}_n \leftarrow [], [], []$
2: **for** $b \in B$ **do**
3: $\quad i \leftarrow I_b$
4: $\quad$ Choose p from $P(i)$ uniformly randomly
5: $\quad$ Choose h from H uniformly randomly
6: $\quad$ Choose ipc from $IPC_h(i)$ uniformly randomly
7: $\quad$ Choose n from $N(ipc) \setminus (\{i\} \cup P(i))$ uniformly randomly
8: $\quad \text{batch}_i \leftarrow \text{concatenate}(\text{batch}_i, [i])$
9: $\quad \text{batch}_p \leftarrow \text{concatenate}(\text{batch}_p, [p])$
10: $\quad \text{batch}_n \leftarrow \text{concatenate}(\text{batch}_n, [n])$
11: **end for**

3.2 IPC を用いた Hard Negative

今回は Hard Negative の作成方法として，分野が同じものを取得することを考え，IPC を用いた Hard Negative のサンプリング方法を提案する．つまり，同じ IPC を持つものは分野が近いため，Hard Negative として学習を進める．ただし，IPC は一つの特許に複数付与されいるため，各分野からバランスよく Hard Negative を作成することにする．そのため，分野をランダムに選択し，その中の特許から Hard Negative として使う特許を選択することを考えた．

また，IPC は階層構造を持っているため，様々な粒度の分野情報を包含している．そのため，複数の階層に従ってサンプリングを実行することで，より良い Hard Negative を作成することができるのではないかと考えた．

以上の考えのもと，提案するアルゴリズムの疑似コードは Algorithm 2 のようになっている．ただし，$N(ipc)$ は ipc が付与されている特許集合，H はクラスやメイングループなどの対象としたい階層の集合で，$IPC_h(i)$ は階層 h における特許 i に付与されている IPC の集合である．負例は 5 から 7 行目で取得しており，それぞれ IPC の階層の選択，階層内の IPC の選択，特許の選択の 3 段階で実行している．このように確率的に様々な分野からのサンプリングを実行することで，様々な粒度の分野において近いかどうか判別できるようなモデルを作成可能である．

4 数値実験

4.1 実験設定

4.1.1 データセット

今回データセットは有料の特許データ取得サービスを通じて入手した．学習，検証データは，2016/07/01〜2016/12/31 に国内で出願された特許（104078 件）をすべて取得し，これらを検索元のデータとした．学習検証の分割は検索元のデータをランダムに 8:2 で分割した．テストデータは 2017/01/01〜2017/01/15 までに国内で出願された特許（4348 件）を用い，それらが引用した特許（13123 件）を検索対象の特許とした．

4.1.2 事前学習

本モデルの入力として，特許の技術内容を記した部分である「請求項」のテキストを用いる．一方で，請求項は長文で，公開されている事前学習済みモデルで良く用いられている MeCab[11] などの分かち書きをベースとしたトークナイザーでは多くの特許で入力トークンが膨大になってしまう．そこで，トークン数を減らすために特許で学習した Unigram Language Modeling[12] を用いたトークナイザーを作成した．これによって，入力できる最大のトークン数である 1024 トークン以内に全文が収まらない特許の割合は，32%程度から 15%程度まで減少した．

表 1 学習モデルによる比較

		Precision			Recall			NDCG		
モデル	Hard Negative	@1	@5	@10	@1	@5	@10	@5	@10	@inf
tf-idf	-	0.234	0.131	0.088	0.103	0.265	0.346	0.241	0.272	0.403
BERT	-	0.257	0.134	0.090	0.114	0.274	0.357	0.253	0.284	0.418
教師有 SimCSE	無	0.352	0.213	0.142	0.164	0.438	0.564	0.390	0.439	0.550
教師有 SimCSE	引用の引用	0.350	0.207	0.137	0.161	0.427	0.549	0.381	0.428	0.541
教師有 SimCSE	クラスのみ	0.358	0.218	0.144	0.163	0.448	0.573	0.397	0.446	0.553
教師有 SimCSE	サブクラスのみ	0.354	0.213	0.142	0.162	0.436	0.568	0.388	0.439	0.547
教師有 SimCSE	クラス+サブクラス	**0.358**	**0.218**	**0.146**	**0.167**	**0.451**	**0.583**	**0.400**	**0.452**	**0.556**

一方で，トークナイザーを自作したため，既存の事前学習済みモデルを適応することはできないため，BERT[13] の事前学習を実施した．モデルサイズは，学習環境と事前学習で長文に対して学習したい点も考え，512 次元 8head の 4 層からなるモデルとした．

4.1.3 評価指標

本提案手法の目的は，類似度が高い特許を業務を行う人に推薦することである．そこで評価方法としては，各特許の埋め込みベクトルのコサイン類似度が高かった順に検索対象の特許を並び替えて，引用した特許が上位に存在するかを評価した．評価指標は，一般に推薦システムで用いられる Precision@k， Recall@k, NDCG@k の 3 つの評価指標を用いた [14].

4.1.4 比較手法

比較手法として，モデルそのものの精度を検証するベースラインをいくつか用意した．まず，単純な機械学習モデルとして tf-idf を用いた単語モデルのコサイン類似度を用いたものを用いた．また，SimCSE の学習の必要性を検証するために BERT の埋め込みをそのまま用いたものと比較した．次に，ハードネガティブのサンプリング方法についていくつかの設定で行った．まず，ハードネガティブを全く用いず，バッチ内の他サンプルのみを負例として用いたものを作成した．次に，引用された特許の中で引用されていた特許のうち，引用されていなかったものをハードネガティブとして使用してモデルを学習した．さらに，単一階層での IPC を用いた Hard Negative との比較を行うため，クラスとサブクラスそれぞれ単独の階層で同一 IPC を保有する特許を Hard Negative としたものを学習した．最後に提案手法として，クラスとサブクラスの両方の階層を用いたものを作成した．

4.2 実験結果

結果は，表 1 のようになった．ただし，NDCG@inf は全データに対して NDCG を算出したものである．

まず，tf-idf の類似度と比較して，事前学習済み BERT の埋め込みを用いた類似度はいずれの評価指標でもわずかに高くなっていることがわかる．更に，教師あり学習を実行することで，いずれの評価指標でも大きく改善していることがわかる．また，Hard Negative について，引用の引用を用いたものは評価がむしろ落ちていることがわかる．これは，引用の引用では意味が近すぎるために Hard Negative としても難しすぎることが原因だと考えられる．一方，IPC を用いた Hard Negative は単一の階層を用いたものよりも，階層を増やすことで更に精度が改善しており，本提案手法が最も良い結果となった．

5 今後の課題

今後の課題としては，大きく二つ存在する．まず，請求項全文の入力である．本研究では，自作トークナイザーの作成によって入力トークンを減らすことを行ったが，15%程度の特許は全文入力することができていない．近年ではメモリ効率の良いエンコーダについても研究が進んでいるため，これらの応用によって対応できる可能性がある．次に，専門用語の入力である．実験によってうまく予測できなかった文書の内容を確認すると，分野特有の専門用語が含まれる特許を選択できていない場合があった．したがって，専門用語をトークンに含めることでその特徴をとらえやすくなり，より精度の良い学習が行える可能性がある．

謝辞

本研究を実施するにあたり, データ及び研究環境を提供してくださいました楽天株式会社の皆様, とりわけ知的財産部の皆様に深く感謝を申し上げます.

参考文献

[1] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 18661–18673. Curran Associates, Inc., 2020.

[2] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 6894–6910, 2021.

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[4] C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. Automated categorization in the international patent classification. **SIGIR Forum**, Vol. 37, No. 1, p. 10–25, April 2003.

[5] Mattyws F. Grawe, Claudia A. Martins, and Andreia G. Bonfante. Automated patent classification using word embedding. In **2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)**, pp. 408–411, 2017.

[6] Jieh-Sheng Lee and Jieh Hsiang. Patent classification by fine-tuning bert language model. **World Patent Information**, Vol. 61, p. 101965, 2020.

[7] Yuki Hoshino, Yoshimasa Utsumi, Yoshiro Matsuda, Yoshitoshi Tanaka, and Kazuhide Nakata. Ipc prediction of patent documents using neural network with attention for hierarchical structure. **Research Square preprint DOI:10.21203/rs.3.rs-1164669/v1**, 2022.

[8] Hidir Aras, Rima Türker, Dieter Geiss, Max Milbradt, and Harald Sack. Get your hands dirty: Evaluating word2vec models for patent data. In **SEMANTiCS Posters&Demos**, 2018.

[9] Aaron Abood and Dave Feltenberger. Automated patent landscaping. **Artificial Intelligence and Law**, Vol. 26, No. 2, pp. 103–125, 2018.

[10] Seokkyu Choi, Hyeonju Lee, Eunjeong Park, and Sungchul Choi. Deep learning for patent landscaping using transformer and graph embedding. **Technological Forecasting and Social Change**, Vol. 175, p. 121413, 2022.

[11] Toshinori Sato, Taiichi Hashimoto, and Manabu Okumura. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In **Proceedings of the Twenty-three Annual Meeting of the Association for Natural Language Processing**, pp. NLP2017–B6–1. The Association for Natural Language Processing, 2017.

[12] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 66–75, 2018.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. **Egyptian informatics journal**, Vol. 16, No. 3, pp. 261–273, 2015.