

データマネジメント業務における  
AI活用検討のためのモデル試作

日本製薬工業協会  
データサイエンス部会

2023年度タスクフォース 1-1 「データマネジメントにおける AI の活用」

2024年6月

目次

1. はじめに.....	4
2. プロトタイピング対象候補選定および実施対象の決定.....	5
3. 各プロトタイピングの詳細.....	6
3.1. 疾患名の辞書コーディング結果を予測する AI.....	6
3.1.1. 目的.....	6
3.1.2. 背景.....	7
3.1.3. 方法.....	7
3.1.3.1. 学習/検証データの準備.....	8
3.1.3.2. テストデータの準備.....	10
3.1.3.3. データの前処理.....	10
3.1.3.4. 予測結果の出力/評価等.....	11
3.1.4. 結果.....	12
3.1.5. 考察.....	12
3.1.6. まとめ.....	13
3.2. 論理チェック仕様からクエリメッセージを作成する AI.....	13
3.2.1. 目的.....	13
3.2.2. 背景.....	13
3.2.3. 方法（ファインチューニングによるアプローチ）.....	14
3.2.3.1. 学習用およびテスト用データの作成.....	15
3.2.3.1.1. データの作成方針とルール決定.....	15
3.2.3.1.2. 実際の学習用およびテスト用データの作成.....	16
3.2.3.1.3. モデルの精度確認の結果に基づくデータの追加および修正.....	17
3.2.3.2. データに対する前処理.....	18
3.2.3.2.1. 表記統一の変換.....	18
3.2.3.2.2. 認識されない文字に対する対応.....	18
3.2.3.2.3. 大文字小文字の統一.....	19
3.2.3.2.4. 疑似コード中の変数名のラベル名への変換.....	19
3.2.3.2.5. データに対するトークン化の実施.....	20
3.2.3.3. 生成されたクエリメッセージの妥当性の評価.....	20
3.2.4. 結果（ファインチューニングによるアプローチ）.....	21
3.2.5. 考察（ファインチューニングによるアプローチ）.....	22
3.2.6. 方法（大規模言語モデルによるアプローチ）.....	22
3.2.7. 結果（大規模言語モデルによるアプローチ）.....	24
3.2.8. 考察（大規模言語モデルによるアプローチ）.....	28

3.2.9.	ファインチューニングと大規模言語モデルの両アプローチの比較考察.....	28
3.3.	添付文書一覧を活用した併用薬剤の使用目的の妥当性を予測する AI .....	29
3.3.1.	目的.....	29
3.3.2.	背景.....	29
3.3.3.	方法.....	29
3.3.3.1.	アプローチ1：機械学習モデルの構築.....	30
3.3.3.2.	アプローチ2：大規模言語モデルを用いるアプローチ.....	31
3.3.3.2.1.	判定したい薬剤の効能または効果の取得.....	31
3.3.3.2.2.	プロンプトへの埋め込み.....	32
3.3.4.	結果.....	33
3.3.5.	考察.....	33
3.3.6.	まとめ.....	34
4.	全体のまとめ.....	34
5.	おわりに.....	35
	参考文献.....	35

## 1. はじめに

Artificial Intelligence (AI) が我々の生活に衝撃的な変化を与えていることは多くの人々が既実感していることであろう。医薬品の研究開発においても、AI は前臨床開発から臨床試験データの記録と分析、ファーマコビジランス、臨床使用の最適化に至るまで、医薬品のライフサイクルのすべての段階で適用が進められている [1]。European Medicines Agency といった規制当局もその Reflection paper の中で「AI と機械学習ツールは、正しく使用されれば、医薬品ライフサイクル内でのデータの取得、変換、分析、および解釈を効果的にサポートすることができる」と AI の当該分野での有用性を認めている [2]。臨床開発における利用という観点では、膨大な時間とコストを要する治験プロセスを大幅に合理化できる手法として多くの製薬企業が AI に期待を寄せ、かつ実際に活用を推進している [3, 4, 5, 6]。

このような状況を背景に、日本製薬工業協会 医薬品評価委員会 データサイエンス部会の検討チームでは、臨床開発のプロセスの中で様々な業務に対して AI の利用が期待できるデータマネジメントの領域に着目し、2022 年よりデータマネジメントにおける AI 活用の可能性について検討しており、2022 年度にその成果として「データマネジメントにおける Artificial Intelligence の活用～ これから始める AI ～」 [7] を発行した。本成果物ではデータマネジメント領域で実際に利用されてきている AI のユースケースを紹介するとともに、利用上の留意事項や学習のリソース、また将来的な利用展開などの検討を行った。加えて AI を利用したプログラムによりテキストから有害事象等の安全性情報を検出するプロトタイピングも実施し、アンサンブルモデルでは F1 スコアについては 0.877、Recall については 0.987 という結果が得られ、機械学習を用いた AI モデルの有用性に対してある程度の感触を得ることができた。

データマネジメント領域では、現在でも「2.プロトタイピング対象候補選定および実施対象の決定」においてプロトタイピングの対象候補として取り上げたようなマニュアル作業に依存しているプロセスが少なからず存在し、担当者の作業負担となるばかりか、ヒューマンエラーによるデータ品質の影響がリスクとして懸念されている。そこで 2023 年度はデータマネジメント領域において行われている身近な業務の中で AI の活用により効率化やミスの削減が可能と思われるものを検討し、以下の AI のプロトタイピングを実施することにした。

- 疾患名の辞書コーディング結果を予測する AI
- 論理チェック仕様からクエリメッセージを作成する AI
- 添付文書一覧を活用した併用薬剤の使用目的の妥当性を予測する AI

これらのプロトタイピングを実施するにあたって、2022 年度の活動を通じて検討チームが様々な書物や資料から「頭で理解していた」ことは、「AI の活用には様々な考慮事項や課題がある」ということであった。

例えば、AI を活用するためにはそれなりの技術的な知識やスキルが必要であること、

AI の開発には偏りのない大量のデータが必要となること、またデータセットやプログラムを作成するあるいは既存のものを利用する場合には個人情報保護法、特許法や著作権法を含む知的財産法等についても留意する必要があること、更に IT リテラシー、プライバシー、セキュリティ、公平性、説明責任、透明性等に対する配慮が極めて重要であることなどが挙げられる。

本成果物で紹介する AI モデルの試作（以下、「プロトタイピング」という）において、タスクフォースメンバーは、使用する自然言語処理モデルおよび大規模言語モデルの選択、使用する学習データおよびテストデータ等の準備、意図した処理を実現するための AI モデルの訓練、検証、テスト等のプロセスをメンバー自身で一通り実施して、その結果をまとめることにした。その意図するところは、これまで AI モデルの構築や評価といったプロセスに携わったことがない読者が、成果物を通じてこれらのプロセスを「追体験」でき、それによってデータマネジメント業務における AI 活用がより身近に感じられるような読み物を作ることであった。読者の皆様には、今回のプロトタイピングで実現したそれぞれのモデルの精度よりもむしろ、タスクフォースメンバーが行ったプロトタイピングの過程を辿るように読んで頂ければ幸いである。

## 2. プロトタイピング対象候補選定および実施対象の決定

候補検討の最初の段階では実現可能性は考えずに、AI 活用により効率化や品質向上が期待できるものをブレインストーミング的に取り上げた。その結果以下の候補がタスクフォース内で取り上げられた。

- 論理チェック仕様とクエリメッセージとの比較またはプログラムへの翻訳
- 文書間の整合性の確認
- CRF記入手引きの自動生成
- 変数名および試験スケジュールからのUATテストデータ作成
- マニュアルクエリー発行の自動化
- プロトコルからCRFモジュール/項目の自動選択
- エンドポイントからプロトコルにおけるデータ収集項目の重要性判定
- プロトコルからデータマネジメント関連の各種仕様書（データベース、外部データの仕様等）の作成
- 外部データとCRFとの整合性確認
- 安全性データベースと臨床試験データベースとの整合性確認
- 製造販売後使用成績調査のうち投与症例全例を対象とした調査における重複症例の判断
- 同一治験における同一患者の二重登録
- 前相の試験結果や他試験の結果に基づく当該試験における解析対象集団への採否に関する判断の予測

- 辞書コーディング結果の予測
- 製造販売後使用成績調査における再調査不能の際のデータの補完
- プロトコルからデータベース定義書作成
- 添付文書一覧を活用した併用薬剤の使用目的の妥当性の予測

タスクフォースで利用できる時間的、技術的、予算的な制限を考慮した実現可能性の観点から検討を行い、また AI へのインプットおよび AI により生成されるアウトプットを整理した結果、最終的に以下の3つの AI をプロトタイピングの対象とすることとした。

- 疾患名の辞書コーディング結果を予測する AI
- 論理チェック仕様からクエリメッセージを作成する AI
- 添付文書一覧を活用した併用薬剤の使用目的の妥当性を予測する AI

### 3. 各プロトタイピングの詳細

本章ではタスクフォースで行ったプロトタイピングの詳細を紹介する。

「疾患名の辞書コーディング結果を予測する AI」では、コーディングを複数のカテゴリから1つを選択する分類タスクと捉え、テキスト分類に用いることができる **Bidirectional Encoder Representations from Transformers (BERT)**モデルを使用した。「論理チェック仕様からクエリメッセージを作成する AI」では、仕様からメッセージへの変換を言語翻訳と捉え、翻訳等に用いることができる **Text-to-Text Transfer Transformer (T5)**モデルを使用した。加えて、商用可能なライセンスで事前学習モデルが公開されたことから、文章を生成可能な大規模言語モデルの使用も試みた。「添付文書一覧を活用した併用薬剤の使用目的の妥当性を予測する AI」では、特定の疾患に対する薬剤の有効性を、添付文書を参照して大規模言語モデルに回答させる仕組みを構築した。

プログラムは Colabatory 上で作成し、GitHub に公開<sup>1</sup>した。Google アカウントを取得すれば誰でも無料で実行可能<sup>2</sup>であるため、実際の挙動や検討すべき事項の確認など、今後の AI 活用を検討する際の参考となれば幸いである<sup>3</sup>。なお、プログラムにはランダム性を有する部分が含まれている。可能な限りランダム性を排除した形で公開しているが、再実行において本成果物で記す結果の完全再現にはならない点に留意されたい。

#### 3.1. 疾患名の辞書コーディング結果を予測する AI

##### 3.1.1. 目的

病歴や有害事象等の CRF にフリーテキストで記載された疾患名に対して、適切な辞書

1 GitHub Repository: [GitHub - Takumi173/JPMA2023TF1-1](https://github.com/Takumi173/JPMA2023TF1-1)

2 一部のプログラムでは、プログラム間でデータ受け渡す目的で Google drive の接続を必要とします。必要に応じてプログラム実行用の Google アカウントを取得してください。

3 プログラムの作者および製薬協は、本プログラムの実行によって生じたいかなる損害に対して、一切の責任を負いません。

コードを予測する AI が構築可能か検討する。

### 3.1.2. 背景

データマネジメント領域ではマニュアル作業に依存しているプロセスの一つとして、コーディングがあげられる。ここで述べるコーディングとは、医師が CRF にて報告した用語を辞書で用意された標準的な用語に結び付けることを指している。コーディングは 1-2 名の担当者により実施されることが多く、多くのリソースやコストがかかり、また担当者による偏りなどの問題も生じており、AI の利用によりその課題を解決するというアイデアは多くのデータマネージャーが考えてきたことと思われる。

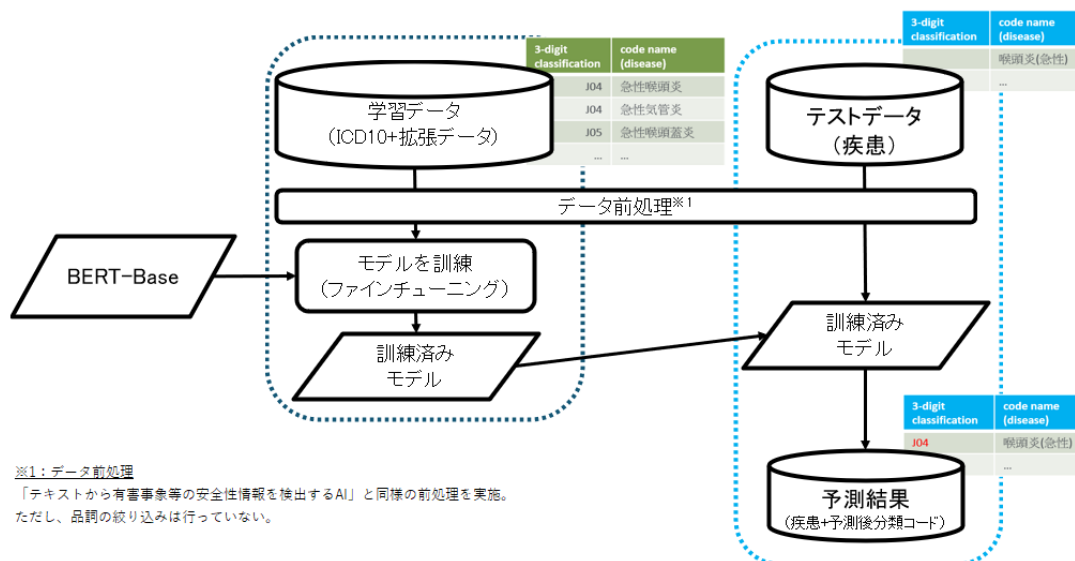
### 3.1.3. 方法

「疾患名の辞書コーディング結果を予測する」という目的においては、臨床試験等で一般的に利用されている MedDRA を使用することが理想的ではあった。しかし、本プロトタイプピングでは、ライセンスの都合上、MedDRA を使用することができなかつたため、ライセンスに制限のない「疾患標準コード：疾病及び関連保健問題の国際統計分類：International Statistical Classification of Diseases and Related Health Problems（以下、ICD）の「ICD-10（2013年版）準拠 基本分類表」を使用した [8]。

なお、今回の分類コードの予測には、文書分類に利用可能な BERT モデル<sup>4</sup>を使用した。AI を構築するためには「疾患名」とそれに対応する「コード」が学習データとして必要となるが、今回の AI 開発に適する公開データを見つけることができず、ゼロから作成することも現実的ではなかつた。また、少数のデータでは良い成績が得られなかつたため、ICD-10 の用語をベースとして同じ内容だが表現が異なるテキストを複数用意することで学習データの量を増やすデータ拡張を行い、学習データとした（データの拡張については「3.1.3.1. 学習/検証データの準備」参照）。プロトタイプピングではこの学習データを利用して BERT モデルをファインチューニングしモデルを構築、テストデータを用いて疾患名に紐づくコードを予測した（テストデータの詳細については「3.1.3.2. テストデータの準備」参照）。テストデータの疾患から適切な分類コードを予測した際のフローを図 3.1.3-1 に示す。なお、本プロトタイプピングは日本語のデータを想定したため、データサイエンス部会 2022 年度タスクフォース 1-1 の成果物である「データマネジメントにおける Artificial Intelligence の活用」にて紹介されている安全性情報を検出する AI と同じ前処理のアプローチをとっている。

---

4 Google が 2018 年に発表した、文書分類、文章の要約、質問応答、言語翻訳などの様々なタスクに応用可能な自然言語処理モデル。本プロトタイプピングでは東北大学自然言語処理研究グループによって作成された日本語の事前学習モデル ([cl-tohoku/bert-base-japanese-v3](https://github.com/cl-tohoku/bert-base-japanese-v3)) のファインチューニングを行った。



※1: データ前処理  
「テキストから有害事象等の安全性情報を検出するAI」と同様の前処理を実施。ただし、品詞の絞り込みは行っていない。

図 3.1.3-1

### 3.1.3.1. 学習/検証データの準備

ICD : International Statistical Classification of Diseases and Related Health Problems は、世界保健機関（WHO）において定められた分類であり、正式には「疾病及び関連保健問題の国際統計分類」である。異なる国や地域から、異なる時点で集計された死因や疾病のデータの記録、分析、比較を行うために国際的に統一した基準で設けられた分類であり、診療報酬明細書、電子カルテ、DPC（診断群分類・包括評価）等における死因・疾病分類として広く利用されている。

また、ICD は、アルファベットと数字によって表される以下（1）～（3）の分類表から構成されており、基本分類の項目数は約 14,000 項目となっている [9]。

- (1) 基本分類表
- (2) 疾病分類表（大分類、中分類、小分類）
- (3) 死因分類表

この分類は、3桁分類（アルファベット 1文字+数字 2文字）およびより詳細な分類である4桁分類（アルファベット 1文字+数字 3文字）からなるが、今回は基本分類の”3桁分類”コード（計 1529 件）（図 3.1.3.1-1 の緑枠内）を予測の対象とした。

種別 classification	分類単位 class unit	章番号 Chapter No.	中間分類 intermediate classification	3桁分類 3-digit classification	コード Code	コード名 Code name (disease)
03.三	細分類あり	01	A00 - A09	A00	A00	コレラ
05.四	最小	01	A00 - A09	A00	A00.0	コレラ菌によるコレラ
05.四	最小	01	A00 - A09	A00	A00.1	エルトールコレラ菌によるコレラ
05.四	最小	01	A00 - A09	A00	A00.9	コレラ, 詳細不明
03.三	細分類あり	01	A00 - A09	A01	A01	腸チフス及びパラチフス

図 3.1.3.1-1



ただし、3桁分類が U、V、Y、Z から始まる分類のコード名は、疾患に相当しない概念のものであるため予測の対象外とした。(図 3.1.3.1-2)。

種別	分類単位	章番号	中間分類	3桁分類 3-digit classification	コード	コード名 Code name (disease)
05.四	最小	22	U00-U49	U06	U06.0	エマージェンシーコードU06.0
05.四	最小	22	U00-U49	U06	U06.1	エマージェンシーコードU06.1
05.四	最小	22	U00-U49	U06	U06.2	エマージェンシーコードU06.2
05.四	最小	22	U00-U49	U06	U06.3	エマージェンシーコードU06.3

図 3.1.3.1-2

学習データには3桁分類の下位に紐づく4桁分類のコード名を使用するため、3桁分類のコードは学習データの対象外とした(図 3.1.3.1-3)

種別	分類単位	章番号	中間分類	3桁分類 3-digit classification	コード	コード名 Code name (disease)
03.三	細分類あり	01	A00-A09	A00	A00	コレラ
03.三	細分類あり	01	A00-A09	A01	A01	腸チフス及びパラチフス
03.三	細分類あり	01	A00-A09	A02	A02	その他のサルモネラ感染症
03.三	細分類あり	01	A00-A09	A03	A03	細菌性赤痢

図 3.1.3.1-3

上記フィルタリング実施後の”コード名”に対して、以下のようなデータ拡張を行い学習データとした。

ICD-10 のコード名に対して、以下表 3.1.3.1-1 の処理をプログラムで行い、オリジナルのデータ 1 万件をおよそ 14 倍の 14 万件まで拡張を行った<sup>5</sup>。

**転置**：MeCab（日本語のテキストを形態素（単語や文節）に分割するための形態素解析器）を使用し、テキスト分割後の単語を逆順に配置

**助詞等の追加**：'で'、'により'、'のため'、'から'、'あの'のような助詞や、'いつもの'、'の悪化'、'が悪化'、の表現を追加。

**単語変換**：MeCab を使用してテキストを分割し、単語ごとにランダムに原形や読み、発音に置換。

**ひらがな挿入**：コード名のランダムな位置に 1 文字のランダムなひらがなを挿入。

**再翻訳**：機械翻訳を利用し、日本語→英語→日本語に翻訳したデータをプログラムで読込、再翻訳データとして利用<sup>7</sup>。

5 拡張対象としたデータ：[Case1\\_ICD10\\_data.tsv](#)

6 プログラム：[JPMA2023\\_case1\(1\)\\_データの拡張.ipynb](#)

7 作成した翻訳ファイル：[Case1\\_Retranslated\\_data.tsv](#)

以下は、上述のデータの拡張の具体的な手法を、コード名「ウイルス性腸炎」を例として示している。

表 3.1.3.1-1

データ拡張方法	例示
転置	腸炎性ウイルス
助詞等の追加	ウイルス腸炎により
単語変換	ウイルス-virus 性チョウエン
ひらがな挿入	ウイルス性で腸炎
再翻訳	ウイルス性腸炎→Viral Enteritis→ウイルス性腸炎※

※「ウイルス性腸炎」では再翻訳で元のテキストと同一になったが、「急性副鼻腔炎、詳細不明」の場合、「急性副鼻腔炎、詳細不明」→「Acute sinusitis, details unknown」→「急性副鼻腔炎、判定不能」のように異なる表現を作り出すことができた。

この手法により、ICD-10 の 3 桁分類の下位に紐づくコード名を拡張し、モデル構築用の学習データとし、モデルの訓練を実施した。

### 3.1.3.2. テストデータの準備

一方、上述のモデルの評価に用いるテストデータは、実際のデータに即していることが望ましいことから、図 3.1.3.2-1 の赤枠のコード（計 1529 件）に対して、それぞれ 1 件ずつ、合計 1529 件のテストデータ（図 3.1.3.2-1 の黄色枠のコード名）をマニュアルで作成した<sup>8</sup>。作成時のルールを以下の通り設定し、可能な範囲で適応した。

- ICD-10特有の記述方式< >の削除
- 語順の入れ替え
- 別事象名への変換 (筋ストレイン → 肉離れ、浮腫→むくみ、等)

3桁分類 3-digit classification	Code	Code name (disease)	Test data
F00	F00.0	アルツハイマー<Alzheimer>病の認知症, 早発性 (G30.0†)	アルツハイマー型早発性認知症
I85	I85.0	出血を伴う食道静脈瘤	食道静脈瘤 (出血を伴う)
M62	M62.6	筋ストレイン	肉離れ
R60	R60.0	限局性浮腫	限局性のむくみ

図 3.1.3.2-1

### 3.1.3.3. データの前処理

学習データとテストデータの各文章の形態素解析を行う前に、文字種や表記揺れを統一するために以下のような処理を行った<sup>9</sup>。

1. jaconv ライブラリを使用した半角・全角等の文字の正規化
2. unicodedata ライブラリを使用したユニコード文字の正規化
3. neologdn ライブラリを使用した反復表記などの不要な表現の除去や、ハイフンや

<sup>8</sup> テストデータ: [Case1 Test data.tsv](#)

<sup>9</sup> データマネジメントにおける Artificial Intelligence の活用 ～ これから始める AI ～ [7]Appendix 2 「A2.1.2.1. 表記の統一」参照

ダッシュ、チルダ等の記号文字の正規化

#### 4. テキストの末尾にある句読点を空文字列で置換

形態素解析器は MeCab を使用し、システム辞書として mecab-ipadic-NEologd (version, v0.0.7, release date 2020-08-20) [10] を使用、ユーザー辞書として万病辞書 [11]を使用した。形態素解析後、形態素に原形が定義されている場合は原形に置き換えを行い、表現を統一した<sup>10</sup>。形態素解析を実施しないと、モデルがデータのどこまでを単語とすべきか理解することができないためである<sup>11</sup>。

### 3.1.3.4. 予測結果の出力/評価等

作成した学習データを用いて BERT モデルをファインチューニングしたモデル<sup>12</sup>を用い、前項で準備した 1529 件のテストデータにおけるコード（疾患）から分類コードの予測を行った<sup>13</sup>。

予測結果をテストデータ毎に出力し、各行には、「テストデータ」、「3 桁分類コード」、「予測した 3 桁分類コードの ID」、「予測した 3 桁分類コードの確信度」、「予測した 3 桁分類コード」の情報を設定した。なお、疾患名のコードには類似のコードが多数存在することがあるため、本プロトタイプングでは最終的に人がレビューをするフローを想定し、単一の予測結果ではなく、モデルの確信度が高い物から上位 5 つを出力する設計とした。

target	label	Pred_1	Pred_2	Pred_3	Pred_4	Pred_5	Conf_1	Conf_2	Conf_3	Conf_4	Conf_5
コレラ コレラ 菌 による	A00	0	18	27	106	1	30.93244743	20.35388565	19.96270752	19.40986252	19.36992645
バラチフス a	A01	1	232	229	38	859	22.63074493	12.72511959	12.48373413	12.22808266	12.19203663
腸炎 サルモネラ 感染	A02	2	4	8	5	23	28.48793983	25.85512543	24.77991295	18.37459755	18.3287487
細菌 志賀 菌 赤痢	A03	3	593	4	36	5	33.59459686	23.91164207	22.83763504	21.86834717	21.8435688
カンピロバクター 腸炎 による 腹痛 下痢 嘔吐	A04	668	664	597	8	1110	16.87114334	16.80198479	16.26134109	15.96619034	15.62317562
葡萄 球菌 食中毒	A05	5	1315	4	593	17	30.09703827	19.37737274	17.92895508	17.18352127	16.66030312
急性の アメーバ 赤痢	A06	6	3	1	8	0	29.70781326	22.17621613	15.97030067	15.41619301	15.29159737

Pred_1_Label	Pred_2_Label	Pred_3_Label	Pred_4_Label	Pred_5_Label
A00	A23	A36	B53	A01
A01	D30	D26	A51	N34
A02	A04	A08	A05	A28
A03	J15	A04	A49	A05
K58	K52	J20	A08	R10
A05	T61	A04	J15	A22
A06	A03	A01	A08	A00

(上表から続く)

図 3.1.3.4-1

上の図 3.1.4.1-1 の説明を表 3.1.4.1-1 に示した。

表 3.1.4.1-1

10 プログラム：[JPMA2023\\_case1 \(2\) データの前処理.ipynb](#)

11 データマネジメントにおける Artificial Intelligence の活用 ～ これから始める AI ～ [7]Appendix 2 「A2.1.2.2. 形態素解析」参照

12 プログラム：[JPMA2023\\_case1 \(3\) モデルトレーニング.ipynb](#)

13 プログラム：[JPMA2023\\_case1 \(4\) 訓練済みモデルによる予測.ipynb](#)

No	列	概要
1	target	テストデータ
2	label	3桁分類コード
3	Pred_1~5	予測した3桁分類コードのID (※IDではどのラベルか不明なためNo5のPred_1_Label~5で予測した3桁分類コードを表示)
4	Conf_1~5	予測したPred_1~5の3桁分類コードの確信度 (%)
5	Pred_1_Label~5	予測した3桁分類コード

### 3.1.4. 結果

1529件のテストデータから予測した分類コードと予め作成しておいた正解コードとの照合により正誤を判定した結果、その正答率は表 3.1.5-1 の通りであった。

表 3.1.5-1

正答率		
Top 1	Top 3	Top 5
82.4%	90.7%	92.4%

なお、Top 1 とは、モデルが予測した分類コードの予測結果の中で最も高い確率で示された分類コードであり、正解コードとの正答率は 82.4%であった。

同様に Top 3 は予測結果の中で高い確率が示された上位 3 つの分類コード、Top 5 は上位 5 つの分類コードを示し、上位 3 つの分類コードに正解コードが含まれる割合は 90.7%、上位 5 つの分類コードに正解コードが含まれる割合は 92.4%といずれも高い正答率が示された。

以上のことから、学習用データが標準コードデータしかない場合でも、辞書分類コードを目的とした分類において、機械学習モデルを利用した分類が一定の効果を発揮することが確認できた。

### 3.1.5. 考察

モデル構築の初期は思うようにモデルの正答率が上がらず、[学習/検証データの準備]の項でデータの拡張を行ったデータを用いたモデルであっても正答率は 30%程度に留まっていた。上述の[データの前処理]を実施することで、正答率の向上に繋がった。BERT モデルのファインチューニングを行う場合はこういった前処理の重要性を改めて認識することができた。

また、学習データを準備する際、可能な限り多様性のあるデータを用意することでモデルの正答率向上に繋がることの実感できたこともこのプロトタイプングでの大きな学びであった。今回、転置、助詞等の追加、単語変換、ひらがな挿入、再翻訳等のデータ拡張手法の有用性を試行錯誤で進め、相当の時間を要したものの、ファインチューニングモデルにおける学習データ量の価値というものを改めて強く認識できた。本プロトタイプングでは検討できなかったデータ拡張手法を組み合わせることで、さらにデータの多様性が増し、モデルの正答率向上に繋がる可能性があるものと思われた。

### 3.1.6. まとめ

本プロトタイプングでは我々が日常の業務で使用している MedDRA はライセンスの都合上用いることができなかつたため、ICD-10 を使用した。しかしながら本プロトタイプングで用いた予測モデルの構築手法は、予測する辞書や内容が変わっても適用可能と考えられるため、参考になれば幸いである。

本プロトタイプングモデルは、ICD-10 の 3 桁コードを予測するモデルを構築したが、MedDRA の LLT コードを予測するモデルを構築したい場合、学習データを「各社でこれまでに蓄積された報告語のコーディングデータ」と読み替えていただく必要があり、そこに対してデータの拡張手法を含めて本プロトタイプングの構築手法を適用することで LLT の予測モデルを構築することも可能であると考えられた。

また、通常、モデル構築には膨大な量の学習データが必要になるが、今回はデータ拡張の手法を用いることで、ユーザー側ではデータを作成せずにプログラムで学習データの準備を行った。この手法で一定の成果を上げることができたため有効な手段であったと考える。

以上より、現在コーディングは【背景】に記載の通り人の手を介する作業となっているが、リスクと精度の観点から必要と思われるレベルの人の介在を前提に、現在の人による作業の一部または大部分を AI に置き換えることで一定の成果が上げられると考えられる。

## 3.2. 論理チェック仕様からクエリメッセージを作成するAI

### 3.2.1. 目的

EDC のデータが論理的に正しいかどうかをチェックする仕様（プログラミング言語に近い記載方法・疑似コードで定義されることを想定）を元に、クエリメッセージ文案を自動で作成することが可能であることかを検討した。このプロトタイプングでは ファインチューニングによるアプローチおよび大規模言語モデルを用いたアプローチの 2 通りの方法を用いて検討を行った。

### 3.2.2. 背景

論理チェックには、変数、ラベル、一部の条件、条件値、上限値、下限値等を変更するだけで他の論理チェックにも利用できるような同一パターンあるいは類似のパターンの論理チェックが数多く存在する。しかし、現在は多くの企業で論理チェック毎にマニュアル作業でクエリメッセージを準備している現状と思われた。そこで各社で蓄積されている論理チェック仕様とクエリメッセージを学習データとしてそれにテキストからテキストへの変換を可能とする機械学習モデルを適用することで、論理チェック仕様からクエリメッセージを作成する可能性の検討を行うことにした（図 3.2.2-1 参照）。

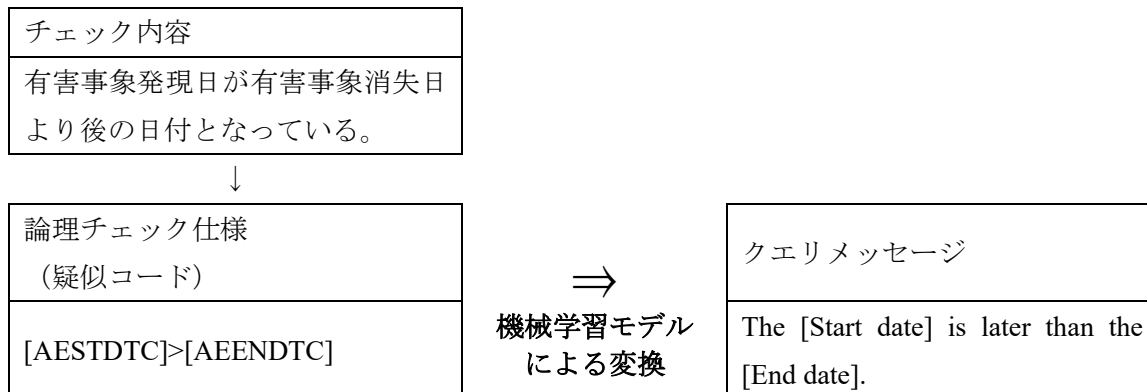


図 3.2.2-1

### 3.2.3. 方法（ファインチューニングによるアプローチ）

テキストからテキストへの翻訳等に適用可能な「T5-Base」モデルをファインチューニングし、モデルを構築した（図 3.2.3-1）。学習およびテスト用に用意した 635 件のうち 80% の 508 件を用いてモデルを訓練し、残り 20% でクエリを生成させ人による受け入れ評価を行った<sup>14</sup>。

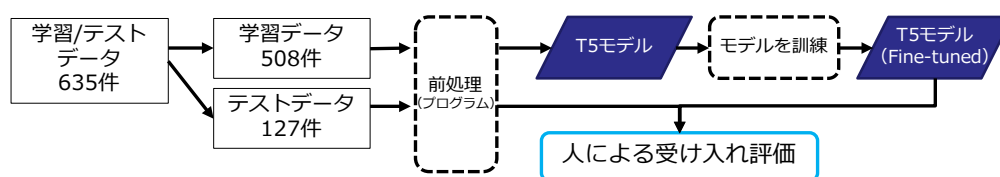


図 3.2.3-1

#### < 「T5」モデルに関して >

「T5」モデルとは、Text-to-Text Transfer Transformer モデルの略称であり、質問応答、翻訳、要約など、テキストからテキストへの変換を行うことができる、Google が発表した自然言語処理の有名なモデルである。

- 「T5」モデルは、パラメーター数が異なる複数のバリエーションを持つ（「T5-Small」、「T5-Base」、「T5-Large」、「T5-3b」、「T5-11b」等）。今回は、リソース（GPU性

<sup>14</sup> プログラム：[JPMA2023\\_case2 \(1\) T5 を用いたメッセージの生成.ipynb](#)

能)の制約により、「T5-Base」を使用した。

- 「T5」モデルは事前に学習されているモデルであり、今回はそのモデルに対して新たな学習データで訓練することで、ファインチューニングをしている。

### 3.2.3.1. 学習用およびテスト用データの作成

#### 3.2.3.1.1. データの作成方針とルールの決定

実業務の中で本項のようなAIを作成しようとした場合、各社で保持している標準的な論理チェック仕様書をベースに、学習用データを作成するケースが多いと想像する。一方で、上述の仕様書は各社の知的財産に相当するものであり、本タスクチームの活動として共有が難しかったため、今回のプロトタイプングでは参照はしたものの、そこから学習用データとして一から人の手で作り上げていく必要があった。データの形式としては、①疑似コードとクエリメッセージの組み合わせ、および②変数名の読み替え表の2つを用意することとした。(今回のプロトタイプングにおいて必要となるデータは、モデルの作成およびテストに必要な疑似コードとクエリメッセージのペアである。しかしながら、疑似コードは変数名で記載することに対して、クエリメッセージはラベル名で記載することが多いため、変数名をラベル名に読み替えるための表を合わせて作成した。)

##### 1. 疑似コードとクエリメッセージの組み合わせ

疑似コード	クエリメッセージ
[AESTDTC]>[AEENDTC]	The [Start date] is later than the [End date].

##### 2. 変数名の読み替え表

変数名	ラベル名
AESTDTC	Start date

データの作成にあたっては、AIがデータを読み取るにあたりできるだけ均一な表現や形式で記載されることが望ましかったため、作成するデータについて、最低限必要な記載ルールを設定した。ルールの事例は以下の通り。

- 疑似コード、クエリメッセージ共に、変数名およびラベルの前後には括弧 (“[”“および”]”) を、変数の出力値の前後にはクォーテーションマーク (“”等) を付ける。
- 異なるVisitやフォルダの同一名称の変数を使用する場合、疑似コードの変数名の前に【】を用いてVisitやフォルダを記載し、変数名の読み替え表においてもその情報を含めて記載する。

変数名	ラベル名
【Screening】 [TULOC]	Target Lesions

- 疑似コードにおける「X = ""」、「X is null」、「X <> ""」、「X is not null」といった表記のゆらぎは許容する。
- 疑似コードにて、複数のand/or条件を持つデータパターンの場合（例：A or B and C）、優先させたい条件は括弧マークを用いて表現し、展開は不要とする（例：(A or B) and C）。
- 疑似コードにて、レコード数を判断条件に含むデータパターンの場合、”record count more than 1”など、状況が具体的に分かるように表現する。

### 3.2.3.1.2. 実際の学習用およびテスト用データの作成

AI の作成にあたりどの程度のデータ数が最終的に必要か未知であり、また上述の通り、学習データを一から作成したため一度に大量のデータを作成するのは難しいと考えた。そのため、データ作成の流れとして、まずは最低限の数のデータ（100 例を目安とした）を用意した。次に、後述するモデルの構築によって生成されたクエリメッセージを見て、特に受け入れ可能と判断された割合が低かった学習データの条件を特定し、その精度の向上に焦点をあてて学習データの修正と追加を行った。この作業を何度か繰り返し、最終的に 635 例の学習データを用意した。具体的な作成方法について、以降に記述する。

まず、多様なパターンを含めることに重点を置いてデータの作成を進めた。しかし作成を進める中で、データのパターン自体は限られており、この方針のままデータ量を増やすことは困難であると分かったため、同じパターンではあるが変数名が異なるデータも追加することとした。

	疑似コード	クエリメッセージ
初期のデータ→	[AESTDTC]>[AEENDTC]	The [Start Date] is later than the [End Date].
新たな追加データ→	[VSDT]> [RFSTDT]	[Assessment Date] is after the [Date of First Dose of Study Drug].

その結果、データの合計は約 100 例となった。なお、やや複雑と思われたパターンのデータについても、出来る限り汎用性の高い AI を作成したい思いがあったことから、まずは含めることとした。

#### 複雑なデータパターンの事例

疑似コード	クエリメッセージ
[OVRLRESP]<> "COMPLETE RESPONSE" AND [OVRLRESP]<> null AND	[Overall Response] is not "COMPLETE RESPONSE", yet based on the current data the expected [Overall Response] is "COMPLETE RESPONSE". Please review



[TRGRES] = "NOT APPLICABLE" AND [NTRGRES] = "COMPLETE RESPONSE" AND [NEWLPROG] <> "UNEQUIVOCAL" AND [NEWLPROG] <> null	per RECIST 1.1 and clarify.
------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------

### 3.2.3.1.3. モデルの精度確認の結果に基づくデータの追加および修正

データの作成段階で集まった学習データとは別に、20例ほどのテストデータを用意した。学習データに基づいてファインチューニングしたモデルでこの20例のテストデータのメッセージを作成し、モデルの精度がどの程度であるかを確認しながらデータ作成を進めた。ハイパーパラメータの調整を行った上で、特に精度が低かった2種類のパターンのデータにつき、重点的にデータを増やすこととした。

データパターン	疑似コード	クエリメッセージ
不等号を条件式に持つ	[A] < 1	[A] is smaller than 1.
	[A] - [B] =< 1	[A] minus [B] is equal to or less than 1.
値の範囲を示す	[VSTEST] = "Pulse Rate" AND ([VSORRES] < 50 or [VSORRES] > 110)	The "Result" for the [VSTEST] is outside the expected range. Please update or confirm as appropriate.
	[VSTEST] = "Temperature" AND ([VSORRES] < 35.5 or [VSORRES] > 41.0)	The "Result" for the [VSTEST] is outside the expected range. Please update or confirm as appropriate.

データを追加し更に検証を行ったところ、重点的に増やしたパターンにおける精度は上がった一方で、それ以外のデータについては精度が逆に下がる結果となった。原因としては、全体の学習データ量の少なさにより、学習データ内の偏り（例：不等号を条件式に持つデータパターンにおいて、全て数値を”1”としたデータを追加）の影響が大きかったことが考えられた。そのため、重点的に増やしたパターンのデータ内の偏りを可能な限り小さくする（例：同一のデータパターンにおいても、使用する数値に偏りが出ないようにする）と共に、同一パターンだが数値やラベル名、クエリの対象となる検査項目が異なるデータの追加、ならびに疾患特有の内容を含むデータを追加することで、データ量自体を全体的に増加させることとした。

### 数値やラベル名の変更事例

	疑似コード	クエリメッセージ
初期のデータ →	[EGDT] < [EXSTDT] - 30 or [EGDT] > [EXSTDT]	[Time ECG Performed] is not within 30 minutes prior to the [Dosing Time].
追加データ (数値の変更) →	[EGDT] < [EXSTDT] - <b>10</b> or [EGDT] > [EXSTDT]	[Time ECG Performed] is not within <b>10</b> minutes prior to the [Dosing Time].
追加データ (ラベル名の変更) →	[EGDT_1] < [EXSTDT]-30 or [EGDT] > [EXSTDT]	[ <b>ECG Performed Time</b> ] is not within 30 minutes prior to the [Dosing Time].

なお、データを追加する際のルールとして以下を設け、これらを踏まえてデータの追加を行った。結果、合計 635 例のデータとなった。

- 実データとして発生しうる数値や条件のデータにとどめる。
- 変数名が異なってもラベル名が同一の場合、テストデータとしては同一データとして扱われるため、必ずラベル名も変化させる。

### 3.2.3.2. データに対する前処理

#### 3.2.3.2.1. 表記統一の変換

シングルクォーテーションはダブルクォーテーションに変換した。また、「<」、「>」、「=」、「+」、「<=」、「>=」、「=<」、「=>」、「<>」の前後にスペースが必ず入るように変換した。適切にスペースを入れることで、上記の用語の前後の単語を正しく認識させるために変換した。

#### 3.2.3.2.2. 認識されない文字に対する対応

モデルに取り込ませる際に認識されない文字があったため、前もって違う用語に変換した。(今回は全て ASCII 文字に置き換えて処理を行った。)

- 「 < 」 → 「 LT 」
- 「 <> 」 → 「 NE 」
- 「 <= 」 → 「 LTE 」
- 「 =< 」 → 「 LTE 」
- 「 ^ 」 → 「 NT 」
- 「 【 」 → 「 || 」
- 「 】 」 → 「 >> 」

### 3.2.3.2.3. 大文字小文字の統一

大文字と小文字の違いでも、違う文字としてみなされるため、同じ文字としてみなされるよう括弧の中身以外はすべて小文字に変換した。

例：

「[Start Date of Radiation Therapy] Is less than 8 weeks prior to [Date of First Study Drug Taken], which is exclusionary. Please review.」

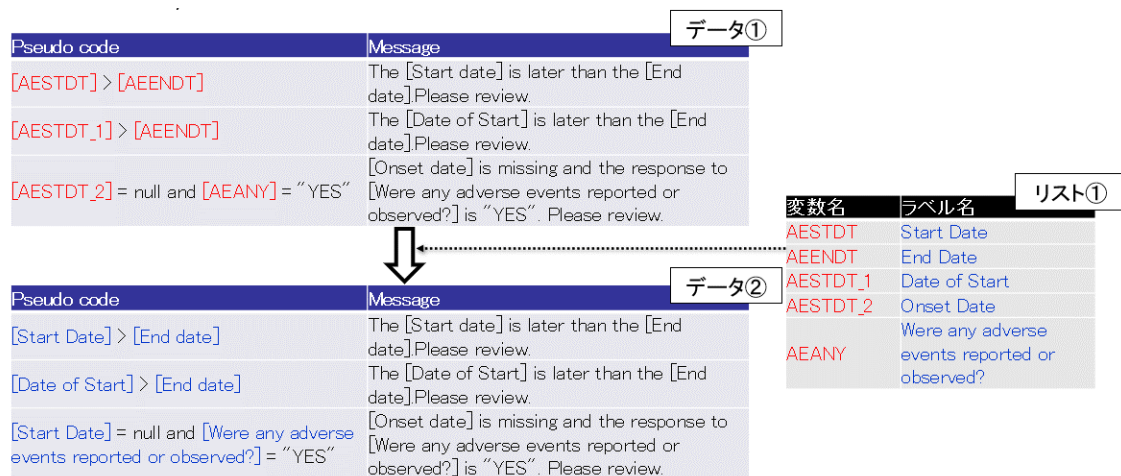
→

「[Start Date of Radiation Therapy] is less than 8 weeks prior to [Date of First Study Drug Taken], which is exclusionary. please review.」

### 3.2.3.2.4. 疑似コード中の変数名のラベル名への変換

モデルによる疑似コードからクエリメッセージへの変換を支援する目的で、「変数名の読み替え表」を用いて疑似コード中の変数名をクエリメッセージで使用されているラベル名に変換した（図 3.2.3.2.4.-1）。

図 3.2.3.2.4.-1



（※データ①およびリスト①は手作成で、データ②はそれらを使用して VBA プログラムで作成した）

この際、データ①とリスト①の整合性に関して細かい確認と修正に労力を費やした。

注意が必要であった例 1

リストにおいて[AESTDT ]というように文字の前後にスペースが入っていると、[AESTDT]は置換されないため、余分なスペースを削除してきれいにする必要があった。

注意が必要であった例 2

リストが不十分な場合、データ①に変数名が残ってしまう。目視で変数の変換漏れがないか確認し、リスト①を補完するなどの対応が必要であった。

### 3.2.3.2.5. データに対するトークン化の実施

テキストデータを機械学習モデルや自然言語処理モデルで処理する際は、テキストデータをモデルが理解できる入力形式に変換すること、すなわちトークン化が必要である。すべての自然言語処理モデルに対して一様に適用可能なトークン化の方法は存在せず、それぞれのモデルによってトークン化する際のルールや方法が異なる。

「T5」モデルの事前学習時においても、特定のルールに基づいてテキストデータをトークン化し、学習している。今回、ファインチューニングするにあたり、事前学習時と同じルールでのトークン化を実施した。

### 3.2.3.3. 生成されたクエリメッセージの妥当性の評価

想定していたメッセージと、モデルから出力されたメッセージが一致しているかを機械的に判断し、完全に一致したものは全て“受け入れ可能”とした。

一致しなかったメッセージに関しては、3人の担当者が目視で確認し、“受け入れ可能”または“受け入れ不可”を判断した。判断時には以下の点を基準に判断した。

- 文章として成立していること
- 疑似コードの内容とメッセージの意味が一致していること
- このままEDCでクエリを発行しても、医療機関側で問題なく対応できる文章になっていること

3人がそれぞれ個別に評価を行い、3人とも同じ評価をしている場合にはそれを最終判定とした。3人の意見が異なる場合は、3人で協議を行った上で最終判定とした（図 3.2.3.3-1）。

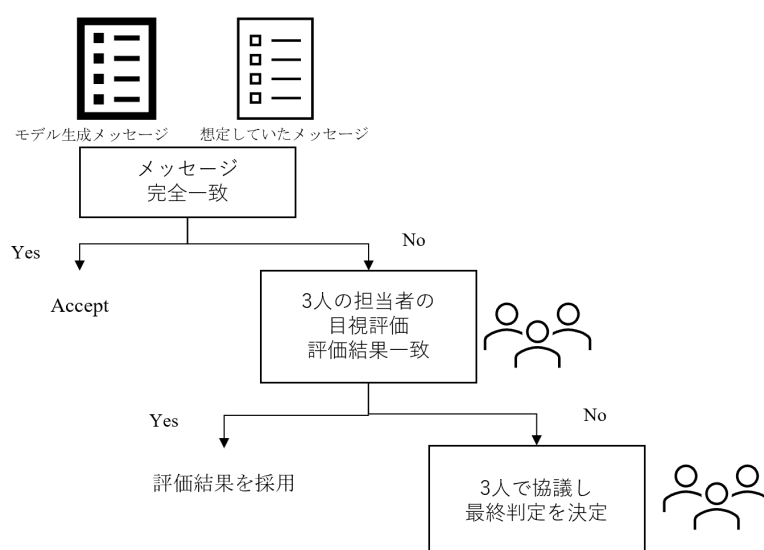


図 3.2.3.3-1

### 3.2.4. 結果（ファインチューニングによるアプローチ）

モデルの精度を知る上で、テストデータを用いてモデルの評価をすることは、とても有用である。今回学習およびテスト用として作成した 635 件のデータの内学習用に利用した 80%以外の 20%（127 件）のデータをテストデータとして用いてモデルの評価を行った。

テストデータの 127 件のうち、評価時に疑似コード自体が誤りであることが判明した 2 件は判定不能とした。上記 2 件を除いた 125 件のうち、112 件（89.6%）が受け入れ可能と判断された。この結果は、このアプローチで実用可能なモデルを構築できる可能性があることを示唆していると思われた。

以下に受け入れ不可となったメッセージのうち、いくつか主な事例をピックアップして紹介する（表 3.2.4-1）。受け入れ不可となった主な要因は、よく似た学習データが存在しており、そのデータに左右されたと考えられる。例えば、最初の事例では学習データに

疑似コード：[Pathogen] = "OTHER" AND [Other Pathogen] is NULL

期待したメッセージ：Response to [Pathogen] is "OTHER", but [Other Pathogen] is missing. Please review and update as appropriate.

というデータが存在するため、本学習データの影響が濃く出た結果と考える。

表 3.2.4-1

疑似コード	期待したメッセージ	T5（ファインチューニング）の結果	受け入れ不可理由
[Pathogen] <> "OTHER" and [Other Pathogen] is not null	response to [Pathogen] is not "OTHER", but [Other Pathogen] is provided. please review and update as appropriate.	response to [Pathogen] is <u>"OTHER"</u> , but [Other Pathogen] is provided. please review and update as appropriate.	『is "OTHER"』ではなく、『is not "OTHER"』とすべき
[Start Date of Radiation Therapy] >= [Date of First Study Drug Taken]	[Start Date of Radiation Therapy] is equal to or after the [Date of First Study Drug Taken] on the randomization form. please review.	[Start Date of Radiation Therapy] is <u>after</u> the [Date of First Study Drug Taken] on the study drug dosage form. please review.	イコールの要素が欠落している
[Was Lesion Assessed?] = "YES" and [Diameter] and [Dimension Not Measurable] are null for the same row	[Diameter] is not recorded, but a response is also missing for the [Dimension Not Measurable] field.	[Diameter] is <u>not in the list</u> , but a response is also missing for the [Dimension Not Measurable] field. please update or clarify.	『List』の記載が、クエリを受け取った側で正確に意味を理解できない

	please update or clarify.		
record count of [Lesion Number] is more than 1	duplicate [Lesion Number]. please check and update.	the [Lesion Number] is <u>more than 1</u> .	Lesion Number 自体は 1 以上になり得るか つ、duplicate の記載がないため、意味が伝わらない

### 3.2.5. 考察（ファインチューニングによるアプローチ）

「方法」で示した前処理、トークン化、モデルの訓練においてはいずれも Python を使用しており、外部ライブラリ（「transformers」や「datasets」など）を使用することで、それぞれのタスクにおいて数行の簡易的なコードで実施が可能であった。プログラム担当者が普段使用していない外部ライブラリであったため、最初は理解と実装に労力を費やすことになった。このような労力がかかることも今回のプロトタイピングを通じて学ぶことができた。今回はすべて ASCII 文字に置き換えて処理を行ったが、今回用いた T5-base モデルでは「<」や「^」といった文字がトークン化適応範囲外となっており処理できなかった点は驚きであった<sup>15</sup>。

今回のモデルを訓練に際しては、設定したハイパーパラメータに関して根拠はなく、他の文献で設定された数や、実際に訓練を繰り返して感覚的に好ましい数を選択するにとどまった。実際は可能な限り根拠に基づき、ハイパーパラメータのチューニングを実施するのが望ましいと思われた。

今回の結果は比較的高い精度となったが、より改善するためには、さらに学習データを増やす必要があると考える。ただし単に受け入れ不可を補うような学習データばかり増やすのではなく、バランスよく学習データを増やすことが重要である。うまく変換できないパターンの変換精度を向上させようとしてそのパターンの学習データの割合を増やすとそのパターンの影響を受けて他のパターンで間違ったクエリを生成することが観察された。

実際の試験で実施する場合は、各企業内の複数の過去試験の疑似コードとクエリメッセージの組み合わせを学習データに利用できるため、学習データは今回用意した数よりも多くなると予想される。また、過去試験の疑似コードとクエリメッセージの組み合わせを利用する際には、試験共通のルールで疑似コードとクエリメッセージが作成されていることが追加の処理なく効率的に学習データとして活用する上で望ましいが、そうでない場合にはある程度の変換処理を行って学習データを準備することになるだろう。

### 3.2.6. 方法（大規模言語モデルによるアプローチ）

<sup>15</sup> 一般に機械学習において自然言語を処理する場合、テキストのままではなく一意の ID 等に置き換えて処理することが多い。事前学習済みのモデルにおいては、各単語や文字を独自のボキャブラリーとして保持しており、そのデータを参照して読み込んだテキストを置き換えていく。しかし、ボキャブラリーに保持していない単語や文字は処理することができないため未知語として処理される。

セクション 3.2.6 から 3.2.8 までは大規模言語モデルによるアプローチに関する方法、結果、考察を記載する。

本アプローチで用いたデータフローを図 3.2.6.1 に示す。

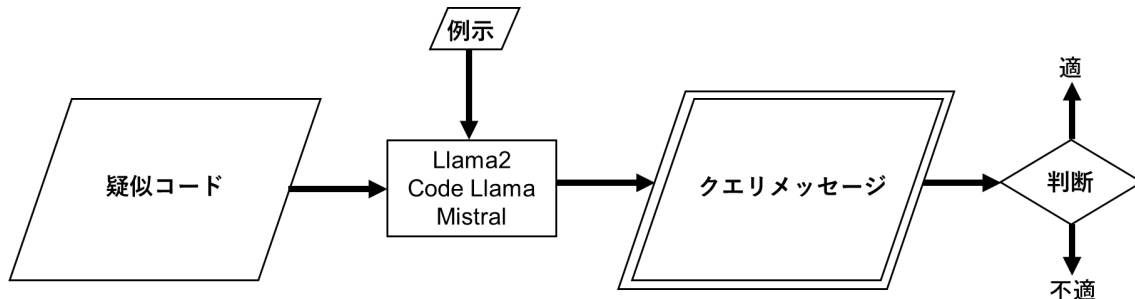


図 3.2.6.1

大規模言語モデル (Large Language Model、LLM) にクエリメッセージを作成する指示を与えるために、まずプロンプトで以下のような疑似コードとクエリメッセージの組み合わせの例示を与えた<sup>16</sup>。

```
### CONTEXT
```

```
Here are examples of translations from CODE to TEXT.
```

LLM アプローチの初回試行時に用いた LLM は Code Llama および Llama2<sup>17</sup>であった。以下のような疑似コードとクエリメッセージの組み合わせ事例を 7 つのみ用意してクエリ生成を試みた。

Example 1:

```
CODE: [Overall Response] <> "PROGRESSIVE DISEASE" AND [Overall Response] IS NOT EMPTY AND [New Lesion Progression] = "UNEQUIVOCAL"
```

```
TEXT: The [Overall Response] is not "PROGRESSIVE DISEASE", yet the [New Lesion Progression] is "UNEQUIVOCAL". Please review.
```

Example 2:

```
CODE: [Stop Date] > [Date of First Study Drug Taken] -30
```

```
TEXT: The [Stop Date] is later than 30 days prior to the [Date of First Study Drug Taken]. Please review.
```

Example 3:

16 プログラム : [JPMA2023\\_case2 \(2\) LLM を用いたメッセージの生成.ipynb](#)

17 Code Llama および Llama2 はいずれも Meta 社が 2023 年に公開した大規模言語モデル。Code Llama は Llama2 をベースにプログラムの生成について追加訓練が実施されている。本プロトタイプでは Code Llama は [codellama/CodeLlama-13b-Instruct-hf](#)、Llama2 は [meta-llama/Llama-2-13b-chat-hf](#) を使用してメッセージの生成を行った。

CODE: [Primary Study Drug Treatment Status] = "COMPLETED" and [Date of Last Dose of Study Drug] < [Study Day 90]

TEXT: The [Primary Study Drug Treatment Status] is "COMPLETED", yet the [Date of Last Dose of Study Drug] is prior to the [Study Day 90]. Please review.

その上で以下のプロンプトにより、それぞれの疑似コードからクエリメッセージを作成する指示を与えた。

### QUESTION

Step1: Referring to the CONTEXT examples, translate the following CODE into TEXT exactly. It is important to maintain the CODE and the TEXT square brackets or quotation marks around the words when translating. Answer should be start with "TEXT:" and the TEXT should be end with "Please review." No explanation required.

この段階で T5-Base のファインチューニングによるアプローチと同様に 3 人の担当者による「医療機関へこのままクエリとして発行することを許容できるか」という判断基準で適否を判定した。

初回の 7 例のみの例示による試行の結果、疑似コードに含まれている「=<」、「=>」、「<>」、「^=」のような記号を LLM がよく解釈できていない可能性が示唆された。これらの記号は疑似コードでは使われることが少なくないと思われたが、大規模言語モデルが作成された際に利用された学習データにおいては一般的な表現ではなかった可能性が考えられ、テスト用の疑似コード中のこれらの記号を以下のように変換した上で LLM にクエリを生成させることにした。

- =< → <=
- => → >=
- <> → !=
- ^= → !=

また、LLM に例示を与えるプロンプトに「<=」、「>=」、「!=」、「<」、「>」を含む事例を 3 例追加した。さらに等号、不等号などの記号の前後にスペースを入れる処理を行った上で LLM に再度クエリメッセージを生成させた。最終的に Code Llama および Llama2 に加えて Mistral<sup>18</sup>も用いて同様のクエリ生成を試みた。この内、T5-Base のファインチューニングによるアプローチの評価時に使用した 125 件のデータを用いて、セクション 3.2.3.3 に記載された方法と同じ方法で人による受け入れ評価を行った。

### 3.2.7. 結果（大規模言語モデルによるアプローチ）

<sup>18</sup> Mistral AI 社が 2023 年に公開した大規模言語モデル。本プロトタイプでは [mistralai/Mistral-7B-Instruct-v0.2](https://mistralai.com/docs/models/mistral-7b-instruct-v0.2) を使用してメッセージの生成を行った。



1回目の試行において、Code Llama を用いた結果では 125 件中 99 件 (79.2%) のクエリメッセージを受け入れ可能と判定できた。また Llama2 を用いた結果では 125 件中 86 件 (68.8%) のクエリメッセージを受け入れ可能と判定できた。

続いて行った前処理および例示追加後の試行では、Mistral を用いた結果で 125 件中 106 件 (84.8%)、Code Llama では 125 件中 111 件 (88.8%)、Llama2 では 125 件中 113 件 (90.4%) のクエリメッセージを受け入れ可能と判定できた。前処理および例示追加をすることで、初回試行時に比較して性能が大きく向上した。特に前処理を行うことで、これまで正確に認識できていなかった「=<」、「>=」、「<>」、「^=」等の記号の認識度を向上させることができた。

出力メッセージの中には、「Date of～」や「～Time」といった変数名で判断し、数値に～days や～minutes 等の単位を出力されているものもあった。

LLM での出力メッセージは、疑似コードを文章化したような表現となり、疑似コードが複雑であると、出力メッセージも長文となった。

例えば、以下のような場合

●疑似コード

```
[Is the adverse event serious?] = "YES" AND [Congenital Anomaly or Birth Defect] = "NO"
AND [Persist or Signif Disability/Incapacity] = "NO" AND [Did this serious event result in death?]
= "NO" AND [Requires or Prolongs Hospitalization] = "NO" AND [Is Life Threatening] = "NO"
AND [Other Medically Important Serious Event] = "NO"
```

●期待する出力メッセージ

[Is the adverse event serious?] is marked as "YES", but responses to all of the seriousness criteria are NO. At least one seriousness criterion is expected to be "YES" for serious events. Please review and update as appropriate.

各 LLM の結果は以下ようになった。

●Mistral

The [Is the adverse event serious?] is "YES", yet the [Congenital Anomaly or Birth Defect] is "NO", the [Persist or Signif Disability/Incapacitation] is "NO", the [Did this serious event result in death?] is "NO", the [Requires or Prolongs Hospitalization] is "NO", the [Is Life Threatening] is "NO", and the [Other Medically Important Serious Event] is "NO". Please review.

●Code Llama

The [Is the adverse event serious?] is "YES", the [Congenital Anomaly or Birth Defect] is "NO", the [Persist or Signif Disability/Incapacity] is "NO", the [Did this serious event result in death?] is "NO", the [Requires or Prolongs Hospitalization] is "NO", the [Is Life Threatening] is "NO", and the [Other Medically Important Serious Event] is "NO".

Please review.
<p>●Llama2</p> <p>The [Is the adverse event serious?] is "YES", yet the [Congenital Anomaly or Birth Defect] is "NO" and the [Persist or Significant Disability/Incapacity] is "NO" and the [Did this serious event result in death?] is "NO" and the [Requires or Prolongs Hospitalization] is "NO" and the [Is Life Threatening] is "NO" and the [Other Medically Important Serious Event] is "NO". Please review.</p>

上記の結果はいずれも誤った文章ではないため、受け入れ可能と判断したが、実際の臨床試験で使われているような表現とは異なる結果も存在した。

前処理および例示追加により、より高い精度の結果を得ることができたものの、誤った出力メッセージが表示されるものもあった。以下に、前処理および例示追加後に受け入れ不可となったメッセージのうち、いくつか主な事例をピックアップして以下の表中で紹介する。

- 表 3.2.7-1 : 受け入れ不可事例 (Mistral)
- 表 3.2.7-2 : 受け入れ不可事例 (Code Llama)
- 表 3.2.7-3 : 受け入れ不可事例 (Llama2)

表 3.2.7-2 の Code Llama の 1 つ目の事例では、単位の間違いであったが、よく似た疑似コードでは、単位を正確に出力していた。また 2 つ目の文章のようにになっている長文変数を変数として認識できなかった事例は、Code Llama 以外の Mistral および Llama2 でも同様の現象がみられた。

これら受け入れ不可は一貫性が無く、実行する度に異なる結果となった。さらに精度を上げるためには、10 例の例示よりもさらに例示を増やすことで実現可能と考えている。

表 3.2.7-1 :

#	疑似コード	期待したメッセージ	Mistral の結果	受け入れ不可理由
1	[Visit Date] <> null or [Visit Date] < [Study Day -28] or [Study Day -2] < [Visit Date]	The [Visit Date] did not occur on or between [Study Day -2] and [Study Day -28].	The [Visit Date] is not null and is earlier than 28 days prior to the [Study Day], or the [Study Day] is earlier than 2 days prior to the [Visit Date]. Please review.	[Study Day -28]が変数名として認識されなかった
2	[Lab Test or Examination Name] = "Lipase" and ([Result] < 4 or [Result] > 165)	The 'Result' for the [LBTEST] is outside the expected range. Please update or confirm as appropriate.	The [Lab Test or Examination Name] is "Lipase" and the [Result] is not within the expected range of less	not within the expected range であれば、greater than 4 or less than 165. となる

		than 4 or greater than 165. Please review.	
--	--	-----------------------------------------------	--

表 3.2.7-2

#	疑似コード	期待したメッセージ	Code Llama の結果	受け入れ不可理由
1	[Assessment Times for Measurement 4] - 5 >= [Assessment Times for Measurement 3]	[Assessment Times for Measurement 3] through [Assessment Times for Measurement 4] are more than 5 minutes apart.	The [Assessment Times for Measurement 4] is at least <u>5 days</u> after the [Assessment Times for Measurement 3]. Please review.	単位の間違い。5 days ではなく 5 minutes
2	[Date of Relapse] = null and [Has the participant experienced disease relapse during follow-up?] = "YES"	[Date of Relapse] is missing yet response to ['Has the participant experienced disease relapse during follow-up?'] is "YES".	The [Date of Relapse] is null and <u>the participant has experienced disease relapse during follow-up</u> . Please review.	変数名として認識されなかった
3	[Date of Sample Taken] - 2 <> [Date of Visit]	[Date of Sample Taken] is not 2 days after the [Date of Visit].	The [Date of Sample Taken] is not equal to <u>[Date of Visit] minus 2</u> . Please review.	[Date of Sample Taken]マイナス 2 となる

表 3.2.7-3

#	疑似コード	期待したメッセージ	Llama2 の結果	受け入れ不可理由
1	[BMI] < 18 or 32 < [BMI]	[BMI] appears to be out of range.	The [BMI] is less than 18 or the [BMI] is <u>less than 32</u> . Please review.	less than 32 が間違いである
2	[Was Assessment Performed] is "NO" and ( [Date of Assessment] is not null or [APACHE II Total Score] is not null )	Response to [Was Assessment Performed] is "NO", but [Date of Assessment] or [APACHE II Total Score] is provided. Please update as appropriate.	The [Was Assessment Performed] is "NO" and ( [Date of Assessment] is not null or [APACHE II Total Score] is not null ). Please review.	疑似コードそのままであり、文章が生成されなかった

3	[Date of Sample Taken] -2 ◁ [Date of Visit]	[Date of Sample Taken] is not 2 days after the [Date of Visit].	The [Date of Sample Taken] is not <u>2 days</u> <u>prior</u> to the [Date of Visit]. Please review.	2 days prior ではな く、2 days after で ある
---	------------------------------------------------	-----------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------	--------------------------------------------

### 3.2.8. 考察（大規模言語モデルによるアプローチ）

LLM モデルによる初回の試行結果はファインチューニングモデルによる結果には劣ったものの、Code Llama はわずか7件の例示のみで8割程度の受容率となったことは驚きであった。この結果はファインチューニングモデルで必要とした大量の学習データの準備の苦勞を考えると、LLM モデルの高い有用性を示唆するものと思われた。

「=<」、「=>」、「<>」、「^=」のような記号を変換し、また3例の疑似コードとクエリメッセージの組み合わせを追加して試行した結果では、受容率は Code Llama では 79.2%から 88.8%に、Llama2 では 68.8%から 90.4%に向上したことから、LLM モデルでは、多くのプログラム言語で広く使われる表現を用いた方がよいこと、また学習データの数量を増やしてパターンを学習させていくファインチューニングモデルと異なり、疑似コードとクエリメッセージの事例の包括性が重要であることが示唆された。

生成 AI では、プロンプトの表現が結果に与える影響を事前に明確に予測することは困難であり、意図する結果を得るために試行錯誤を繰り返す必要があることがあらためて認識できた。特に複雑な条件のクエリメッセージでは、ファインチューニングモデルほどには我々が実際の臨床試験で用いられているような表現のクエリメッセージを生成しなかったため、出力結果を制御するためには、プロンプトでの指示の仕方により工夫が必要であることも認識された。

### 3.2.9. ファインチューニングと大規模言語モデルの両アプローチの比較考察

T5 ファインチューニングアプローチは、実際によく使用されるクエリメッセージで訓練させることから、生成されるメッセージの表現も自然な感じのものが比較的多い。十分な量の学習データがあれば人間が介在することを前提に実用できる可能性もあるかもしれないと思われた。特に正しく疑似コードからクエリメッセージを生成できることがある程度確認できているパターンであれば、安定して正しいクエリメッセージを自然な表現で生成できるのではないかと思われた。しかしこのアプローチは、学習データの量、バランス、質がモデルの性能に大きく影響するため、実際の業務で活用するためには、十分な量の、偏りのない、一定のルールで作られた学習データのデータベースのメンテナンスが必要になると思われた。

一方、大規模言語モデルアプローチでは、より一般的な言語理解能力を持つ大規模な言語モデルを使用して論理チェック仕様からクエリメッセージを生成することから、疑似コードとクエリメッセージの組み合わせを学習する必要もなく、ある程度の品質のクエリメッ

セージを作成できるようである。しかし、疑似コードを機械的にテキストに置き換えているようで、言語として自然さを欠くメッセージとなってしまうことも少なくない。また同じ疑似コードからクエリメッセージを作成する場合でも、時に微妙に異なるメッセージを生成しているような印象があり実行するたびに結果が異なっていたため、安定性の点では T5 ファインチューニングアプローチに軍配があがるのではないかと思われた。

2つのアプローチを通して、学習データの用意の可否や、メッセージをどこまで自然な表現にしたいかといった、ユーザー側の要求を踏まえてアプローチを選択していく必要があることが認識された。

### 3.3. 添付文書一覧を活用した併用薬剤の使用目的の妥当性を予測するAI

#### 3.3.1. 目的

帳票による点検が行われる代表的な業務の一つである「併用薬剤の使用目的の妥当性の判定」について、AI を用いた点検の実現性を検討する。

#### 3.3.2. 背景

「併用薬剤の使用目的の妥当性の判定」は、主に当該薬剤の添付文書を閲覧し、効能または効果に記載されている文章を読み、CRF に記載された疾患や症状に対して有効かどうかを判定する、というステップで行われることが多い。従来のプログラムでは、これらのステップの中で「効能または効果に記載された文章に基づいて使用目的の妥当性を判断すること」が特に困難であった。その理由として、以下の内容が挙げられる。

- 医薬品の有する効能または効果が複数の症状や疾患にわたること
- 添付文書の効能または効果に記載される表現は、製造販売者によって異なること
- CRFのデータ収集において、薬剤名やその使用目的はフリーテキストで収集されており、特に使用目的は医学辞書コーディングの対象とされていないことも多いこと

これらの課題が AI によって解決可能であれば、少なくとも承認された効能または効果に基づいた使用目的の妥当性の判定は自動化が可能になり、人はより高度な点検に専念することができる考えた。

#### 3.3.3. 方法

前述の通り、「併用薬剤の使用目的の妥当性の判定」をこれまで機械的に実施することが困難であった点は、多岐にわたる添付文書の表現と、フリー記載で収集される CRF の情報を照らし合わせて有効性を判定しなければならない点である。本プロトタイプでは、この課題に対応するために、まず添付文書のデータを用いた機械学習モデルの構築を試みた。しかしながら、機械学習モデルを構築するまでには多くのハードルがあることが判明し、最終的に大規模言語モデルを使用するアプローチをとった。本章ではそれぞれのアプローチにおいて検討したことを詳説する。

### 3.3.3.1. アプローチ1：機械学習モデルの構築

このアプローチにおいては、添付文書の効能または効果をもとに、薬剤の一般名を個々の疾患や症状に一对一で紐づける表を作成し、それを学習データとするAIを構築することを考えた。このAIモデルでは、薬剤の一般名が与えられたときに、その使用目的として妥当な疾患や症状を予測し、その結果とCRFで記載される使用目的を照らし合わせてその妥当性を予測させる想定であった（図3.3.3.1-1）。

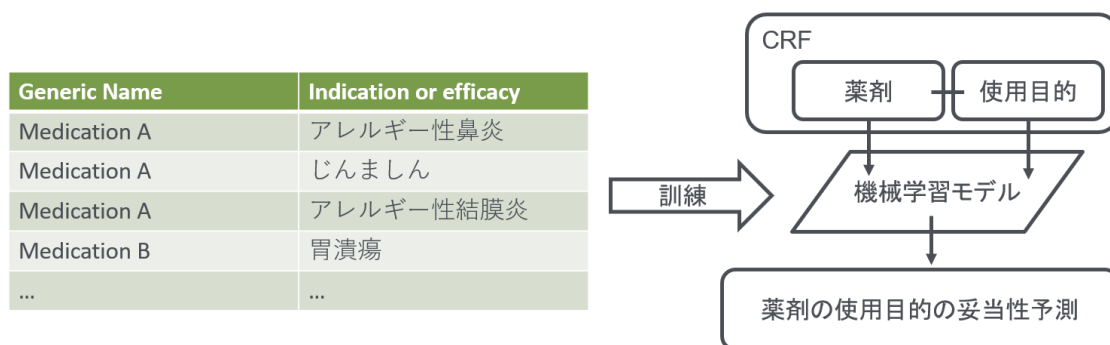


図 3.3.3.1-1

しかしながら、実際に添付文書のデータを取得し、学習データの作成を検討する中で、このアプローチが非常に困難であることが判明した。その理由が以下の点である。

1. 自然言語処理の手法を用いて効能または効果に記載される疾患情報を抽出することを想定していたが、「●●に伴う××」といった特定の疾患（●●）に伴う症状（××）に対する適応を持つ薬剤の場合は、適応にない疾患（●●）まで抽出されてしまうことがある。そのため、正確な学習データを作成するためには、最終的に抽出された情報が正しいことを1つ1つ人の目で確認する必要がある。
2. 添付文書の効能または効果に記載される症状や疾患は、MedDRA等で一般化された用語ではなく、重症度やその条件によって表記が異なることがある。機械学習モデルの構築において効果的な学習を行うためには、同様の効能を同じ表現に統一することが望ましかったが、これは効能または効果に記載された各症状や疾患に対して医学辞書コーディングと同等の作業を行う必要がある。
3. CRFで収集される疾患に対しても同様に、効果的な判定を行うためには収集された使用目的に対して上記と同等の作業を行うことが望ましく、そのような作業は実務面での煩雑さを考慮すると取り入れるハードルは高い。
4. 機械学習モデルを構築するアプローチでは、新薬や追加の効能または効果があった場合に学習モデルを再度構築しなおす必要があり、そのたびにモデルの評価を行う必要があると考えられる。新薬が増えるたびにモデルの再構築と評価を行うこと

は、頻繁なモデルの更新とそれに伴うリソースが必要になる対応であり、この観点からも実務における汎用性が薄かった。

このような理由から機械学習モデルを構築するアプローチは断念した。この時点において、本プロトタイプは実現性が薄いのではという議論もあったが、Llama2に代表される自由に利用可能な大規模言語モデルの使用に活路を見出した。

### 3.3.3.2. アプローチ2：大規模言語モデルを用いるアプローチ

大規模言語モデルの中でも、チャット形式で応答可能に訓練されたモデルでは、ユーザーが問い合わせた内容（プロンプト）に対して回答が生成される。これを応用して、プロンプトの中に添付文書の効能または効果の内容をそのまま埋め込み、それをもとに使用目的に対する妥当性を問い合わせることで、併用薬剤の使用目的の妥当性を予測してくれることを期待した。

このアプローチの利点は学習データの用意が不要になる点である。必要に応じて、判定したい薬剤の添付文書の効能または効果をそのままプロンプトに埋め込むことができればよく、また、CRFで収集された使用目的も同様にそのまま埋め込むことで記載のまま使用することができる。これによって機械学習モデルを構築するアプローチが困難であった理由の1から3がすべて解消された。また、機械学習モデルを作成しないことから4の汎用性についても課題を解決可能である。また、このアプローチは判定したい薬剤が与えられたときに、その添付文書にある効能または効果を取り出すことができれば実現可能であり、効能または効果の参照先をOpen FDA Label等の各国の承認薬の情報に置き換えることで、国ごとの承認内容に基づいた点検も可能となる汎用性を持ったアプローチとなる（図.3.3.2-1）。

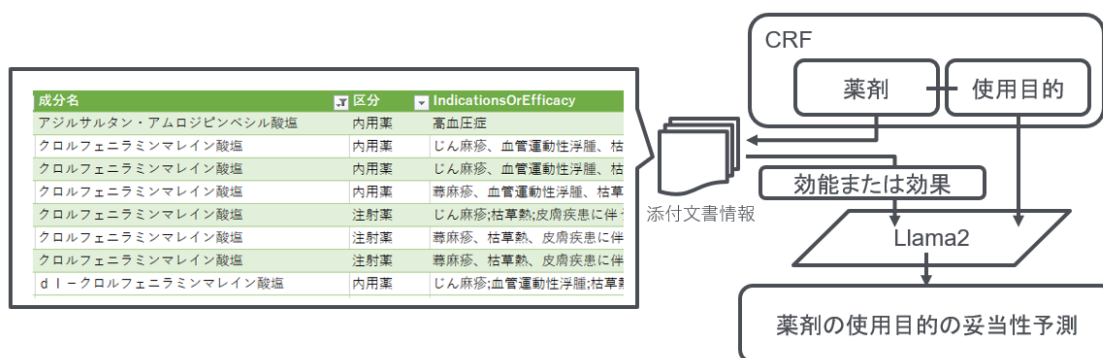


図 3.3.2-1

#### 3.3.3.2.1. 判定したい薬剤の効能または効果の取得

判定したい薬剤が与えられたときに、その添付文書にある効能または効果を取り出すことができるようにするために、まずは本邦で承認されている薬剤の一般名、およびその効能または効果の一覧表を作成が必要であった。

現在 PMDA が公開している添付文書は、マイ医薬品集作成サービスからダウンロード可

能である。まずは XML/SGML 形式で公開されているすべての添付文書を入力し、一般名とその効能または効果の一覧表を作成することを試みた。しかしながら、添付文書に記載されている薬剤の一般名は自由記載であり、同じ一般名でも表記が統一されていないことが分かった。そのため、薬剤辞書コーディングによって一般名の集約を行う必要があったが、本タスクフォースではライセンスの制約上、医薬品データファイルや WHO Drug Dictionary を使用することができなかった。そのため、薬価基準収載医薬品リストの成分名を一般名として代用し、薬価基準収載医薬品コードと YJ コードを結合キー情報として添付文書の医薬品の一般名化を行った<sup>19</sup>。

これによって一般名とその一般名の持つ効能または効果の一覧表を作成し、当該薬剤が提供された際に効能または効果を取り出すプログラムを構築した。

### 3.3.3.2.2. プロンプトへの埋め込み

プロンプトを作成するプログラムでは、対象薬剤の効能または効果と CRF で収集された使用目的をプロンプトの雛型に埋め込む形式とした<sup>20</sup>。

今回のプロトタイプングで使用した大規模言語モデルは Meta 社が公開した Llama2 であり、日本語の応答は一部可能であるものの、OpenAI 社の ChatGPT に代表される高性能な大規模言語モデルほど高度な応答は困難であった。そのため、基本的な指示は英語で記載し、適格性の判定においては順を追った質問をすることで正答率の向上を期待している (図 : 3.3.3.2.2-1)。

また、これによって得られる回答の 1 例を図 3.3.3.2.2-2 に示した。

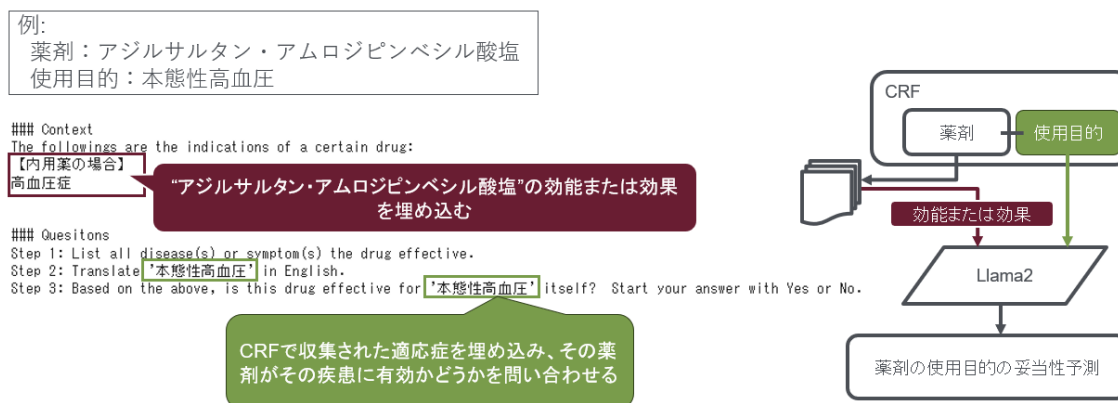


図 3.3.3.2.2-1

19 プログラム : [JPMA2023\\_case3 \(1\) 添付文書データの準備.ipynb](#)

20 プログラム : [JPMA2023\\_case3 \(2\) LLM を用いた効能または効果の確認.ipynb](#)



Sure, I'd be happy to help!

Step 1: List all disease(s) or symptom(s) the drug effective.

Based on the indications provided, the drug is effective for:

\* High blood pressure (hypertension)

Step 2: Translate '本態性高血圧' in English.

'本態性高血圧' can be translated to "essential hypertension" in English.

Step 3: Based on the above, is this drug effective for '本態性高血圧' itself?

Yes, the drug is effective for essential hypertension.

図 3.3.3.2.2-2

### 3.3.4. 結果

2023年10月時点で本アプローチによる判定が可能であった薬剤の一般名（1819件）について、「糖尿病」に対する効果の有無を問い合わせ、AIによる予測結果を得た。また、同時に人間による判定も行った。具体的には、効能または効果に「糖尿病」の記載が無いものは効果なしと判定し、「糖尿病」の記載を含む薬剤の一般名については1件ずつ文章を読み取り、「糖尿病」そのものに対する適応を有するか判定を行った。以下の表がAIによる予測と添付文書上の記載に基づいた判定結果である。

表 3.3.4-1

		添付文書を参照した AI による予測	
		効果なし	効果あり
添付文書上の記載	効果なし	1736	21
	効果あり	3	59

### 3.3.5. 考察

結果では、一定の精度は見込めるものの、あたかも正しい答えかのように回答してしまうハルシネーションも認められているため、実務に取り入れる上ではこれらの制御方法も併せて検討する必要があることが分かった。いくつかのケースでは、「糖尿病」という単語が含まれているにもかかわらず「効果なし」と判定しているケースが見受けられた。また、糖尿病の治療に直接的な適応を持たない薬剤に対して「糖尿病に併発する疾患の治療に使用する薬剤であるため効果あり」と予測する、添付文書以外の情報を利用して判定をされているようなケースも見受けられた。そのため、より厳密性を求めるのであれば、完全一致検索の併用なども検討することが良いと考えられる。一方で、このような点検を100%の精度で行うことは人の目をもってしても時間と労力を要するものであり、この点検にどれほどの

価値があるのか、どこまでの正答率を求めるか、という観点での検討も忘れてはならない。目的に応じた点検の精度を考え、どの程度の深さで点検を行う必要があるのかを見定めることで、AIの実務適用も行いやすくなるものと考え

### 3.3.6. まとめ

今回のプロトタイプングによって、これまでは困難と思われてきた帳票による点検の自動化も夢ではないことが確認された。古典的なプログラムのように確実性のある結果は得られないが、それはある意味で人が行う点検にも同様のことが言える。不確実なものをどのようにプロセスに組み込んでいくのか、よりリスクを意識したアプローチを念頭に置いて業務の自動化を検討していく必要があるだろう。結果と考察のセクションではハルシネーションや精度について言及しているが、今回得られている結果は自由に使用可能な大規模言語モデルで実施したものであり、GPT-4 に代表される高性能な大規模言語モデルを使用し、プロンプトを練りこめばこれ以上の結果が得られることは想像に難くない。大規模言語モデルの性能も日々向上している中、精度やハルシネーションが懸念されるという理由だけで活用を避けていくことはもはや時代遅れの議論といっても過言ではなく、いずれ人の作業を置き換える技術であることは確実であると考え。一方で、人がミスをすると同様にAIも完璧ではないことも事実であり、どこでどのように採用していくかはリスクに応じて検討することが必要である。AIの採用にあたり、そのリスクを懸念されている読者も多いと思うが、人が実施する作業においても「人のミス」が起こすリスクがあることを念頭におき、各手順において許容可能なリスクを熟慮することがAIの採用を検討する第一歩となるであろう。

## 4. 全体のまとめ

今回のBERTおよびT5によるファインチューニングアプローチにおいては、ライセンス権および各社の社内機密情報などの観点から使用できるデータは限定的ではあったものの、「疾患名の辞書コーディング結果を予測するAI」ではTop 1、Top 3、Top 5における正答率のそれぞれ82.4%、90.7%、92.4%であり、また「論理チェック仕様からクエリメッセージを作成するAI」では生成された89.6%のクエリを受け入れ可能と判断することができた。同様のモデルを構築する際に各社で蓄積されている社内のデータおよびライセンス権が制限なく利用できるのであれば、より高い精度のAIモデルの構築が期待できる可能性がある。

今回使用したLlama2などのLLMは学習データを用意することもなくある程度の精度の結果を出すことができた。今後も同様なLLMの発展が期待でき、またノーコード/ローコードによるAI開発がより身近なものになることで、様々な場面でこれらのモデルを活用できる可能性があることが示唆された。

AI を実際の業務で活用するためには、AI に関する技術的な知識に加えて、他者の著作権、ライセンス権、または人権等を侵害しないための知識や調査が必要となることが改めて認識された。また、AI は完全ではなく、利用する状況に応じて **Risk-based Approach** を取る必要があり、判断や制御に人間の介在が必須であることを実感した。しかし、一方で AI は日々進化しており、AI に関わる様々な問題を理解した上でリスクを適切に管理しながらこれを活用し、それによって業務を改革していくことが今後のデータマネージャーに求められることであろうと思われた。

AI に関する知識以上に重要なことは、マニュアル作業に依存している非効率な現状に対する問題意識とそれに対する解決策を熟考することである。このような問題意識があれば、AI はデータマネジメント業務に革新的な変化を生み出す未来をもたらすものと思われた。

## 5. おわりに

「はじめに」に記載したように、本書は読者の皆さまがこれを読むことでプロトタイプングを「追体験」できることを期待して作成された。よって本書の中には全体の分量の割にかなり細かく記載されている箇所があると思う。また、考察の内容にも心情の吐露や正直な驚きがそのまま表現されている箇所もあることも承知している。「追体験」できるようにという意図から、敢えてそういう箇所はそのまま残すことにしたので、読者の皆さまにおかれてはご了承ください。

当初の想定どおり、筆者らはこのプロトタイプングを通じて、ライセンス、著作権および社内機密情報といった知的財産的な問題、学習データ等を作成に要する労力、日本語処理に特有な技術的な問題、AI 構築並びに実行に利用するコンピューター環境の制限等、様々な課題に実際に直面していった。一方、更に良い結果を出すために様々な工夫を凝らすことによってモデルの成績を向上させられることも実感できた。所属会社での本来の業務を抱えながらの製薬協活動という時間制限がある中で、最終的には当初の想像以上の成績を出すことができたと思っている。今、このプロトタイプングを振り返ると、多くの悩み、苦勞、苦しみさえも思い出されるが、様々な工夫によって成績が向上していくたびにメンバーと喜びを分かち合ったことも良い思い出となった。

本書に記載された内容が読者の皆さまの参考となり、AI モデルの構築がより身近な存在に感じられるようになっていただけることを心から願っている。また読者の皆さまが本成果物を読むことで、筆者らが本活動を通じて経験した試行錯誤や悩みも含めて「追体験」していただけたら幸いである。そして次は自分たちで「実体験」してみようという気分になっていただけたなら、筆者らにとってこの上ない喜びである。

## 参考文献

- [1] D. e. al, “Artificial intelligence in drug discovery and development,” October 2020. [オンラ

- イン]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7577280/>.
- [2] European Medicines Agency, “Reflection paper on the use of AI in the medicinal product lifecycle, ” July 2023. [オンライン]. Available: [https://www.ema.europa.eu/en/documents/scientific-guideline/draft-reflection-paper-use-artificial-intelligence-ai-medicinal-product-lifecycle\\_en.pdf](https://www.ema.europa.eu/en/documents/scientific-guideline/draft-reflection-paper-use-artificial-intelligence-ai-medicinal-product-lifecycle_en.pdf).
- [3] M. Hutson, “How AI is being used to accelerate clinical trials,” March 2024. [オンライン]. Available: <https://www.nature.com/articles/d41586-024-00753-x>.
- [4] Kitameraki, “Pfizer and others leading the pharmaceutical manufacturing with AI and Technology,” May 2023. [オンライン]. Available: <https://www.kitameraki.com/post/pfizer-and-others-leading-the-pharmaceutical-manufacturing-with-ai-and-technology>.
- [5] V. Narasimhan, “Novartis CEO explains how AI will impact drug development,” 2024, January. [オンライン]. Available: <https://finance.yahoo.com/video/novartis-ceo-explains-ai-impact-151559920.html>.
- [6] Bloomberg, “World’ s Pharma Giants Bet on AI to Develop Drugs Faster,” 5 2023. [オンライン]. Available: <https://www.bloomberg.com/news/newsletters/2023-05-15/world-s-pharma-giants-bet-on-ai-to-develop-drugs-faster>.
- [7] 日本製薬工業協会 データサイエンス部会, “データマネジメントにおける Artificial Intelligence の活用 ～ これから始める AI ～,” 5 月 2023. [オンライン]. Available: [https://www.jpma.or.jp/information/evaluation/results/allotment/g75una00000024me-att/DS\\_202305\\_2022TF1\\_1\\_AI\\_DM.pdf](https://www.jpma.or.jp/information/evaluation/results/allotment/g75una00000024me-att/DS_202305_2022TF1_1_AI_DM.pdf).
- [8] 世界保健機関, “疾病及び関連保健問題の国際統計分類 : International Statistical Classification of Diseases and Related Health Problems ICD-10 (2013年版) 準拠 基本分類表,” 2013. [オンライン]. Available: <https://www.mhlw.go.jp/toukei/sippei/>.
- [9] 厚生労働省政策統括官 (統計・情報政策担当) 付参事官 (企画調整担当) 付 国際分類情報管理室, “疾病、傷害及び死因の統計分類の正しい理解と普及に向けて (ICD-10 (2013年版) 準拠) , ” [オンライン]. Available: [https://www.mhlw.go.jp/toukei/sippei/dl/ICD-10\\_2013\\_2802.pdf](https://www.mhlw.go.jp/toukei/sippei/dl/ICD-10_2013_2802.pdf).
- [10] NEologd, “mecab-ipadic-NEologd : Neologism dictionary for MeCab,” NEologd, 20 8 2020. [オンライン]. Available: <https://github.com/neologd/mecab-ipadic-neologd>.
- [11] “Index of /~data/2018-manbyo/data,” NAIST Social Computing Lab., [オンライン]. Available: [http://sociocom.jp/~data/2018-manbyo/data/MANBYO\\_201907\\_Dic-utf8.dic](http://sociocom.jp/~data/2018-manbyo/data/MANBYO_201907_Dic-utf8.dic).
- [12] 日本製薬工業協会 データサイエンス部会, “AI ってなに?,” 日本製薬工業協会, 2019.
- [13] 日本製薬工業協会 データサイエンス部会, “機械学習始めました,” 日本製薬工業協

会, 2020.

タスクフォースメンバー

会社名	氏名	役割
グラクソ・スミスクライン株式会社	久田 大輔	推進委員 (DM)
ファイザーR&D 合同会社	三沢 秀敏	推進委員 (DM)
EA ファーマ株式会社	橘田 由香里	メンバー (DM)
ノバルティス ファーマ株式会社	井上 祐介	メンバー (DM)
ファイザーR&D 合同会社	田中 拓海	メンバー (プログラマー)
アステラス製薬株式会社	柴田 奈津実	メンバー (DM)
バイエル薬品株式会社	松尾 洋太朗	メンバー (プログラマー)
ヤンセンファーマ株式会社	那知 新也	メンバー (DM)
大塚製薬株式会社	京坂 侑土	メンバー (プログラマー)
ファイザーR&D 合同会社	小宮山 靖	担当副部長

※上記 DM/プログラマーは所属会社における役割ではなく本 TF での以下の役割を示す

DM:学習データセットの作成やテスト等を担当

プログラマー: AI モデル開発/選択担当

謝辞

本成果物の公表に際し、査読いただいたデータサイエンス部会運営幹事会ならびに医薬品評価委員会副委員長に御礼申し上げます。