

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



株式会社 amplified ai[※] 追川 康之

要 約

深層ニューラルネットワークを用いた機械学習による自然言語処理の発展は著しく、基本的な言語処理能力に関して機械が人間を上回る時代が到来しています。本稿ではここ数年のこの技術的発展を振り返るとともに、その知的財産業務への応用の一例として、特許調査における最新状況をお伝えします。特に従来の特許調査を人手による一連の情報処理と捉え、そこで行われている情報処理が深層学習によってどのように改善・代替されるかを議論し、深層学習の発展によって特許情報の利用およびそれに携わる人材の役割の未来について私見を述べたいと思います。

目次

1. AI が発明者になる時代
2. 自然言語処理の大前提
3. 深層学習の万能さ
4. 深層学習を利用した自然言語処理の発展
 4. 1 WORD2VEC
 4. 2 単語から文章へ：RNN/LSTM/GRU
 4. 3 TRANSFORMER：ATTENTION 革命
 4. 4 BERT/GPT3 大規模汎用言語モデルへの進化
 4. 5 今、機械はどれほどなのか？
5. 情報処理としての特許調査と、深層学習の応用
 5. 1 クエリ設計による情報抽出から、機械による類似性評価に
 5. 2 手動のタグ付けから自動のタグ付けへ
 5. 3 解釈という難問
6. 特許情報はどこへ向かうのか
7. まとめ

1. AI が発明者になる時代

この数年の間に、機械は絵を描き、音楽を奏で、言葉を紡ぐことができるようになりました。これまで人にしかできないと思われていたタスクをこなせる「人工知能」は、あらゆる業界で人口に膾炙するテクノロジーです。特許業界では、DABUS という人工知能システムを発明者とした特許出願がなされ⁽¹⁾、この出願に対して、南アフリカ特許庁が特許を認可したり⁽²⁾、オーストラリア連邦裁判所が人工知能を発明者として認める判決⁽³⁾を出したりと、大きな注目を集めました。英国では人工知能を発明者として認めないという判決

が出ました⁽⁴⁾。人工知能は、結晶構造の決定や素粒子加速器実験のためのデータ解析や、回路の消費電力最適化など、これまでデータのなアプローチがなされてこなかった研究開発の現場でも、当たり前利用されています。

人工知能が発明を提案するという時代が来ている中、特許に関わる様々な業務にも人工知能すなわち「深層ニューラルネットワークを用いた機械学習（深層学習）」の利用が進んでいます。筆者が関わってきた特許調査や特許情報分析は、その大きな変化の真只中にいるとあって良いでしょう。特許調査は、本質的には「人手で行う情報処理」です。これまで「人手」が必要であった最大の原因は、「発明」を表現する「文章」が「複雑」かつ「多様」であるために、言語と発明を理解した人間の方が、機械よりも上手であった事情があります。しかし、機械が複雑な言語を処理できるようになりつつある今、その状況は変わりつつあります。

本稿の前半では、深層学習を用いた自然言語解析の近年の発展を概観し、後半の議論に必要な技術的な背景を解説します。後半では特許調査業務を人手による情報処理と捉え、機械学習の発達によってその情報処理に起きている変化を説明します。そして、この変化から、特許情報がその本来の役割を取り戻す未来を共有し、属人的な知識やアイデアを自然言語という情報

[※] 共同創業者 Chief Operating Officer 博士（物理学）

で保存することが、組織や社会に有用なデータ資産を作り出すことを述べたいと思います。本稿を通じて現代的な自然言語処理の現状を共有し、読者が必要な文献を参照したり、新たなツールやサービスを検討したりする一助となれば幸いです。

なお筆者の浅学のため、数学的・技術的な厳密さに欠けていたり、また分野の急速な進歩のため、必要な文献・最新の文献を引用できていなかったりすることがあり得ます。諸兄弟姉からのご指摘を歓迎いたします。また、本稿の文責は筆者個人にあります。

2. 自然言語処理の大前提

さて、自然言語処理の目標は、機械に人間の言語を理解させることです。そのような機械（計算機）はどうしたら作れるのでしょうか？機械に人間の言語を理解させるというのはいささか文学的な問題設定なので、これを数学の問題に焼き直してみます。次の穴埋め問題を考えてみましょう [図 1 (a)]。

問題：次のカッコに当てはまる単語を選べ。

太郎は () を食べた。

- a. 机 b. りんご c. アメリカ d. 校庭

図 1 (a) 穴埋め問題

大勢の人にこの質問すると、100%の人が「りんご」と答えるでしょう。それはりんごという単語が食べ物という意味だと知っているからです。当然、「言語の意味を理解した」計算機には、この問題を人間と同じように解いて欲しいわけです。そこでこの問題が解ける機械の動作を考えてみます [図 1 (b)]。入力として、機械に問題文と選択肢が与えられた時、その出力はどうあるべきでしょうか？機械が人間と同じ確率でそれぞれの選択肢を正解として答えれば、その機械の出力は人間と区別ができないことになります。これを抽象化すると、自然言語処理の目標は次のような確率分布を求めることになります [図 1 (c)]。

$$P(w_n | w_1, w_2, \dots, w_{n-1})$$

つまり、先に与えられた $n-1$ 個の単語を見たときに、次 (n 番目) に来るべき単語に適切な確率を与えてくれる計算機を作りたいわけです。このように単語が「ある非自明な確率分布」からサンプリングされて言語が作られると考えて、その確率分布を確率的言語モデルと呼びます。このサンプリングの連続は、人間にとっては単語の意味に基づいて行われるはずなので、

「単語の意味は周りの単語で与えられる」ということもできます。これは分布仮説と呼ばれます⁽⁵⁾。

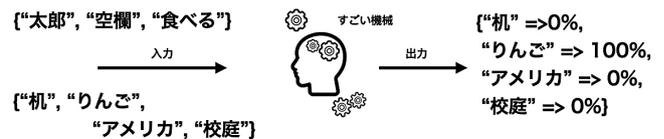


図 1 (b) 穴埋め問題を解ける機械の動作

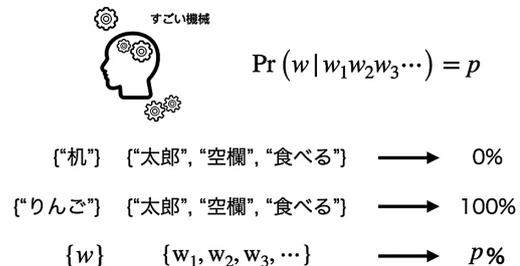


図 1 (c) 穴埋め問題が解ける機械の動作の抽象化：言語モデル

この言語モデルを何らかの公理から演繹的に構築することはおそらく不可能でしょう。現代的な自然言語処理は、言語モデルをデータから近似的に再現して、私たちの言語を計算機に処理させる技術です。言語は四則演算の対象にはなりませんから、自然言語処理のためには言語を数学的に表現して計算機に渡す必要があります。そのために言語はベクトルに表現されます。言語が従う確率分布は、このベクトルを変数に持つ関数としてモデル化されます。この確率分布と言語のベクトル表現をデータから構築するために、深層学習が利用されます⁽⁶⁾。

さて、深層学習の話に入る前に、深層学習以前の概念検索の考え方にも触れておきましょう。言語の確率分布は非常に複雑なので、近似をしないと手に負えません。そこで、最も簡単に

$$P(w_n | w_1, w_2, \dots, w_{n-1}) \sim P(w_n)$$

と周囲の単語の情報を無視してしましましょう。要するに、文章にその単語が出てくる確率を、データから実際に数えて計算してしましましょう、という戦略です。これはカウント・ベースや統計モデルなどと呼ばれたりします。よく知られている TF/IDF などはその典型例です⁽⁷⁾。

3. 深層学習の万能さ

従来の方法では、未知の言語の確率分布を、人間が手に負える関数にモデル化していました。例えば TF/IDF では、言語モデルを単語の出現確率として単純化していたわけです。

深層学習を利用した自然言語処理（ニューラル言語

モデル)は、言語の確率分布をそっくりそのままニューラルネットワークで表現してしまおうというものです⁽⁶⁾。これを支持するのは、Cybenkoによって証明された万能近似定理です⁽⁸⁾。この定理は、与えられた実数コンパクト集合上の有界な連続関数を、任意の精度で近似できるニューラルネットワークの存在を保証してくれます(もちろん、この定理は適切なネットワークの形状や必要なパラメータ数は教えてくれません)。言語の確率分布が実数変数の有界な連続関数だと仮定するならば、うまいニューラルネットワークを見つければ、与えられたデータを再現する言語モデルを良く近似できる可能性があります。自然言語処理の問題の複雑さを、全てニューラルネットワークの設計と学習法に押し付けているわけです。

筆者が初めてニューラル言語モデルのインパクトを耳にしたのは、構文解析についてでした。2016年にGoogleによって発表されたSyntaxNetは、様々な言語の品詞タグ付けで97%程度の精度を達成しました⁽⁹⁾。当時筆者は大学で理学の研究に従事していましたが、近隣のラボで構文解析の研究をしていた友人が「Googleがすべてを持っていった」と苦々しげに語っていたのを記憶しています。余談ですが、ニューラルネットワーク開発のための標準的なフレームワークであるTensorflow^{®(10)}がオープンソースとなった2016年はAmplifiedの創業が計画された年でもあります。

4. 深層学習を利用した自然言語処理の発展

この節ではニューラル言語モデルの近年の進展を具体的なモデルを挙げながら概観したいと思います。ここで感じていただきたいのは、ニューラルネットワークが言語モデルという関数そのものを表現しており、文章に現れる複雑な単語同士の相関を取り入れて、より長く複雑な文章が示す抽象的な概念の意味をベクト

ルで表現する仕組みを発展させてきたことです。

4. 1 Word2Vec

ニューラル言語モデルとして必ず名前が上がるのがWord2Vec (W2V)です⁽¹¹⁾。W2Vの構造は非常にシンプルです。W2VはCBOW (Continuous Bag-of-Words)とSkip-Gramと呼ばれる二つのモデルの総称です。

図2はCBOWの構造です。このCBOWは周囲の単語から中央の単語を推定することをモデル化しています。順に動作を追っていきましょう。まず入力として、語彙集合上のOne-Hotベクトル^(註1)で表現された複数の単語が文章の一部から用意されます。これらの単語に共通する行列が掛け算されて、それぞれのOne-hotベクトルが埋込ベクトルへと変換されます。この埋込ベクトルの平均値をとって中央の単語に対応する埋込ベクトルとします。次にこの埋込ベクトルを逆変換してOne-hotベクトルの次元に戻します。そしてSoftmax関数を通すことでこのベクトルを確率と解釈できるように変換します。この出力は、正解の単語に相当する成分に高い確率を持ったベクトルになっています。まさに周囲の単語が与えられた時に、中央に来べき単語に高い確率を割り当てる確率分布モデル $P(v_3|v_1, v_2, v_4, v_5)$ になっています。すなわち、穴埋め問題を解く機械が作れました、というわけです。

ニューラルネットワークの学習は、与えられた入力に対して正解の単語に高い確率が与えられるように、モデルの行列の成分を決めることです。この単語推定のタスクを既知のデータに行うことで、単語の非自明なベクトル表現(図中の v_i)が自動的に得られる仕組みになっています。(One-hotベクトルに対比して分散表現と呼ばれます。)W2Vはニューラルネットによる単語ベクトルの学習が効率的かつ高性能であることを

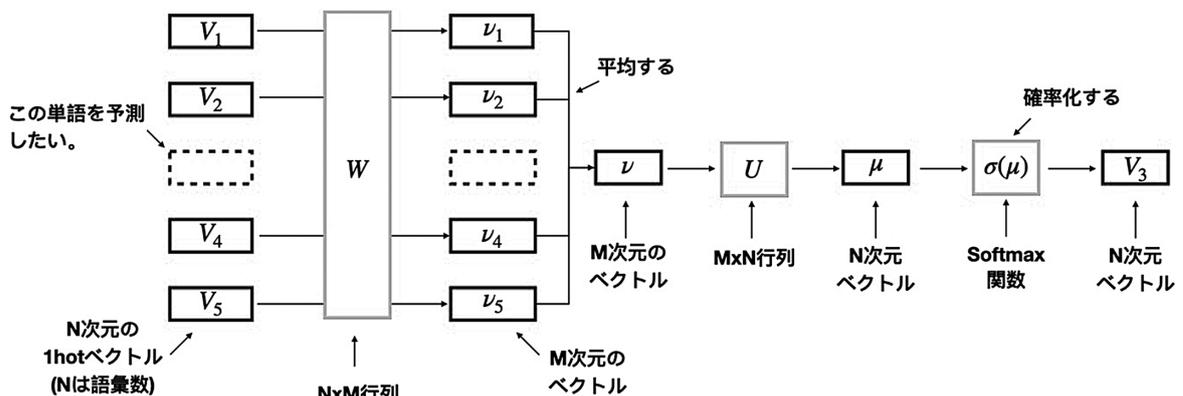


図2 Word2Vec : CBOW の模式図

示しました。

周囲の単語からその単語の意味を表すベクトル表現を獲得するという意味では、W2Vの埋込ベクトルは従来のLSA（潜在意味解析）で得られてきたものと本質的には同じものと思われま。実際W2Vは、平均化相互情報量の特異値分解に対応することが議論されています⁽¹²⁾。その意味でW2V自体は、従来の自然言語処理の仮定を覆すような発見ではなくて、新しい計算手法というべきだと思います。

4.2 単語から文章へ：RNN/LSTM/GRU

W2Vは穴埋め問題を解ける機械でした。一方、実際の応用を考えると、穴埋め問題は世の中の役に立ちそうにありません。「言語を理解した機械」には、文章を翻訳したり、質問文に回答したりして欲しいところです。このような複雑なタスクを機械が実行できるためには、一単語の情報をベクトルにするのではなく、文の情報をを持ったベクトルを作れる必要があります。例えば翻訳のタスクでは、翻訳前の文と翻訳後の文の情報が保存されていて欲しいからです。ここでは、文構造の情報を如何にベクトル化するかというのが問題になります。

どうすればいいのでしょうか？文を連続した単語の列と捉えて「並び順の情報を保持して学習」することで文の構造情報を保存する方法が、再帰型ニューラルネットワーク（Recurrent Neural Network: RNN）⁽¹³⁾です [図3 (a)]。RNNは文の中で前に出てきた単語の情報を「隠れ層」として保持し、後に出てくる単語の情報を既存の「隠れ層」の情報に追加していくことで、連続する単語の並び順の情報の表現を作っていきます。一つの文を処理し終わったときに出力される隠

れ層ベクトル h_i が文の情報を表現すると期待できます。

RNNではこの隠れ層の動作が重要になります。隠れ層も実体は有限の大きさの行列ですから、何も考えずに文章を学習させるとすぐにその容量を使い切ってしまう。また、学習の際の勾配消失という技術的な問題を避けられないことも知られていました。文章の中で何が重要な情報かを把握しながら、有限の大きさの行列で、その情報を効率的に表現する方法を学習させたいわけです。文の構造を長く埋め込むための工夫としてLSTM（Long-Short-term Memory）⁽¹⁴⁾の利用が進み、現在ではGRU（Gated Recurrent Unit）⁽¹⁵⁾など複雑なモデルに発展しています。

LSTMの再帰層の中身を追ってみます [図3 (b)]。LSTMの再帰層には三つの入力があります。まず、それまでに処理してきた単語の情報をを持った隠れ層ベクトルがあります。次に、これから処理する単語の情報である入力ベクトルがあります。これらの言語に関する情報に加えて、再帰層の状態を表す状態ベクトルがあります。

再帰層は、まず単語の入力ベクトルを隠れ層ベクトルと連結します。この連結された文章ベクトルは忘却ゲートと呼ばれる行列・活性化関数を通じて、状態ベクトルに影響を与えます。これは新しく入力された単語を処理するにあたって、これまで蓄積されてきた再帰層の情報を今回の処理の前にどの程度忘れるかという設定 f_t を与えます。次に、結合された文章ベクトルは、入力ゲートを通して、入力から作られる新しい状態ベクトル \tilde{C}_t とその重み i_t を作成します。これは先の忘却ゲートを通ってきた状態ベクトルと足し合わされて、新しい再帰層の状態ベクトル C_t を作ります。この動作を通じて、再帰層に入力されてきた一連の単語

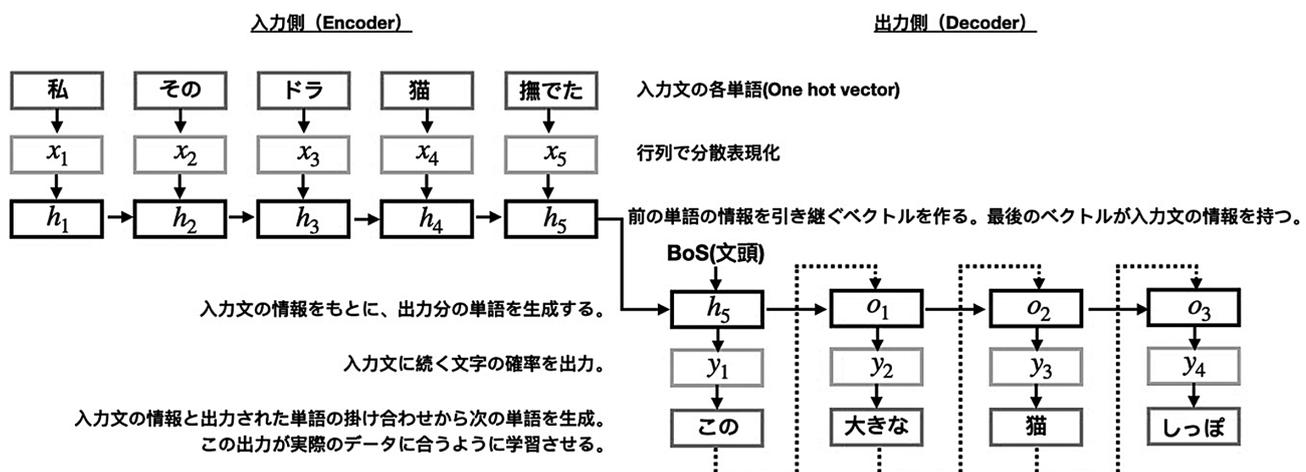


図3 (a) 再帰型ニューラルネットワークの模式図

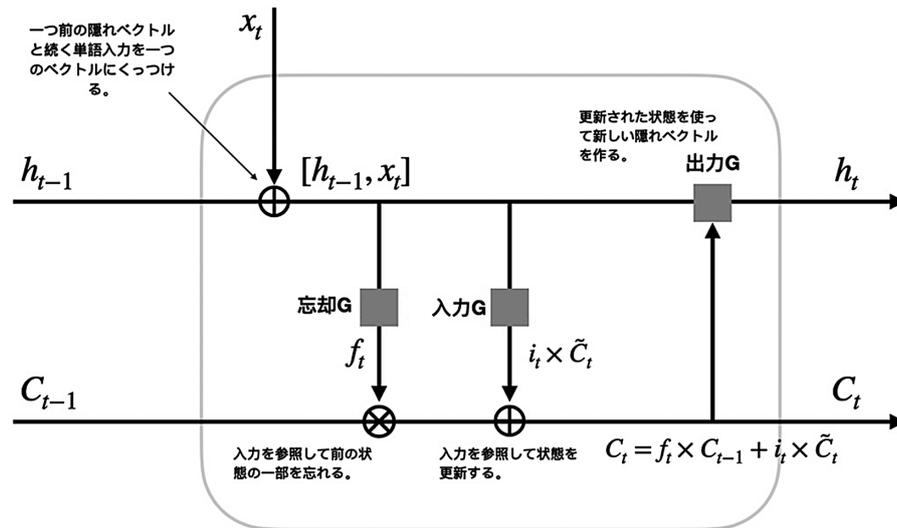


図 3 (b) LSTM の再帰構造の模式図

の情報を反映して、再帰層の状態が更新されます。この更新された再帰層の状態ベクトルと、連結された入力ベクトルから、再帰層の出力である新たな隠れ層ベクトルが構成されます。この出力が入力された文章に続く単語となるようにそれぞれのゲートのパラメータを学習させることで、入力文章から続く単語を推定する計算機となるわけです。つまり、文 (s) を入力して、それに続く単語 (w) を予測することで言語モデル $P(w|s)$ を表現できるわけです。

LSTM の重要なポイントは、文の情報を表すベクトルだけではなく、ニューラルネットワーク自体が保持しておくべき情報を状態ベクトルとして別に表現させたことです。状態ベクトルのおかげで、与えられた文の情報を全て覚えるのではなくて、重要な単語の関係性（相関）を優先して覚えているようなものです。こうして簡易的な記憶のような機構を設けることで、文の情報をベクトルに埋め込むということが実現できます。LSTM は言語だけでなく様々な時系列データに利用することができ、GRU (Gated Recurrent Unit) と共に様々なタスクに応用されています。

先に述べたとおり、RNN は文 (s) を入力して、それに続く単語 (w) を予測することで言語モデル $P(w|s)$ を設計できます。さらに、Encoder-Decoder と呼ばれる構造にすることで、特定のタスクに特化して文の埋め込みを行うことができます [図 3 (a)]。例えば、Encoder 側に原文を埋め込ませて、Decoder 側で原文の埋込ベクトルから翻訳文を予測させれば、翻訳機になります。Encoder 側に質問文を、Decoder 側に回答文を予測させれば対話機になります。この場合は、入力文 (s_1) から出力文 (s_2) を作る $P(s_2|s_1)$ を設計し

ています。

RNN の問題は、学習が逐次的な処理であるために、計算時間が文章の長さに比例して延びる上に、学習時に並列処理ができないという問題もありました。そのため大規模なデータを使ってより汎用な言語モデルを作る困難がありました。この問題を解決したのが次に述べる Transformer です。

4. 3 Transformer : Attention 革命

ニューラルネットワークを用いた自然言語処理の最新モデルは、この Transformer⁽¹⁶⁾ が基礎になっていると言っても過言ではないと思います。W2V は「短いコンテキスト以上の単語の間の情報を維持できない」という問題があり、LSTM では「複雑な再帰層の構造や逐次的な処理による計算コスト」という問題がありました。Transformer はこれらの問題を Self-Attention という仕組みを組み込んだ Encoder-Decoder ネットワークで解決しています [図 4 (a)]。

図 4 (a) を見ながら Transformer の動作の概要を順に追ってみましょう。特に重要な Self-Attention に注目して説明します。大きく分けて左側がエンコーダ、右側がデコーダです。Transformer は入力に二つの文のペアを使います。例えば、翻訳タスクに使うのならば、エンコーダ側に原文、デコーダ側に翻訳文といった具合です。出力文の一部の単語を抜いて (Mask)、その抜かれた単語を推定するというタスクによって学習をします。

まず各単語の文章中での位置の情報を保存するために、文中の位置を示す情報を位置の一価関数で付与します (Positional Encoding)。これのおかげで、RNN

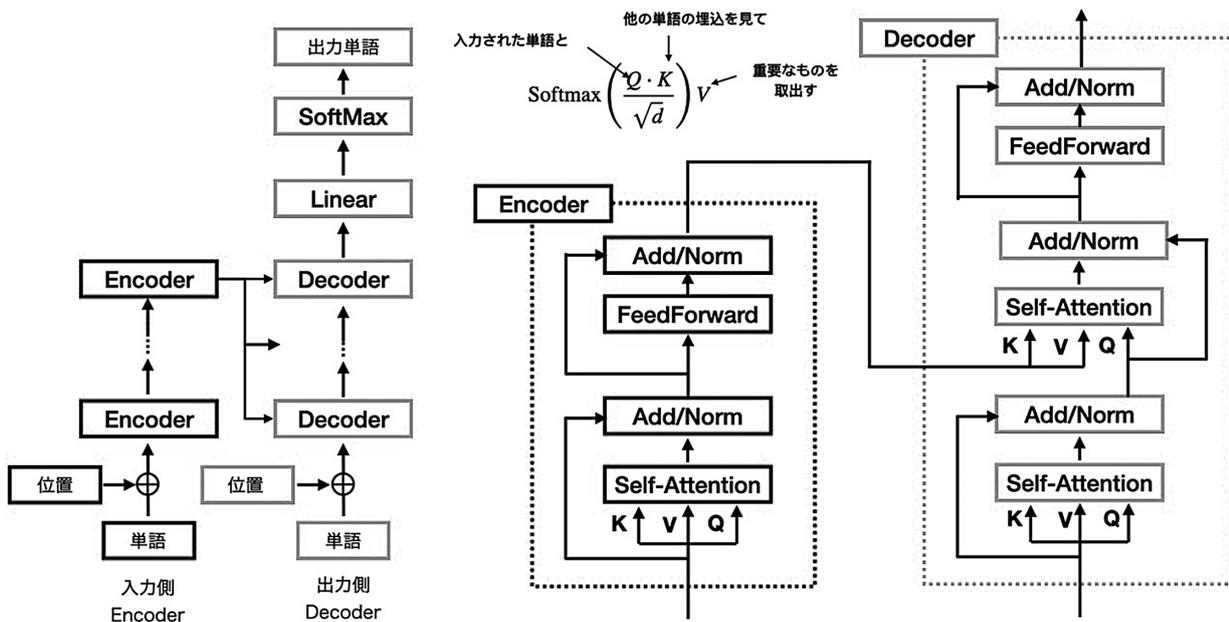


図 4 (a) Transformer のネットワーク模式図

のように順番に処理をしていくのではなく、文に含まれる複数の単語を一括処理できます。

入力された単語は Self-Attention^(注2) を含むサブ層を進みます。一つの Self-Attention 層は 3 つの行列 Q, K, V で構成されています [図 4 (b)]。Self-Attention に単語 x が入ると $q = Qx, k = Kx, v = Vx$ というベクトルが作られます。この Q, K, V のセットは記憶の役割を果たしています。簡単な具体例で説明してみます。 $x_1 =$ 「ラグドール」、 $x_2 =$ 「猫」、 $x_3 =$ 「深海魚」という三単語の入りがあったとしましょう。そうすると、 x_1, x_2, x_3 のそれぞれに対してベクトル q_i, k_i, v_i ($i=1, 2, 3$) のセットが作られます。 $x_1 =$ ラグドールの Self-

Attention の出力は、 q_1, k_1, v_1 だけでなく他の二つの入力に由来する k_2, v_2, k_3, v_3 を参照して構成されます。具体的には q_1 と k_1, k_2, k_3 それぞれとの内積で決まる Attention 重みを計算します。そうすると、(ラグドールは猫なので) $q_1 \cdot k_1$ と $q_1 \cdot k_2$ の値は大きく、(ラグドールと深海魚はあまり関係ないので) $q_1 \cdot k_3$ の値が小さくなります。すると、 $x_1 =$ 「ラグドール」に対する出力として、Self-Attention はそれぞれの単語の v に Attention 重みをかけて和を取ったものを作ります。今の場合には v_1 と v_2 が出力の主要なベクトルになります。この動作は「ラグドール」と「猫」の相関を記憶する動きを実現していて、ラグドールは猫の一品種

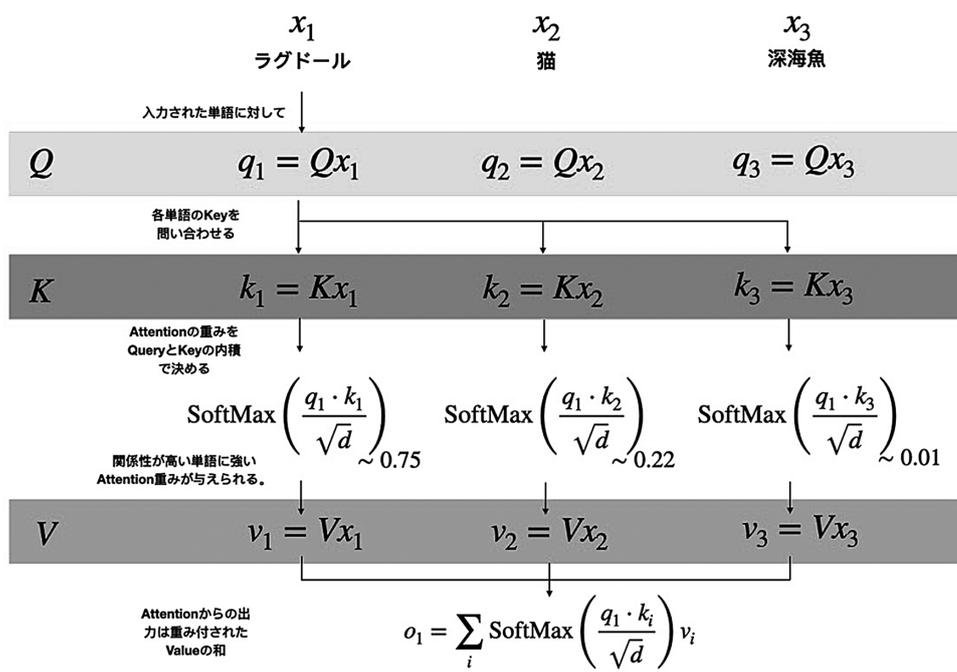


図 4 (b) Self-Attention のプロセス概要

という情報を元のベクトル表現に加えて出力しているわけですが。この Self-Attention 層を複数層重ねることで入力された文中の単語同士の様々な関係を反映して、それぞれの単語のベクトル表現に加えていきます。

Self-Attention のサブ層を進んでいくのはエンコーダ側・デコーダ側同様です。最後に単語の推定タスクのために、エンコーダ側の出力がデコーダ側に入力されています。これはデコード側の q に対して、エンコーダ側の k と v を利用する Self-Attention になっています。つまりデコーダ側の入力 q に対してエンコーダ側の複数の単語の相関を埋め込んだ単語ベクトルの中から、重要なものを呼び出すという動作になっています。

このような動作を既知のデータに対して学習することで、各層の行列 Q, K, V たちが学習されていきます。原論文では GPU8 個で 3.5 日かけて作ったモデルが当時の機械翻訳の SoTA^(注3) を達成しました。Transformer には他にも細かな工夫がたくさんありますが、それらは原論文を参照ください。

4. 4 BERT/GPT3 大規模汎用言語モデルへの進化

Transformer を基礎に作られた汎用的な言語モデルが BERT (Bidirectional Encoder Representations from Transformers) です⁽¹⁷⁾。BERT は Transformer のエンコーダを利用して、学習方法を改善して、様々なタスクに利用可能な汎用の言語モデルの構築を可能にしました。従来の言語モデルは、文が与えられた際に、それに続く単語を推定するタスクによって学習を行っていました。BERT はランダムにマスクされた単語を推定するタスクと、二つの文の組が与えられた時に、一方が他方に続く文かどうかを推定するタスクとの二つ学習方法を同時に学習することで言語モデルを構築します。これは文の前後関係という大局的な意味関係と、単語推定という局所的な意味関係を同時に学ぶことになっており、与えられた文に続く単語を予測するよりも、より広い意味関係を捉える言語モデルになっています。

BERT が重要であったのは、汎用の言語モデルの構築と、その言語モデルを利用した特定のタスクへの最適化 (Transfer Learning と呼ばれます) とが別個に行えることを示したことです。従来は翻訳なら翻訳のために、タスク固有のデータ (原文-翻訳文ペア、質問-回答文ペアなど) を用いて言語モデルを作るといったのが主流でした。BERT は上記の事前学習によ

って作られた汎用モデルを用いて、利用したいタスクのために追加学習をして、様々なタスクに対応して高い性能を発揮しました。

こうして大規模な事前学習によって汎用の言語モデルを構築するというトレンドが生まれ、その最大規模のものが GPT3 (Generative Pre-trained Transformer) です⁽¹⁸⁾。GPT3 はニューラルネットワークとしての層数が 96 層、パラメータ数で 1750 億という大きさのモデルです。このモデルは、単語数にして 5000 億余りのデータによって学習され、汎用の言語モデルとして機能しています。

4. 5 今、機械はどれほどなのか？

これまで見てきたように、ニューラル言語モデルは文中の単語の相関を埋め込むことで、再現すべき言語の確率分布をより高精度に近似することに成功しています。それでは現在の性能はどの程度なのでしょう？

自然言語処理の性能を評価する一般的な試験の一つとして GLUE (General Language Understanding Evaluation) があります⁽¹⁹⁾。GLUE では一般的な文章に対する 8 つの性能試験が準備されています。試験の内容は質問文と回答文に対する推論や二つの文の類似性比較などです。8 試験のスコアの平均値で言語モデルの性能を評価しています。

この評価指標の詳細はリンク先を参照していただくとして、この指標に基づく言語処理能力評価について、本稿執筆時点^(注4)では、人間は 16 位です。1 位は Baidu による ERNIE, 2 位は Alibaba の AliceMind, 3 位は Microsoft の DeBERTa と続いています。人間よりも上位のモデルの半数以上で BERT が基礎になっています。

このように一般的な言語処理という意味では、機械の自然言語処理の性能は人間と同等あるいはそれ以上に進化しています。大規模な汎用言語モデルとして Open-AI が 2019 年に発表した GPT-2, それに続いて 2020 年に発表された GPT-3 は、人間が書いたものと区別できない文章を生成し、大きな話題となりました。GPT-3 は API を通じてプログラミングコードや文章の自動生成に利用できます⁽²⁰⁾。

Amplified は特許の類似性評価というタスクに特化して、深層学習を利用した独自の言語モデルを構築しています。このモデルは任意の長さの明細書を、明細書同士の類似性評価が人間のそれを再現するように、

ベクトルに埋め込むことを実現しているユニークなモデルです。他にも特許調査にBERTなどの汎用言語モデルを利用する取り組みは報告が出てきています⁽³⁰⁾。本稿執筆中に、GoogleからTransformerを階層化した新しいモデルとしてHourglass⁽²¹⁾が発表されるなど、深層学習を用いた自然言語処理の発展は今も急速に進んでいます。

5. 情報処理としての特許調査と、深層学習の応用

これまで述べてきたように、深層学習を利用した自然言語処理は、すでに人間に比肩する言語能力を獲得しています。この技術は特許調査に対して、どのように活かすことができるのでしょうか？特許調査に対して自然言語処理を利用するときにも、どのようなタスクで利用するかが重要になります。一般的に特許調査には「出願前調査」「有効性調査」「侵害防止調査」「収集調査」といった業務があります。これらの特許調査を情報処理と捉えたとき、人間はどのような情報処理を行っているのでしょうか？そしてそれぞれの情報処理は深層学習による言語モデルによって、どの程度代替あるいは改善が可能なのでしょうか？本節では、特許調査に含まれる情報処理のそれぞれについてこの言語モデルをどのように活用できるかAmplifiedの事例をもとに説明します。

5. 1 クエリ設計による情報抽出から、機械による類似性評価に

特許調査には、コストという制約条件があります。特許の数は膨大にありますが、所詮は有限個の文章集合です。したがって、原理的には、無限の時間とお金があれば、全ての特許を人間が緻密に読んで、どんな特許調査も完璧に行うことができます。もちろん、そんなことは起こりません。限られた時間とお金の範囲で、必要な情報にたどり着かなければなりません。

そのために、「調査範囲」を設計することになります。タスクとしては、与えられた調査主題からキーワードや特許分類を選定し、データベースに入力するクエリを設計することです。これは、探している文献は、「クエリの条件のどれかに当てはまるはずだ」という仮定によって、調査対象とするデータを限定します。その「クエリ」が目的の文献を捉えているかどうかはわからないため、「クエリ」の妥当性が重要視されます。コスト面での制約から調査は必ず不完全な

で^(注5)、その不完全さを特許分類やキーワードの選定の妥当性で正当化するわけです。これが「調査主題をクエリに書き換える」という人間による情報処理になります。

この「調査範囲」を決めるクエリ設計は、深層学習による文章の類似評価で置き換えられつつあります。調査の目的にもよりますが、少なくとも「出願前調査」および「無効資料調査」については、そのトレンドは不可逆に始まっています^(注6)。例えばAmplifiedが深層学習によって構築した言語モデルは、明細書の一つのベクトルに変換する能力を持ちます。このベクトルは、特許同士の類似性について、その内積が人間の判断を再現するような表現として獲得されています。そのため、特許同士の相対的な類似性は、言語モデルの構築の時点で全て計算が完了していることになります。このベクトル化と類似性評価を用いて、任意の文章と既存の特許との類似性を比較して、全特許をランク付けできます。このランク付けの上位がAIによって提示される類似文献のリストということになります。機械が調査主題を自然文として受け取り、適切な類似特許のリストを返してくれるのであれば、調査主題の自然文からクエリに焼き直す情報処理を代替できます。

このタスクの性能評価の一例としてAmplifiedでのベンチマークを紹介します(図5)。Amplifiedのベンチマークは、国際特許出願に対して発行されるサーチレポートの引用文献を、どの程度見つけられるか、という試験になっています。サーチャーがサーチレポートを作成する場合、種々のクエリを掛け合わせて500~1000件程度の先行特許を確認しています^(注7)。これらの文献から、X文献があるかないかを判断する作業をしています。Amplifiedはファミリーカウントで7500万件程度のデータベースを、調査対象である公報に類似した順にソートします。その際に、上位1000件以内に少なくとも一つX文献を再発見できるか、ということの評価をしています。

X文献が引用されているサーチレポート150万件を対象に試験を行ったところ、実に60%のケースで1000件以内にX文献が抽出されています。また、対象公報とX文献とが少なくとも一つ共通のCPCサブクラスを持っているケースを対象にすると、100万件程度が対象となり、1000件以内への絞り込みの成功率は95%を超えています。これらは技術分野を問わ

ず行われており、統計的に有意な結果です。深層学習に基づく類似性評価によって類似順に並べられた上位1000件は、人間が検索式によって設定した調査範囲と同等にX文献を含むということを、この結果は実証しています。

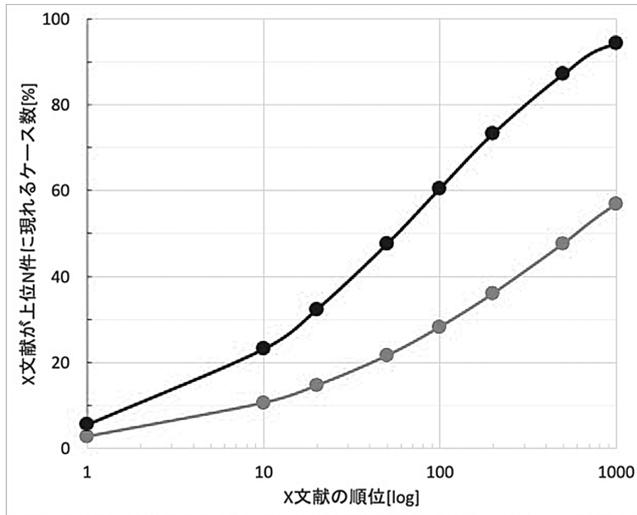


図5 Amplifiedのサーチレポートベンチマーク

このベンチマーク結果を実際の業務の言葉で表現すると、Amplifiedに特許番号を入力すれば(1クリック)、60%のケースでは1000件以内に一つX文献が含まれているということです。分類の絞り込みを行えば(2クリック)、95%になります。これらのケースでは、データベースに収録された全7500万件の公報ファミリのわずか0.0013%に、一切人間の手を加えずに正解を持ってこられるわけです。この性能は年々向上しています。実際の業務では、この深層学習による類似性評価に、通常のキーワード検索を組み合わせることで、さらに高い性能を発揮します。サーチレポートでいうところの関連文献(X/Y)を見つけるというタスクについては、極めて短時間で実施できることが実証されています。このように深層学習に基づく類似性評価を用いると、クエリ設計なく、類似する特許の集合を、ほぼノイズなく得られるようになっています。

このニューラル言語モデルを用いた類似特許の探索は今後必須のツールになると期待されます。それは昨今の技術発展に対して「クエリの妥当性」を評価することが不可能になりつつあるからです。その主たる原因は年々増える特許出願と、新しいビジネスの出現に、既存の特許情報の整理方法が追いついていないためです。デバイス間通信(D2D)によるサービス、仮想空間での情報処理、量子情報や人工知能などは分類が整理されていなかったり、キーワードが多様になっ

たり、いくらでも抽象的な表現が可能であったり、特に調べにくい分野でしょう。そのためクエリベースの検索で数を絞り込むことが難しくなり、クエリの結果に含まれるノイズが増え、調査が許容できるコストに収まらなくなりつつあります。特許庁であっても審査に同様の問題を抱えており、審査に人工知能を取り入れようというのはまさにこの問題解決を目指したものでしょう⁽²²⁾。

5.2 手動のタグ付けから自動のタグ付けへ

特許調査は特定の発明やアイデアに類似した1件の文献を見つけることだけが目標ではありません。パテントランドスケープという単語が流行って久しいですが、古くから特許戦略を考える上で、ビジネス情報と特許情報が連携したデータ資産を維持するという取り組みは、重要な企業活動です。「収集調査」はそのデータ資産の構築のために行われるものです⁽²³⁾。

「収集調査」では、大量の特許に対して、その目的に応じてメタデータを付与していきます。付与されたメタデータは自社特許に対する製品タグや技術タグなど、それぞれの企業に特有の情報です。こうして作られたデータを何らかの方法で可視化して解釈することでインテリジェンスを獲得するのがランドスケープの考え方です。

パテントランドスケープの難しさは、「データ作成の困難さ」につきます。一般的なマーケティングや金融などのデータ分析では、巨大なデータレイクから目的に応じたデータマートを設計されたプロセスによって作ることができ、データの定期的な更新と可視化が自動化して行われています。それは多くの場合で、ユーザーの性別や年齢、サイト内の滞在時間や資産額や取引頻度など、生データからの加工が一意かつ機械的に行えるケースがほとんどだからです。

一方、特許データの場合は、「ある技術に関連する特許を一括してデータベースから抽出する」という作業が一筋縄ではいきません。例えば「量子コンピュータに関するハードウェア技術」と言われたときに、どんなクエリを書けば良いのでしょうか？調査対象をクエリで表現する場合、クエリの結果に必ず無関係なノイズがそれなりに入ってきてしまいます。クエリで分離できないデータは人が綺麗にするしかありません。千件単位の特許から適切に関連する特許により分けていくのは、大変な作業でありコストがかかります。

うまく特許文献のデータが抽出できても、それだけでは意味がありません。単に特許が分布している図を見ても何も言えないでしょう。ビジネス上何が重要なのが見えるためには、製品や事業の情報が特許と紐づいていなければなりません。この情報の紐付けはタグによって行われるのが通常ですが、このデータを作るのも大変です。自社の特許については製品や事業との関わりを理解してタグ付けをしたり、発明者の人事情報などから整理したりできますが、他社特許となるとそう簡単ではありません。

このように大量の特許を前に、一つ一つの特許に「タグを付ける作業」は特許調査における人間の情報処理の一つです。このタスクも、人工知能による置き換えが進んでいます。このタグ付けのタスクは、「分類分け」のタスクとして機械学習を利用できます。最も単純な例は、事前に構築した言語モデルの上で、あるタグを付けるか付けないかという判断を、ユーザーからの教師データを基に学習させることです。

Amplified の場合、明細書から構築した言語モデルの上に、ユーザーが分類分けした特許リストをフィードして学習を行います。この学習に基づいて、未知の特許集合に対して分類分けを付与することができます(図6)。ここでの事例では「物理系の方式によらず、Qubit ハードウェアをクレームした特許かどうか」というタグ付けを、400 件の教師データをもとに、未知の 1000 件の特許に行ったものです。この性能評価は人間が 1000 件にタグを付与したデータとの比較で行われます。機械が推定した結果全てのうち、人間と同じタグ付が行われた性能 (Precision) は 90% 程度で

す。機械がタグをつけるべきと推定したもののうちの正解の割合 (Recall) は 93% 程度です。一般にこの二つの数値から定義される F1 スコアでの性能評価が適当であり、このケースでは F1 は 0.92 です。一般に F1 が 0.9 を超えると人間が行なった結果と区別がつかないと言われています。

実際の業務においては、付与されるタグはユーザー毎に異なるため、ユーザー毎に教師データが必要になります。例えば、大雑把なクエリで切り出した 1500 件を用意して、そのうちの自社と重要な競合の特許 2-300 件を選んで、人手でタグ付けを行います。そうして作ったデータをアップロードして分類器を学習させれば、残りの 1500 件にごく短時間で自動的にタグをつけることができるというわけです。Amplified の場合、この学習とタグ付けにかかる時間は 2-3 分です。一度教師データを作って、そのデータに基づくタグ付けの Fine Tuning ができていれば、その分類器は繰り返し利用することができます。新規の公開公報を同じ分類器でタグ付けしていくことができるので、定期的にマクロな視点で競合特許をモニタリングするにあたっては、こうした機械学習を利用したデータ整備の自動化が必須になってきています。

分類分けタスクを高精度かつ高速に行うためには、特許データの上で基礎となる言語モデルを事前に構築しておく必要があります。Amplified のようにその言語モデルを事前学習で獲得できていれば、個々のユーザーの要求に応じた分類器を Fine Tuning によって実現できます。筆者はかつて特許調査サービスとして、特定の技術分野に関してタグ付けを人手で行い、その

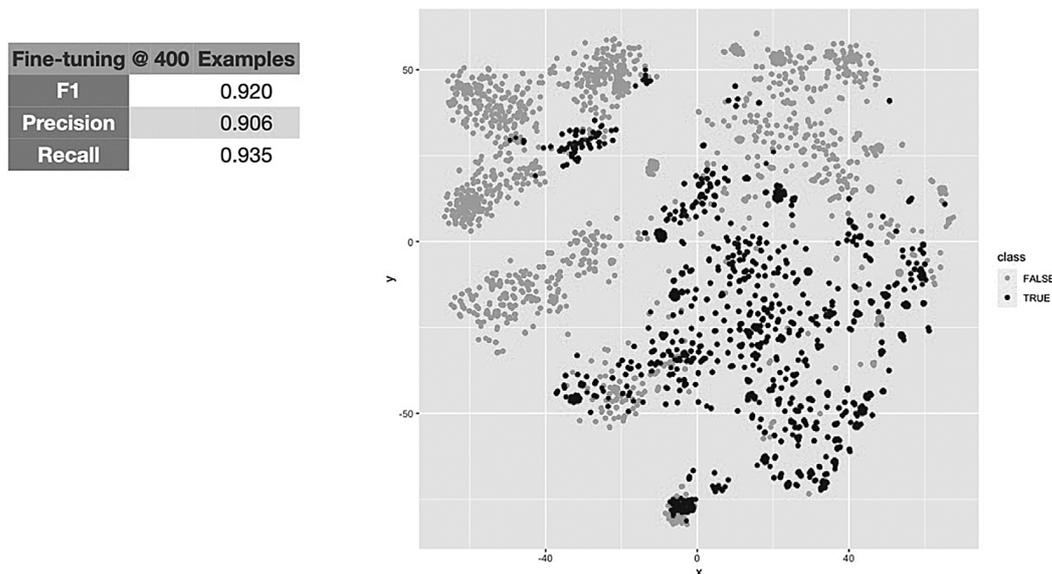


図6 Amplified の基本言語モデルの FineTuning による 2 クラス分類器の結果

データに基づくランドスケープレポートを提供していることがあります。こういったサービスは典型的には1000~2000件程度の特許集合を分析の対象としたもので、少なくとも2-300万円程度の費用がかかります。当時を振り返ると、今手元に存在する分類器が当時あれば、もっと楽により良いレポートが書けただろうと回顧するばかりです。現代では、ユーザー側がある程度の教師データを用意しておけば、タグ付けを機械学習で行うことは容易く、多少の間違いはデータの定期的な運用の中で修正していく程度で十分に対応できる水準に到達しつつあります。この修正を反映させることで分類器自体の精度も向上していきます。タグ付けの典型例は欧州特許庁による特許分類の自動付与が挙げられるでしょう⁽²⁴⁾。

5.3 解釈という難問

特許調査では、文意を解釈する必要が出てきます。例えば「侵害防止調査」ではそれが顕著でしょう。侵害防止調査は、ある製品を権利範囲に含むような特許がないかどうかを確認する調査です。キーワードによって抽出した特許を読み、特にクレームという抽象的な文章が、手元の製品と関係があるかどうかを解釈して判断しなければなりません。

この文意の解釈は、人間が行う自然言語処理の中でも最も高度なものの一つでしょう。クレームが手元の製品を含みうるかどうかを判断するときに、人間は二つの情報を使います。一つは、そのクレームを説明する明細書の内容です。明細書の実施例や図面に書かれた情報はクレームが何を意味するか、より具体的な情報を提供します。その具体的な情報（例えば実施例）を参考にして、クレームの言葉の意味を解釈します。明細書は、発明者が想定した具体例を教えてくれるという意味で、クレームの外延を提供してくれます。もう一つは、その製品について自分が持っている知識です。自分の頭の中にある製品やそこに使われている技術の知識を参照すれば、クレームに自分たちの製品は侵害しようのない限定がついていることや、自分たちの製品とは関連するキーワードの意味が異なることを、瞬時に判断できるでしょう。あるいは明細書に明示的に記載されていない知識を利用することで、クレームの表現の内包をも把握できるでしょう。こう

したクレームの表現の上位概念や下位概念を理解することで、人間はクレームを解釈して、その要素と製品との対比ができるわけです。つまり、「任意の製品とクレームを比較する」という情報処理が行われています。ここで任意の製品の情報は自然文ではなく、図面、仕様書、写真や動画などで与えられるので、より正しくは「任意の形式のデータの内容と、クレーム（自然言語）を比較する」という情報処理と言ってもいいでしょう。

一般に、人工知能、すなわち深層ニューラルネットワークを用いた機械学習は、データからデータを生成する確率分布を近似したり、データに潜むパターンを抽出したりする計算機に過ぎません。その意味で、学習に使ったデータとは無関係な情報を利用しないと解決できない問題には太刀打ちできません。製品とクレームを比較する人工知能は、製品情報を入力として受け取るでしょう。この場合、人工知能は、製品情報と、クレームや明細書・図面の内容を繋ぐ情報を学習していなければなりません。そしてその情報は、ある製品を権利範囲に含むようなクレームはどれか、というデータであるべきです。そのような情報を事前の学習のために大量に準備することは難しいでしょう。つまり、クレーム解釈が難しい原因は、深層学習の技術的問題もさることながら、むしろ機械がパターンを探すためのデータが存在しないことがあります。

この問題に対する解決策としては、自社の製品の情報（図面や写真や製品仕様書）と自社特許を関係づけたデータを用いて、何らかの学習を実施することが考えられます。しかしながら製品情報とは、その量も様式も様々です。汎用的にこの「解釈」を特許に対して実行できるニューラル言語モデルや、タスク最適化されたモデルを筆者は知りません。筆者は、この「解釈」という情報処理は、数年は人間の優位性が続くのではないかと考えています。現状で実務家が検討できる将来への準備は、どのような情報を特許情報と紐づけておくと、こうした解釈が可能になりうるかを検討し、日常業務における情報の保持方法を考えることでしょう。Amplifiedはユーザーが特許を読んで得た知識をシステム上に保存し、自動的にその知識と特許を紐づけていくKnowledge Automationという機能を備えています。ここでいう知識とは、「クレームのどこが新しいのか」とか「実施例の重要箇所」といった情報が該当します。このような情報の蓄積が特許の解

積を実現してくれるかは未知ですが、少なくともこれらの蓄積された情報が、特許に書かれていない知識を機械に提供することは間違いありません。将来的にはこれらの蓄積されたデータを活用して、重要特許をスクリーニングしていくことが可能になるでしょう。

この解釈の難しさから、現状でも侵害防止調査は人工知能の活用が難しい状況にあります。多くの企業で、侵害防止調査は、「あるクエリで定義された特許集合のクレームを全件確認するタスク」と定義されていることがほとんどです。全件確認することがタスクの要件である以上、その手間を減らすことは不可能です。そのため、現状の実務では、既知の問題特許を基に、言語モデルによる類似性評価を使って、他の問題特許を探すということが行われています。特にクエリに基づく全件調査と言語モデルによる類似性評価を並行して行くと、不運にもクエリに含まれていなかった特許を人工知能が提案してくれます。人間と機械とが協働することで、侵害防止調査の効率ではなく、質を向上していくというのが現状の最適解ではないかと思えます。

本節では、特許調査を人間による情報処理と考えた際に、どのような処理が内在しており、何が既存の深層学習による自然言語処理によって代替・改善しうるかを述べました。最後に、自然言語処理の発展によって、特許情報にどのような未来が待っているかについて、私見を述べます。

6. 特許情報はどこへ向かうのか

特許法の目的について書籍を参照すると、「新しい技術を社会に公表する代償として独占権を与えて技術を保護し、同時に他の発明を刺激して社会全体の技術を引き上げる」ことであると、様々な表現で語られています。特許情報、つまり公報は、その発明を社会に対して開示する役割を担うものです⁽²⁵⁾。そしてそれは、何か新しい発明に繋がる目的に利用されて欲しいはずですが。

特許公報には過去から連綿と続く、発明者たちのアイデアが潜んでいます。筆者自身も大学において発明検討と特許出願するにあたり、たくさんの公報を調べ、参考にした経験があります。しかしそれは、筆者自身がある程度特許調査のノウハウを持っていたからであり、周囲の研究者は特許公報に全く興味を持っていませんでした。大企業の研究開発部門でも、技術者

が特許を読んでもくれない、研究者が特許を調べてくれない、という声は知財担当者からよく聞きます。それは特許が、調べるにも、読むにも、技術者・研究者の立場ではあまりにも不便だからです。

また組織の規模によっては、特許実務を担う部門のリソースが十分ではありません。大学の研究現場や、専任の特許担当者がいない中小企業やスタートアップにとっては、特許を調べることは明らかにコストであり、偶発的な発明や、突発的な外注のための予算すら確保できないことはざらです。そのため、インフラと人員を持つ大企業との間には特許情報の利用に大きな差があります。つまり、特許情報は、一部の大企業を除き、ほとんど本来の目的のために使われていないのです。

そして増えすぎた、調べにくい特許情報は、特許庁さえも困らせています。米国会計監査院の2016年レポートでは、67%の審査官が必要な先行技術調査をするには時間が足りないと答えているそうです⁽²⁶⁾。審査における不十分な特許調査は、特許の品質を下げ、社会的なコストを増大させているのではないかと、という指摘は長くあります⁽²⁷⁾。それでは審査官が使える時間を2倍にすれば良いのでしょうか？そのコストは年間6.6億ドル（日本円で約700億円）に及ぶというのが試算です⁽²⁷⁾。そうするとやり方を変えるしかありません。そのためにUSPTOは「AIに投資する」と明言していますが、ユーザーである審査官のスキルも問題になり得ます。USPTOにおけるAI利用に関する論文⁽²⁸⁾では、従来のキーワード検索に慣れている状況でAIツールを使おうとする時には、新しい技能が必要であるということが指摘されています。このUSPTOとHarvard Business Schoolの共同研究では、HBSの学生に先行技術調査をやらせたところ、コンピューターサイエンスの知識の有無によって、AIツールを利用した際の先行技術文献を発見する効率に大きな差があったということです。この結果は、人工知能を用いた特許情報の活用というパラダイムシフトにおいては、調査をする私たち自身が人工知能を理解し、使いこなすスキルが必要になることを示唆しています。

このスキルはどのようなものなのでしょう？特許調査に用いられる人工知能とは、明細書を学習して作られた言語モデルでした。このモデルへの入力には必然的に自然言語になります。すなわち、発明という複雑な技術的対象を自然言語で表現するスキルが、深層学

習の発展に伴って重要になってくると予想されます。これは弁理士という実務家の能力が大きく寄与する部分でしょう。深層学習の発展によって、自分のアイデアや発明を文章で表現できれば、自分が読むべき特許を簡単に調べられるようになりつつあります。弁理士は発明者のアイデアを言葉にすると同時に、適切な先行文献を見つけ出し、より良い権利の設計や新たな着想を促せるでしょう。それは特許情報の本来の役割を取り戻すことになり、同時に特許審査の迅速化と特許権の安定化にも貢献するはずで

そして、深層学習の時代において、特許情報の役割はますます重要になるはずで

す。現代社会において無料で手に入る大規模な技術文献のデータソースは特許情報か、学術論文のプレプリントでしょう。特許情報は、それ自体がデータとして重要な公共財産であり、技術文献の言語モデル構築のための基本データです。このデータを学習して作られるツールは、何も特許調査を目的にしたものに限りません。冒頭に述べたDABUSのように既知の明細書の中から未知のアイデアを抽出してくるようなアプリケーションや、クレームから明細書を生成するようなアプリケーションも生まれてくるでしょう。このような人工知能を利用したデータツールを業務に利用するためには、業務に関わる情報を、人間だけでなく機械にも理解できる形式で蓄積することが必要です⁽²⁹⁾。特に技術者・研究者の知見を情報として資産化するために、製品アイデアや発明提案を自然文や図面として蓄積していくことは企業活動として必定です。人の頭の中に保存されていた知識を、人から引き出して正確な自然文に書き換えるタスクは、まだまだ機械にはできないでしょう。今後、知的財産に関わる人材は、創造的組織において、知識やアイデアといった属人的な情報をデータ化する役割を担っていくのだらうと想像しております。

7. まとめ

本稿ではここ数年の深層学習を利用した自然言語処理技術の進展を概観し、基本的な自然言語処理の考え方である言語モデルと、それを表現する計算機としてのニューラルネットワークを紹介してきました。特にWord2Vecのような単純な単語のベクトル表現を獲得するものから、TransformerやBERTのように文章中の単語同士の複雑な相関のベクトル表現を獲得するモデルへの変遷を追いました。これらの最新の人工知

能の自然言語処理能力はすでに人間を上回っており、その特許調査への応用を議論しました。特に特許調査を手での情報処理として捉え、「調査主題のクエリ化」が「人工知能による類似評価」に、「手動タグ付け」が「自動タグ付け」へと人工知能の応用が実用に達していることをAmplifiedの事例をもとに示しました。「文章解釈」の困難さを述べ、それを可能にする組織内でのデータ整備の必要性を述べるとともに、特許情報に関わる人材の役割が属人的な知識やアイデアを自然言語にデータ化・整備することへと変わっていくであろうことを提案しました。本稿執筆時点でも様々なアプリケーションが生まれています。自然言語処理技術の発展によって、特許制度の本来の仕組みがより良く機能していくことを期待しています。

(注)

- (1) One-hot ベクトルとは一つの成分だけが1で他の成分が0であるようなベクトルです。文章データは、事前処理によって単語の集合として保持されます。ニューラルネットに入力される単語（トークン）は、集合の何番目の要素かという情報で指定されるので、One-hot ベクトルで表現できます。
- (2) Self-Attention というものは一般に Attention と呼ばれる機構の特別な例です。“Self” は Attention 層の行列 QKV と作用する入力、Q と KV に対するものが同じ入力であることを意味しているそうです。
- (3) SoTA: State of the art の略。
- (4) 2021 年 11 月
- (5) 正確には調査の完全性を証明できないというべきかもしれませんが。
- (6) Amplified ユーザーの多くがこの目的のために利用しており、現場の感触として、まず AI で調査してみるという手順が浸透しつつあります。
- (7) ランダムに選んだサーチラポートに記載されている検索式を再現した時のおおよその値です。文献数は技術分野に依存します。

(参考文献)

- (1) The Artificial Inventor Project, <https://artificialinventor.com/>
- (2) “DABUS: South Africa issues first-ever patent with AI inventor”, <https://www.managingip.com/article/b1sx9mh1m35rd9/dabus-south-africa-issues-first-ever-patent-with-ai-inventor>
- (3) Commissioner to appeal court decision allowing artificial intelligence to be an inventor, <https://www.ipaustralia.gov.au/about-us/news-and-community/news/commissioner-appeal-court-decision-allowing-artificial-intelligence>
- (4) Stephen L Thaler v The Comptroller-General of Patents,

- Designs And Trade Marks, <https://www.bailii.org/ew/cases/EWHC/Patents/2020/2412.html>
- (5) Speech and Language Processing, <https://web.stanford.edu/~jurafsky/slp3/>
- (6) Bengio, Y., “A Neural Probabilistic Language Model” *Journal of Machine Learning Research* 3 (2003) 1137-1155
- (7) An introduction to latent semantic analysis, <https://doi.org/10.1080/01638539809545028>
- (8) Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems* 2, 303-314 (1989). <https://doi.org/10.1007/BF02551274>
- (9) Ardor, D. and et al., “Globally Normalized Transition-Based Neural Networks”, <https://arxiv.org/abs/1603.06042> (2016)
- (10) Martin Abadi, et. al., “TensorFlow: A system for large-scale machine learning”, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association (2016), pp.265-283
- (11) Mikolov, T. and et al., “Efficient Estimation of Word Representations in Vector Space”, <https://arxiv.org/abs/1301.3781> (2013)
- (12) O. Levy, and Y. Goldberg, “Neural Word Embedding as Implicit Matrix Factorization” *Advances in Neural Information Processing Systems (NIPS)*, pages 2177-2185, 2014.
- (13) Sutskever, I., Vinyals, O., and Le, Q., “Sequence to Sequence Learning with Neural Networks”, <https://arxiv.org/abs/1409.3215>
- (14) Sepp Hochreiter, Jurgen Schmidhuber “Long Short-Term Memory”, *Neural Computation* (1997) 9 (8) : 1735-1780.
- (15) Cho, K., et. al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, <https://arxiv.org/abs/1406.1078>
- (16) Vaswani, A. and et al., “Attention is all you need”, <https://arxiv.org/abs/1706.03762>
- (17) Devlin, J. and et al., “BERT: Pre-training Deep Bidirectional Transformers for Language Understanding”, <https://arxiv.org/abs/1810.04805>
- (18) Brown, T., et., al., “Language Models are Few-Shot Learners”, <https://arxiv.org/abs/2005.14165>
- (19) GLUE Benchmark, <https://gluebenchmark.com/>
- (20) OpenAI API, <https://beta.openai.com/>
- (21) Nawrot, P., et., al., “Hierarchical Transformers Are More Efficient Language Models”, <https://arxiv.org/abs/2110.13711>
- (22) A.Krishna et. al., The 2016 AAAI Fall Symposium Series, Cognitive Assistance in Government and Public Sector Applications, Technical Report FS-16-02
- (23) 増山博昭 “知的財産戦略経営” 日経 BP (2006) ; 野崎篤 “特許情報分析とパテントマップ作成入門” 発明推進協会 (2016) ; Jieun Kim, Buyong Jeong, & Daejung Kim “Patent Analytics” Springer (2021)
- (24) “Automatic categorisation applications at the European patent office”, *World Patent Information*, Vol.24, 3, 187-196 (2002)
- “The EPO’s experience in machine learning/automation” https://www.wipo.int/edocs/mdocs/classifications/en/ipc_ce_52/ipc_ce_52_p7.pdf
- “Large-Scale Patent Classification at the European Patent Office” <https://2019.semantics.cc/large-scale-patent-classification-european-patent-office>
- (25) 武田和彦 “特許の知識” ダイアモンド社 ; Janice M. Mueller “An Introduction to Patent Law” ASPEN,
- (26) U.S. Government Accountability Office, GAO-16-479, “Patent Office Should Strengthen Search Capabilities and Better Monitor Examiners’ Work” (2016); Michael D. Frakes, Melissa F. Wasserman, “Is the Time Allocated to Review Patent Applications Inducing Examiners to Grant Invalid Patents? Evidence from Microlevel Application Data”, *The Review of Economics and Statistics* (2017) 99 (3) : 550-563.
- (27) Michael D. Frakes, Melissa F. Wasserman, “Irrational Ignorance at the Patent Office”, *Vanderbilt Law Review*, Vol. 72, 2019, Forthcoming; Duke Law School Public Law & Legal Theory Series No.2018-64;
- (28) A.Krishna et. al., The 2016 AAAI Fall Symposium Series, Cognitive Assistance in Government and Public Sector Applications, Technical Report FS-16-02; Arti K. Rai, “Machine Learning at the Patent Office”, *Iowa Law Review*, Vol.104, 2617
- (29) Sara Brown, “How to built a data driven company”, MIT Management Sloan School, <https://mitsloan.mit.edu/ideas-made-to-matter/how-to-build-a-data-driven-company>; Randy Bean, “Why Is It So Hard to Become a Data-Driven Company?”, *Harvard Business Review*, <https://hbr.org/2021/02/why-is-it-so-hard-to-become-a-data-driven-company>
- (30) Bekamiri, H., Hain, S.D., and Jurowetzki, R., “PatentSBERTa: A Deep NLP based Hybrid Model for Patent Distance and Classification using Augmented SBERT”, <https://arxiv.org/abs/2103.11933>;
- Althammer, S., et., al., “Linguistically Informed Masking for Representation Learning in the Patent Domain”, <https://arxiv.org/abs/2106.05768>;
- Voskuil, K., and Verberne, S., “Improving reference mining in patents with BERT”, <https://arxiv.org/abs/2101.01039>;
- Freunek, M., and Bodmer, A., “BERT based patent novelty search by training claims to their own description”, <https://arxiv.org/abs/2103.01126>
- Freunek, M., and Bodmer, A., “BERT based freedom to operate patent analysis”, <https://arxiv.org/abs/2105.00817>

(原稿受領 2021.11.12)