研究コース５（類似度活用）

繰り返しのテストを要する生成 AI テストの効率化

- 類似度算出と同義文判定による検証コスト削減の検討 -

Efficiency Improvement of Generative AI Testing Requiring Repeated Tests

- Consideration of reducing verification costs through similarity calculation and synonym sentence determination -

リーダー：中川　桂（東京海上日動システムズ株式会社）
研 究 員：金丸　優介（株式会社 AGEST）
　　多田　麻沙子（TIS 株式会社）
主　査　：石川　冬樹（国立情報学研究所）
副 主 査：徳本　晋（富士通株式会社）
アドバイザー：栗田　太郎（ソニー株式会社）

研究概要
近年，システム開発ではチャットボットなどで生成 AI（Generative AI）による文章生成機能が利用されている．生成 AI によって自動生成される回答について，その膨大なテキストデータ量と多義性，確率性によって従来のソフトウェアテストによる品質評価が容易ではない．この課題に対して我々は，類似度をテストでの品質評価に用いることを検討した．本研究では，その評価として埋め込み表現のコサイン類似度での評価と生成 AI による類似度評価について数値化し，実験と結果考察を行った．実験の結果，生成 AI による類似度評価は，長文化によるスコア増加の課題などはあるものの人間の直感的理解をサポートしつつ，実務にて利用できる一定の可能性があることが示された．

1.　はじめに

生成 AI が生成する膨大なテキストデータを人間が一貫して評価することは，時間的および認知的コストが高く，実務的には負担が大きい．従来のソフトウェアテスト手法は，明確な入力と期待される出力の一致を評価するものであったが，生成 AI の出力の多義性や確率性はその手法を困難にしている．本研究では，既存のテキスト生成の自動評価指標[1]の調査から始まり，既に広く活用されている埋め込み表現のコサイン類似度や生成 AI を用いたテキスト類似度評価の一部について定量化が可能であり，テスト省力化を目的に使用することを検討した．この実証実験と精度の比較を行いその妥当性や有効性を検証した．

以降 2 章では研究の背景を説明し，3 章では類似度評価について研究課題立案をおこなう．4 章で実験の内容と結果を受けた考察を示し，5 章では研究成果のまとめと今後の展望について述べる．

2.　背景

2.1 生成 AI とは

生成 AI とは，主にディープラーニング技術を用いて新たなデータ（文章，画像，音声など）を自動生成するモデル群の総称である．Chat-GPT（Chat Generative Pre-trained Transformer）が代表例として挙げられる．2024 年現在，企業における社内照会応答チャットボットや社内マニュアル回答チャットボットなど，文章生成機能を活用したシステム開発事例[2-1,2,3]がある．こうした事例においては，自然言語で回答が生成され，その内容をテストで検証する事となる．

2.2 生成 AI のテストとは

生成 AI を実システムで活用する際，ソフトウェアテスト手法の一環として要求仕様を満たす出力を得られているかどうかを確認する必要がある．しかし，従来型のソフトウェアテストが主に「入力 A に対して出力 B を期待する」という明確な定義に基づくのに対し，生成 AI の出力は多義的・確率的であるため，単純な“期待値”による評価が困難である．例えば ChatGPT に進捗の評価させた場合の出力は，入力が同じでも「全体としては悪いとは言えませんが，良いとも言い難い状況」や「問題解決に向けた積極的なアプローチが取られている点は評価できます」と実行のたびに出力が変化する事が起こり得る．よって生成 AI においては毎回回答文章が変化することを前提に複数回のテストが必要となる．

特に UAT（ユーザアクセプタンステスト）で利用ユーザにこの特性を説明し，従来型以上にテストにコストをかけてもらう場面も生じる．しかし，生成 AI が生成する膨大なテキストデータを人間が一貫して評価することは実務的に負担である．こうした負担感の軽減検討が本研究のきっかけとなっている．

生成 AI のテストが複数回を要する事については AI プロダクトの品質保証コンソーシアムである QA4AI（Quality Assurance for AI）[3]が公開するガイドラインにも言及がある．LLM（Large Language Model：大規模言語モデル）は同じ入力に対して異なる結果を生成することも特徴の 1 つであること，それに対して利用目的によっては LLM に対する問合せに対して安定的な回答が期待される場面があること，LLM を用いて安定的な結果を得るための検討は今後の課題であるとされている．

2.3 生成 AI のテストの効率化

生成 AI の出力文章が大量である場合，検証を人間だけで行うのは負担が大きい為負担の軽減策が望まれる．そこで考えられる手段の一つに「同義文判定」を行う仕組みの導入がある．本研究では同義文の判定をするために類似度を数値化し，「同義文判定」実現の助けとする．

生成 AI は多義的・確率的であるため，テストが一度成功するだけでは安定的に成功することの確認にならないため複数回の実行が必要である．しかし毎回回答の文章表現は変わることがあるため，文章の内容を読み内容が正しいかを人間が複数回読む事が必要となってしまいテスト工数が高くなる．そこで 「同義文判定」技術を使ったうえで，最初に一度検証済みの回答を用意できれば ，その後の回答は「同義文判定」にて同じと判定できれば機械的にテスト成功と整理することが可能となる．この「同義文判定」が人間の感覚に近い精度で機械的に実施できれば，テストの省力化が実現できる．

適用が期待される場面として，機能テストでの繰り返し行う検証ケース数の削減，また生成 AI モデルがバージョンアップした際のリグレッションを機械的に行う事も期待される．

3. 研究課題

生成 AI のテストの効率化のための同義文判定の方法として、テキスト間の類似度を数値化し判定する方法がある。類似度の数値化の方法として今回、埋め込み表現のコサイン類似度や生成 AI による評価に着目し,以下の研究課題を設定した．

(1) 研究課題 1：類似度の数値化．文章同士の類似度を埋め込み表現のコサイン類似度や生成 AI による評価で数値化し，人間の感覚に合うか（埋め込み表現のコサイン類似度の説明は後述する）

(2) 研究課題 1-1:コサイン類似度での評価．埋め込み表現のコサイン類似度で文章の類似度が人間の感覚に合うか

(3) 研究課題 1-2:生成 AI による類似度評価．文章の類似度の判定は埋め込み表現のコサイン類似度よりも ChatGPT による評価の方が人間の感覚に合うか．

(4) 研究課題 2 長文化の影響の評価．長文になると検出性能が落ちるのではないか

研究課題 1 は定性的な評価でなく,定数的な評価が可能であれば,テストを複数回実行し

て結果を評価する際に判断がしやすいため有益と考え研究課題に設定した．本研究課題は本研究の最も主要なテーマである．

研究課題 1-1 は文章の類似度の評価にあたっての具体的な評価モデルとして埋め込み表現のコサイン類似度で評価をした．コサイン類似度については 4.1.1.(1)にて説明する．

研究課題 1-2 は同じく文章の類似度の評価に当たって,具体的な評価モデルの 2 つ目として生成 AI による判断をあげ,評価をした．

研究課題 2 は実務で多く扱う長文でも検出が可能かを確認したい．類似度判定対象の切り出し方によっては数値が影響を受ける可能性もあると考えたため,本研究課題を設定した．

4. 実験

4.1 実験内容

本章では，文章類似度判定を行う２つの手法，すなわち埋め込み表現のコサイン類似度と生成 AI による類似度評価を比較し，人間の感覚とどの程度一致するかを実験した．

4.1.1 実験の前提

(1)本研究で使用する埋め込み表現のコサイン類似度の説明

本研究で使用するコサイン類似度とは，文章を埋め込みベクトルに変換したうえで，下記式によりベクトル同士の類似度を測定するものである．

$$cosine\ similarity = \frac{A \cdot B}{\|A\|\|B\|}$$

値が 1 に近いほど,ベクトルの向きが近い(文章の意味合いが似通っている)とみなす．逆に値が 0 に近い場合，ベクトルの向きが直交しており，文章の意味合いに関連性が低いとみなす．値が負になる場合，文章に意味合いは対立するとみなす．

(2)文章のベクトル化（埋め込みベクトル）

本研究では OpenAI の embedding API を利用して埋め込みベクトルを作成した．埋め込みベクトルを作成する OpenAI の Embedding モデルは，大規模なトランスフォーマーモデルを基盤として構築されており，入力テキストの文脈を捉えたベクトル化が可能となる．トランスフォーマーモデルは「自己注意機構（Self-Attention)」を使用している．自己注意機構とは入力された各単語が文中の他のすべての単語とどのように関係しているかを計算する仕組みである．これが文脈の把握に作用している．

この時類似する意味合いの文章は近しいベクトルとなることが OpenAI の embedding API[4]に示されており，この特性を本研究では使用している．

(3)生成 AI モデルによる類似度評価

生成 AI に「二文の類似度を 0～100 の数値で示す」様に指示し，モデルが返すスコアをそのまま類似度とみなす．LLM as a Judge と呼ばれる生成 AI の使用の仕方に該当する．

4.1.2 実験環境

開発言語および使用したライブラリ，実験に使用したコードは付録 1 に記載する．

4.1.3 実験設定

本研究では，以下の二つの OpenAI API 機能を活用した評価ツールを用意した．

- 埋め込み表現のコサイン類似度算出ツール（Embedding API 利用）：テキストを高次元ベクトルに変換するための埋め込みモデル．事前学習されたモデル（text-embedding-3-small と text-embedding-3-large）を利用し，入力文章のベクトルを取得する．得られたベクトル同士のコサイン類似度を算出する．

  text-embedding-3-small と text-embedding-3-large の違い[5]は性能と価格である．Small は性能は劣るものの価格にメリットがある．Large は逆である．今回の用途において結果に差が出るか比較する意図で二つを使用した．

- 生成 AI による類似度評価ツール(Chat GPT API 利用）:生成 AI（GPT-4o）に対して，プロンプトを入力し出力を取得する． 本研究では「生成 AI に文章ペアの類似度を 0～

1 の範囲で数値化させる」タスクを与えて，生成 AI 自体を“類似度判定器”として活用する．具体的には二つのプロンプトで実験した．

(a)文章構造と意味合い的類似度を意識させるケース（以降，ChatGPT 比較類似度）

「#命令:以下の条件で入力文 1 と入力文 2 を比較して文章構造の類似度と意味合い的な類似度をそれぞれ評価してください． #制約;;1，条件:・文章構造の類似度と意味合い的類似度は 1〜100 のスコアで示すこと・それぞれの評価の根拠を簡潔に記載すること・入力文 1 と入力文 2 は出力しない #入力文 1:{text1} #入力文 2:{text2} #出力文:スコア，根拠」

このプロンプトでは文章構造の類似度も出力されるが評価結果としては本研究では使用しない．

(b)意味合い的な類似度だけ意識させるケース（以降，ChatGPT 独立類似度）

「#命令:以下の条件で入力文 1 と入力文 2 を比較して意味合い的な類似度をそれぞれ評価してください． #制約条件:・意味合い的類似度は 1〜100 のスコアで示すこと・評価の根拠を簡潔に記載すること・入力文 1 と入力文 2 は出力しない #入力文 1:{text1} #入力文 2:{text2} #出力文:スコア，根拠」

4.1.4 実験データ

実験データは（1）文章作成，(2) ラベリング（正解データの付与）．(3) レビュー，(4) 長文化データの追加作成，の手順で計 147 件を作成した．

文章作成手順で実験データはジャンルを情報技術関連とし,質問と text1 を生成 AI で出力し,text2 を生成 AI もしくは人力で作成した.質問は回答文作成時の指針として利用し,この後の実験の評価には用いていない．

ラベリング手順で，text1,text2 の類似度の人間の評価として,類似度が「同じ」であるか「別」であるかをラベリングした.これが正解データとなる．

レビュー手順で,「同じ」か「別」かのラベリングが妥当かを,2 名でレビューし,一致を確認した.また,評価が異なった場合のみ,3 人目がレビューし,多数決でラベリングした．

長文化データの追加作成手順で，研究課題 2 の実験用に短文を(1)〜(3)の手順で追加作成した．さらに文脈的に合致する同じ文章を text1 と text2 の両方に挿入し,文章の長文化をしたものを長文化データとして実験データに加えた.また同様にラベリング，レビューを実施した．研究課題 2 では専用のデータのみを，研究課題 1 では研究課題 2 のデータを含めたすべてのデータを用いて評価を行った．

前述の類似度評価ツールを用い,Embedding 系モデルの text-embedding-3-small,text-embedding-3-large の 2 種および,ChatGPT 系モデルの ChatGPT 比較類似度,ChatGPT 独立類似度の 2 種類の合計 4 種類で類似度の評価を行った．実験データとその結果の一部は付録 2 に掲載している．

4.2 実験結果

4.2.1 モデルの性能-標準偏差，平均，箱ひげ図（研究課題 1）

まず研究課題 1 として，実験した結果の平均値，標準偏差は表 1 である．（正規化として，Embedding 系モデルは小数第 2 位までとした上で 100 倍した結果を掲載している.）

表1-実験結果の平均，標準偏差

人間の評価	text-embedding-3-small(100倍)	text-embedding-3-large(100倍)	ChatGPT比較類似度	ChatGPT独立類似度
平均（同じ）	94.4	94.6	93.3	93.6
平均（別）	83.7	81.1	73.4	74.8
平均（同じ）-（別）	10.7	13.5	19.9	18.8
標準偏差（同じ）	5.0	4.1	11.4	4.6
標準偏差（別）	12.6	13.5	20.4	19.0

平均は，Embedding 系モデルでは「同じ」と「別」の差が 10.7 と 13.5 であるのに対し，ChatGPT 系モデルでの差は 19.9 と 18.9 でと少し大きい結果となった．評価値の差が大き

く出るため，ChatGPT 系モデルの方が人間は判断しやすいと考えられる．

標準偏差は正解が「別」のケースでは，Embedding 系モデルの 12.6 と 13.5 に対し，ChatGPT 系モデルは 20.4 と 19.0 と少し高い結果となり，ChatGPT 系モデルの方が広く分布した． 正解が「同じ」のケースでは，Embedding 系モデルの標準偏差が 5.0 と 4.1 に対し，ChatGPT 独立類似度は 4.6 と似た傾向を示し， ChatGPT 比較類似度だけ 11.4 と広く分布した．「別」のデータは似た文章からかなり異なるものまでバリエーションがあるため分散しても問題ないが，「同じ」はバリエーションがあるような性質のものではないため標準偏差は小さい方が好ましく，ChatGPT 比較類似度はほかの 3 つのモデルに比較して不安定さが懸念される．

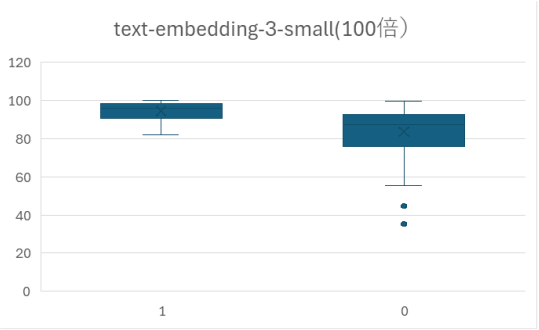

図1-"同じ"と"別"の場合別の類似度の分布(text-embedding-3-small)

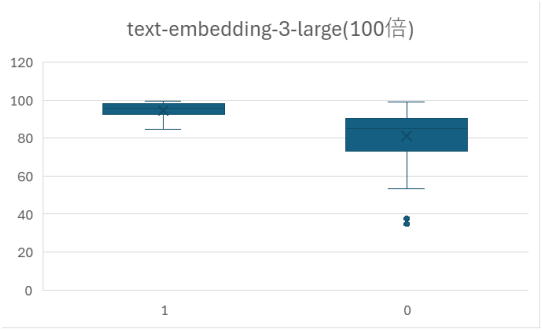

図2-"同じ"と"別"の場合別の類似度の分布(text-embedding-3-large)

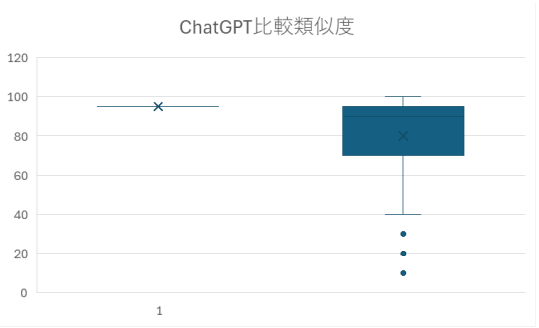

図3-"同じ"と"別"の場合別の類似度の分布(ChatGPT比較類似度)

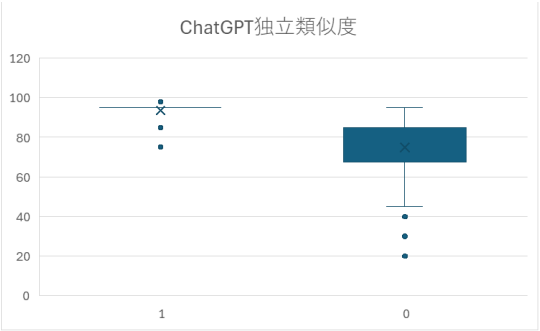

図4-"同じ"と"別"の場合別の類似度の分布(ChatGPT独立類似度)

視覚的に確認するために箱ひげ図にしたものが図 1～4 である．全体に共通する傾向として「同じ」と「別」を比較したとき，「別」は全体的にぶれが大きいことがよみとれるが，対象データの例文のバリエーションが多く，似ているといえるものから別物といえるものまであるため実情に即している．

本研究において理想的な状態とは，期待値として「同じ」と「別」の箱ひげ図の範囲が重複しないことである．しかしながら，結果はいずれの手法においても一部の範囲が重複した．この結果は，「同じ」と「別」を明確に区別するための閾値を一意に定めることが出来ないことを示している．

このような状況下では，比較的重複が少ない手法の選択が精度上有利である．本研究では Embedding 系モデルに比べて Chat-GPT に基づく手法が重複が少ない傾向が確認されたのでこちらを選択することが考えられる．

ただしこれらの箱ひげ図は，「同じ」と「別」を厳密に区別できる一意の閾値がない事を示しているので，実務上の特性に応じて閾値を設定する必要がある．

4.2.2 モデルの性能-ROC 曲線，AUC（研究課題 1）

引き続き研究課題 1 について，実験した結果を ROC 曲線で表したものが図 5～8 で，各モデルの AUC をまとめたものが表 2 である．

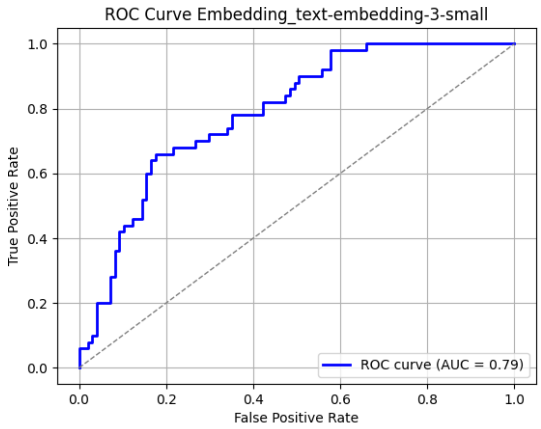

図5-ROC曲線(text-embedding-3-small)

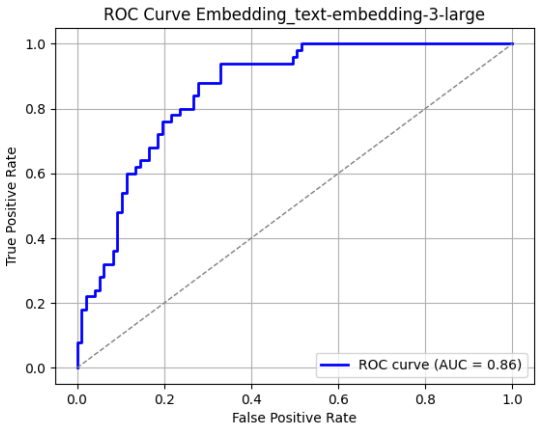

図6-ROC曲線(text-embedding-3-large)

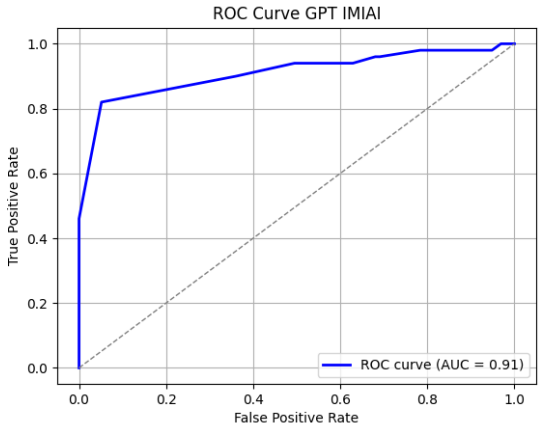

図7-ROC曲線(ChatGPT意味合い的類似度)

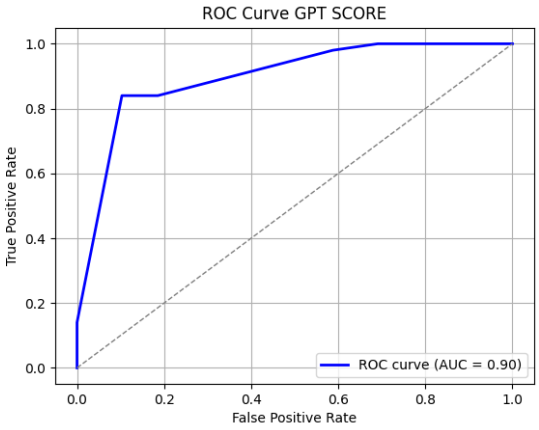

図8-ROC曲線(ChatGPTスコア)

前述の通り,ROC 曲線は該当モデルの性能を表す.ここでは類似度の閾値をとらえ，閾値を変化させることで得られる TPR と FPR のペアをプロットしたものである．イメージとして TPR は見逃さない性能,FPR は誤認の度合いを表す.縦軸は値が大きいほどよく,横軸は値が小さいほど良い.

ROC 曲線の下の面積が AUC だが,その面積が大きいほど,分類の精度が高いことを示す.結果は表 2 の通りである． AUC は ChatGPT 比較類似度が AUC0.91 で一番性能が良い結果となり,次点が ChatGPT 独立類似度の AUC は 0.90 である.傾向として,ChatGPT 系モデルと Embedding 系モデルでは ChatGPT 系モデルが良い結果となった.

表2-各モデルのAUC

モデル名	AUC
text-embedding-3-small	0.79
text-embedding-3-large	0.86
ChatGPT比較類似度	0.91
ChatGPT独立類似度	0.90

実務では，ROC 曲線のカーブを元に各モデルで類似度の閾値を設定し,機械的に同義文判定を行うことも可能だが,最高 0.91 の AUC のため，誤判定による影響をどの程度深刻にとらえるかは対象業務の特性に合わせて判断することを推奨する.

4.2.3 長文化による影響(研究課題 2)

研究課題 2 の長文化前の短文と長文化した文章について,人間の評価は前後で変更がない点に対し,各モデルによる評価は全体的に類似度が上昇した．(詳細な各データの変化は付録 3 に示した)

図 9～12 には長文化前の類似度から短文の類似度を減算した数値を，人間の評価が「同じ」,「別」のケース別に示した.

実験結果はほぼすべてのケースで上昇した.ChatGPT系モデルは「同じ」の増加幅の50%（第一四分位数～第三四分位数）が0～3,0～5,「別」が15～40，10～35だった．embedding-smallでは「同じ」が3～10.7,「別」は4.4～11,embedding-largeは「同じ」で，2.2～7.1,「別」は3.8～11.8となった．これよりChatGPT系モデルの「別」が最も長文化の影響を受けた．にEmbedding系は「別」の変化量がChatGPT系よりも少なく、ChatGPT系に比べ長文化の影響を受けにくかった.

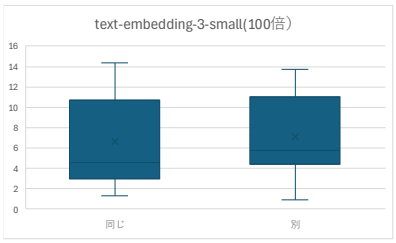

図9-長文から短文の類似度を減算した際の同じ，別の分布
text-embedding-3-small（100倍）

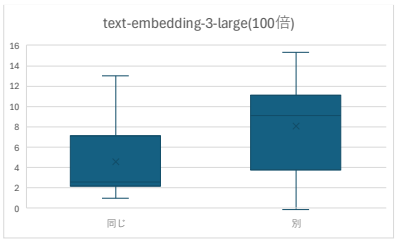

図10-長文から短文の類似度を減算した際の同じ，別の分布
text-embedding-3-large（100倍）

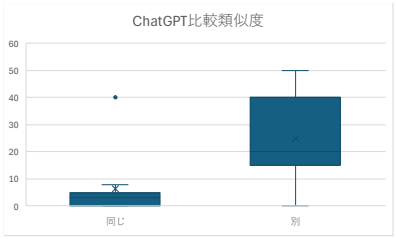

図11-長文から短文の類似度を減算した際の同じ，別の分布
ChatGPT比較類似度

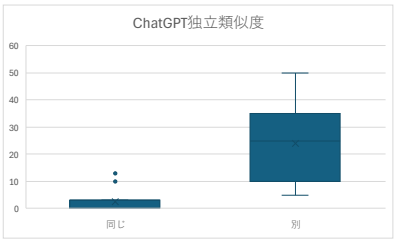

図12-長文から短文の類似度を減算した際の同じ，別の分布
ChatGPT独立類似度

「同じ」と「別」は類似度の数値が重複する範囲が少ない方が良いため，期待としては「同じ」は変化なしか類似度が上昇する方が良く,「別」は変化なし，もしくは類似度が減少することが望ましい.「同じ」は基本増加傾向のため問題ないが,「別」の増加量が検出性能の面で重要となる．今回,「別」では全てのモデルで類似度が増加したため，検出性能の劣化を確認したが，より顕著なのはChatGPT系モデルだった.

4.3 考察

研究課題1-1の埋め込み表現のコサイン類似度での評価については,AUCは0.79以上の結果を出すことができ、文章の類似度を測る用途使用できる可能性を示された。ただしChatGPTの結果と比較すると，箱ひげ図にて「同じ」と「別」の値が接近・重複が比較的多いため，ChatGPTの方が人間の感覚により近いという結果である.

研究課題1-2の生成AIによる類似度評価については，コサイン類似度に基づく結果よりも高い評価を得る傾向があり，人間の感覚により近いと考えられる．特に,「同じ」とされたケースにおいてChatGPT系モデルの評価は比較的安定していた.

研究課題1の類似度の数値化については全てのケースでAUCにおいて有意とされる値（0.79以上）を達成しており,文章の数値化により人間の感覚に近い状態で類似度の妥当な判断をサポートできると考えた.但し，100%正確な同義分判定を行える閾値は存在しない為，閾値を用いた同義文判定を行う場合は、機械的に同義と判定するか，人間が参考値として利用するかは対象業務の性質に合わせて判断を推奨する.

研究課題2の長文になった場合の検出性能については,長文化すると全体的に類似度が高いと判定されやすく,Embedding系モデルの方が影響を受けにくい結果ではあったものの,「別」における検出性能が劣化することが確認できた.実務で類似度を測る際は,対象を長文で捉えず,短文での評価をすることが推奨されると考える.

4.4 妥当性への脅威

本研究には以下の妥当性への脅威が考えられる

(1)実験データの偏り

本研究のデータは情報技術関連に限定されており，数量も限られている．社内規定や法令など専門性が著しく高いものについては，正確な判定ができない可能性がある.

(2)評価者の主観的影響

ラベリング作業は人間の主観に依存しており，レビュープロセスを通じて精度を確保しているが，完全な客観性は保証されない．評価者間の一致度を定量的に測るための基準やルールが求められる.

(3)長文化データの傾向の違い
研究課題 2 は長文化するため，長文化前の短文が研究課題 1 でしか利用しないデータに比較すると，非常に短い文章となっている．研究課題 1 では研究課題 2 で利用した文章も含め評価したため，研究課題 1 と 2 の間でデータの傾向が異なる可能性がある．
(4)生成 AI 評価は都度結果が変わる可能性がある
はじめにで述べた通り生成 AI のアウトプットは都度変動する可能性がある．

5. 終わりに

5.1 まとめ

本研究では，生成 AI の出力を効率的に評価するための手法として，既に広く活用されているコサイン類似度や生成 AI を用いたテキスト類似度評価の一部について実験を行い，その妥当性や有効性を検証した．実験の結果，生成 AI を用いた類似度評価は，ChatGPT 独立類似度で「同じ」の判定に不安定さがみられるものの，人間の直感的理解をサポートしつつ，AUC の精度において良好な予測力があるとされる値が示された．一方で，長文化したデータでは類似度が上昇しやすい傾向が確認され，評価精度向上のためのさらなる工夫が必要であることもわかった．

業務での活用に向けては該当業務の重要性を鑑みて本研究の結果を利用する事が望ましい．参考意見として扱うか，テストの自動化まで行うかはその業務の重要性に応じた判断が必要である．本研究では生成 AI を用いた類似度評価の成績が良かったが，生成 AI の不得意な文脈においては精度が変わる事が予想される．不得意な文脈の一例としては生成 AI のトレーニングデータに含まれない企業独自の名詞や動詞を用いた文章が挙げられる．その様な精度が変動する様な利用事例で無いか，文章類似度評価が人間の感覚と一致するか ROC カーブを活用した精度の確認を踏まえて利用する事を推奨する．

5.2 今後の展望

研究での検証結果を受け，類似度評価をさらに伸展させるため，社内規定や法令などの専門性の高い分野で特定情報のみ与えた LLM に対しての精度の検証や，逆に汎用性を向上させるための検証を行う．また，類似度評価の精度向上を目指し，長文化したデータへの手法検討や，同義文判定に向くデータ傾向を検証する．生成 AI による評価は都度評価結果が変動する可能性があるため対応を検討し，プロンプトの改良にも取り組む．

参考文献

[1] a t，テキスト生成の自動評価指標について，https://qiita.com/amtsyh/items/a926b79b90dfabe895e9，2020
[2-1]東京海上日動火災保険株式会社，複雑性の高い保険領域に特化した照会応答システム「AI Search Pro」を共同開発，https://www.tokiomarine-nichido.co.jp/company/release/pdf/241129_01.pdf，2024
[2-2] TIS 株式会社，TIS、クラウド型経費精算システム「Spendia」に生成 AI の高度な解析技術を活用した新機能を追加，https://www.tis.co.jp/news/2024/tis_news/20241128_1.html,2024
[2-3] AGEST 株式会社，AGEST、新サービス AI テストツール「TFACT」導入開始～QA プロセスのデファクトスタンダード確立へ～，https://agest.co.jp/news/2024-11-07/,2024
[3]AI プロダクト品質保証コンソーシアム，AI プロダクト品質保証ガイドライン 2024.04 版,https://github.com/qa4ai/Guidelines/blob/main/QA4AI_Guideline.202404.pdf,2024.
[4]OpenAI，OpenAI Embeddings API Reference，https://platform.openai.com/docs/api-reference/embeddings，2024.
[5]OpenAI，新しい埋め込みモデルと API の更新，https://openai.com/ja-JP/index/new-embedding-models-and-api-updates/，2024.