研究コース 5（D2E-AI）

文書間差分抽出における生成 AI 活用の考察

研 究 員　　：益子なるみ（個人参加）
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
ソフトウェア開発における開発プロセス全体への本格的な生成 AI 活用が始まっている．本稿では，文書レビューに注目し，自然言語で記述された複数文書間の差異抽出に焦点を絞り有用性評価を試みた．生成 AI サービスを用いて 2 つの文書が適切に評価され差異が出力されることを検証し，生成 AI で複数文書間の差分比較が可能となるかを検証する．

1.　はじめに

近年のシステム構築ではアジリティが求められる反面，個人情報や換金性の高い情報を取り扱うシステムでは高度なセキュリティ規定・ルールの準拠をより厳密に求められることが多い．スピード感を持ったサービス提供が求められる一方でセキュリティや内部統制などのルールはより厳しく複雑になっていく．そこで各種成果物に対して規定文書の適用についてチェックを行う際に，生成 AI を活用できればより効率的に行えるのではないかと考えた．

生成 AI 活用のサービスや研究はレビュー分野においてもさまざまにおこなわれているが，ファインチューニングや Retrieval-Augmented Generation (RAG)利用が提示されるなど本格的なものもがある．今回の実験では大量の文書検索は目的としておらず，また通常の生成 AI の性能向上でインプットや応答テキスト量も増加したこともあり，プロンプトに直接文書を添付する複数文書間のレビューの活用を探っていくことにした．

2.　課題と検証の焦点

2．1　レビューに関する課題

開発プロセスにおける各種成果物において様々な観点からレビューを行うことは品質の向上の上で必須である．レビューは，工数の確保，レビューアのレベルによる指摘事項のばらつき，見逃しや漏れによる手戻り，レビュー可能な人材を育てるにも時間がかかるなど課題[1]があった．それらの課題から，レビューの品質向上，スキル依存の解消など生成 AI を活用した先行研究[2]などなされてきた．またレビューの効率化や品質向上のため様々なツールが自作や製品として提供されており，文書フォーマットの工夫やチェックリスト，トレーサビリティの維持や用語をシステム的にチェックするなど様々なものがある．ただそれらのツールは高価で，維持メンテナンスに手間がかかり有効ではあったが持続的な定着が難しかった．

上記に挙げたレビュー課題は多岐にわたるが，本稿ではレビューの準備工数の確保に貢献することに絞った．具体的には差分内容を理解する作業に，既存のツールやサービスの代替として生成 AI を活用は有効であるか検証を行う．既存の差分ツールでは変更前と変更後の差異がある部分が明示されるが，変更意図や文書全体への影響まではわからない．高レベルのレビューアであれば経験として単純な比較のみから良し悪しを判断できるがそれでも文書を読み込む必要がある．生成 AI を活用することで判断のもととなる情報を素早く得られれば準備として活用できると考える．

2．2　生成 AI に関する課題

生成 AI の課題としては，まず出力結果の精度がある．性能向上著しいモデルではプロンプトを複雑にしなくても高度な回答が出力されるが,それが質問の意図に沿っているか,正確な情報か，常に同一の結果が出力されるか等が保証されるわけではない．本来品質担保するはずのレビューで生成 AI の回答誤りに気が付かずそのまま受け入れるとレビューの漏れと同等の問題が発生する．レビューは設計やコーディングなどの成果物作成作業とは異なり生成 AI の回答結果を成果物としてそのまま取込む可能性は低いものの，誤りをそのまま信用し成果物の誤りやチェックのすり抜けに気が付かないままでは品質に問題が発生する．すでに様々なメディア等で有識者が語っているように[3]，生成 AI で出力される結果はハルシネーションが混在している前提で人間が結果の採用是非を決める必要があり，レビューアのレベルを把握しておく必要がある．

2．3　検証の焦点

本稿では，生成 AI を活用した文書レビューの事前準備として，差分比較の精度の課題に焦点を当てる．プロンプトにレビュー観点や比較指示を行い得られた出力結果をレビューとして活用できることを検証する．

相互参照関係のある複数文書間，例えば上位文書と下位文書のような親子関係のある文書の検証を目指しているが，まずは単純な差異の抽出を確認する．

理由としては上位文書と下位文書の関係は記述内容が上位文書は抽象化，下位文書は詳細・具体化的になっている場合が多く，単純に同一であるという解釈が難しい．例えば上位文書では「多要素認証を行う」といった記載があった場合，下位文書では「電話番号で本人確認を行う」などその具体的な認証手法が記述されていた場合，一般的なアーキテクチャとしては有効であってもこれを単純に比較し正解とは出来ない．

実際の開発の場面では，図 1 で示すように文書間の行間を埋める他の文書や情報が存在するはずである．

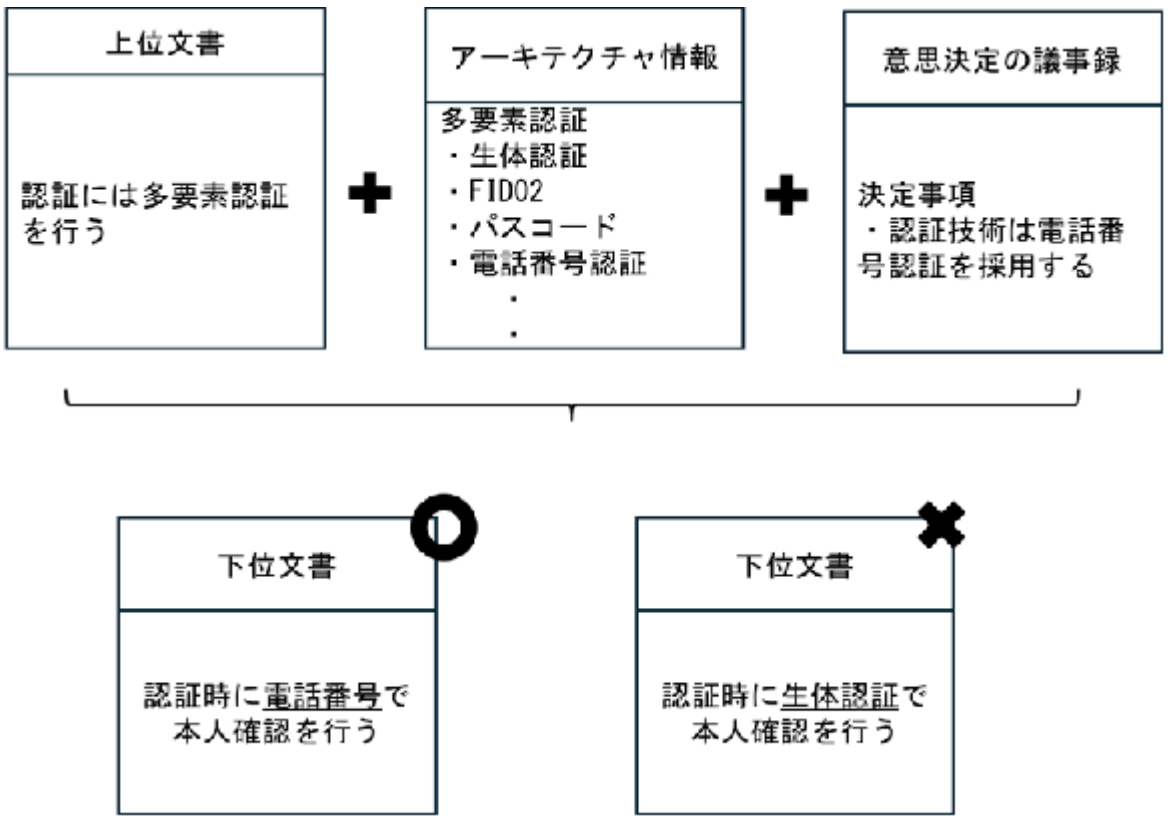

図 1　上位・下位文書間の親子関係イメージ

本稿ではこれらのすべての文書を最初からまとめて比較すると相互参照関係が複雑になるため二文書（以降文書①，文書②とする）を扱うこととした．文書①の内容を決定し，文書②は文書①の文章に特定の変更を加えていった．加える変更は意味上の同一性，意味上のずれがあるものを用意し，変更量は少ないものから段階的に増やす．これらに対し生成 AI に同一性を判断させる検証を進めていくこととした．表１に文書①から文書②に対しどのような変更を加えるか例を示す．

表 1 文章の変更イメージ

	文書①	文書②
意味上の同一性がある変更	レビューは下記を行う ・ピアレビュー ・工程レビュー	レビューは下記を行う ・工程レビュー ・ピアレビュー
意味上のずれがある変更	レビューは成果物の確定前に行う	レビューは成果物の確定後に行う

3. 検証方法

3．1 検証前の事前性能確認

検証に使用した生成 AI は Microsoft 社が提供している Azure OpenAI Service の「チャット プレイグラウンド」を利用した．利用目的は検証環境として準備できるものとしてモデルの指定とパラメータの選択を明示的にできるためである．

本事前検証では文書②に変更を加えない状態で「変更なし」と出力されるかどうか確認した． 結果「差異はない」「同一といえる」といった表現で出力されることは確認できた．ただし，文書②が「同一といってよいがより～の工夫がされているかもしれない」といった「よりよくなっているはず」といった推論をしたと思われる結果も出力される．あくまでリコメンド的な出力であり決定的なハルシネーションではないが，本稿では文書間の比較をどのように取られるか確認する意図もあり，そのまま結果として出力し，厳密さを求めるプロンプトは本検証では取り上げないこととした．

また今回利用したチャット プレイグラウンド」ではプロンプトに文書①および文書②を含めて使用した．試行ごとの出力のぶれを最小限にするため，パラメータ（temperature，TopP）は表１の内容とし以降の検証を進めていくこととした．

表 2 Azure OpenAI パラメータ

パラメータ名	設定値
モデル	gpt-4o-mini (version:2024-07-18)
過去の応答メッセージ	20
最大応答	16384
temperature	0
TopP	0

なお，本検証では同一条件下で回答精度を確認するため，プロンプトを入力し，出力結果を得られたらプロンプトをクリアして，新しいチャットの状態としてから次の検証を実行した．一般的に生成 AI を使用する際は，対話しながら情報を引き出していくことが良いとされているが，本検証では出力結果を引用の必要がある場合，例えば一度抽出してから回答の深堀をする場合のみでこれらも固定し探索的なプロンプトを省いている．

3．2 利用資材

本稿では一般公開されている「政府 CIO ポータル」[4]の『デジタル・ガバメント推進標

準ガイドライン』より一部（第２章　プロジェクトの管理）を抜粋しテキスト化して利用した．インターネット公開されている文書のため学習済文書の可能性もあり未知の文書となるよう一部の用語を置き変えて使用している．

また使用したプロンプトのフォーマットを示す(図 2)．本稿では二文書間の比較や変更箇所抽出に注力し，レビュー有用性を検証するため，出力フォーマットの形式には特にこだわらないこととした．（任意）の個所は出力したい差異に合わせ追加変更する．

```
ドキュメントのレビューをしています．以下の２つの資料の確認項目について抽出して比較してください．
## 資料
『文書①の資料名』
『文書②の資料名』

##　確認項目(任意)
・ターゲットとする節名

##　比較項目(任意)
・用語の変更/文章の流れ

### 文書①の資料名
～省略

### 文書②の資料名
～省略
```

図 2 プロンプト

3．3　文書レビューの観点

文書レビューにおける観点を表 3 に挙げておく．本検証では 2.3 検証の焦点で挙げた，No．3 一貫性のうち用語の部分と相違点を採用することにした．

表 3 文書レビューの観点

No.	観点	説明	検証対象
1	網羅性	必要な項目やセクションの網羅	
2	正確性	記載情報やデータが正確	
3	一貫性　整合性	ドキュメント内やドキュメント間での矛盾,用語の一貫性	○
4	相違点	追加・削除された情報や変化点，情報の過不足	○
5	適合性	想定している文書の利用対象者に適しているか	
6	形式・書式	文法見出し等スタイルの適切さ，記載粒度・表現	
7	視覚的要素	フォーマットや図表の形式や色遣い	
8	参照関係	ドキュメント間の参照や依存関係の正しさ	

3．4　検証観点と変更内容

本検証では文書②で加える変更文言には意味上の同一性やずれがあるものを含めておき，粒度や変更量，変更の位置を変更し生成 AI の判断に影響があるか検証した．検証した

パターンは表 4 に示す.

検証の粒度を１文と文書，テキスト全体構成に分けているのは１文の変更と１段落複数文で構成されている場合に抽出性能に差異が出るかを確認するためである.

表 4 検証観点と変更内容

No.	粒度	変更量	位置	変更内容	
1	箇条書き（１文）	１文字	文頭	総合～	結合～
			文中	～入る前に～	～入る後に～
			文末	～終了前	～終了後
2		複数個所・文字	上記の混在		
3	文章（段落）	１単語	前半	自己点検	内部監査
			中間	レビューポイント	レビュー観点
			後半	品質管理	品質保証
4		複数個所・文字	上記の混在		
5	テキスト全体構成	箇条書き順序変更	中間	箇条書きソート変更 〈変更イメージ〉 a)生成 AI Generative AI ｂ)機械学習 Machine Learning ↓ a)機械学習 Machine Learning b)生成 AI Generative AI	
6		章立ての変更	中間	２節と３節を入替え	
7		箇条書き追加	中間	箇条書きの行を追加	
8		箇条書き削除	中間	箇条書きの行を削除	
9		箇条書き変更	中間	箇条書きの説明文を入れ替え 〈変更イメージ〉 a)生成 AI Generative AI ｂ)機械学習 Machine Learning ↓ a)生成 AI Machine Learning ｂ)機械学習 Generative AI	

4.　検証結果と考察

4．1　検証結果

検証した差分内容と確認結果を表 5 に示す．３回実行し１回でも文書②に加えた変更を判断できた場合に「〇」としている．また複数個所変更を同時に含めた場合はすべての変更が出力された場合に「〇」とした.

表 5 の「プロンプト指定」は図 2 プロンプトにある「確認項目」,「比較項目」の指定に

該当する.

「確認項目」について無指定の場合は文書全体を比較対象とすることになるが，本稿では使用した文書サイズが 24KB 程度ということもあり「確認項目」を指定しなくても用語の変更箇所は出力される場合があった．ただし No.4 では「確認項目」を指定しない場合では「ほぼ同一」といった結果の出力となった.「テキスト全体構成」の No.6 では文章の構造を変更するなど変更箇所が複数の節にまたがっている場合は「確認項目」に対象の節を複数指定するのではなく無指定とする必要があった.「確認項目」を指定した場合は「流れはほぼ同一」といった出力結果になった.

プロンプト「比較項目」は無指定では変更判断されない場合や変更の判断内容を明確に絞りたい場合に追加した．No．9 の削除した[項目]は対象の節タイトルを[項目]に置き換えて検証した.

表 5 検証結果

No.	粒度	変更量	プロンプト指定		変更の位置と結果		
			確認項目	比較項目	先頭	中間	後半
1	箇条書き（１文）	１文字	指定あり	なし	○	×	○
2		複数個所・文字	指定あり	なし	○		
3	文章(段落)	１単語	指定あり	用語変更	○	○	○
4		複数個所・文字	指定あり	用語変更	×		
			指定なし	なし	×		
5	テキスト全体構成	箇条書き要素順序変更	指定あり	なし	○		
6		章立ての変更	指定なし	文章の流れ	○		
7		箇条書き要素追加	指定あり	なし	○		
8		箇条書き要素削除	指定あり	削除した[項目]	○		
9		箇条書き要素変更	指定あり	なし	○		

4．2　検証結果の考察

4．2．1　文字・単語の変更についての考察

文書②に加えた変更が１文字変更の場合，文頭・文末は抽出されるが，文中を変更とした場合判断できなかった．変更が単語の場合は位置に関係なく変更を判断したが，節のタイトルに含まれている単語や主語などに使用されている単語は，明確に変更と判断される傾向があると思われた．プロンプトの工夫や重要な用語に対しては認識されやすい文書の記述方法などの検討の余地がある.

複数個所の変更では節をまたがるなど分散されて変更箇所がある場合，何らかの「用語の違いがある」とあいまいな変更判断はされるものの，具体的な変更箇所の結果が出力されない．No.4 の１回目の応答結果に対して，プロンプトに追加で「用語や表現の違いについて厳密に出力してください」とすると具体的な変更点について出力されるようになったが，複数個所ある変更のうち最初の変更箇所しか出力されなかった．レビュー対象文書に複数変更があるのは通常のことであるため，追加のプロンプト指示などで残りの変更判断が可能となるかさらにプロンプトは工夫を検討する必要がある.

4．2．2　テキスト全体の変更についての考察

箇条書き形式の並びの変更は特に比較項目を指定しなくても「順序が異なる」と変更があると判断された．節の入れ替えについても，「比較項目」を「文章の流れ」と指定すると対象の節の順序がかわっていることが判断されるようになった．4.2.1 のような少ない変更量よりも変更量は多くても文章の塊として扱えるほうが生成 AI としては変更が判断しやすいのかもしれない．

箇条書きの要素追加については，プロンプトに「比較項目」がなくても問題なく変更ありと出力されたが，要素削除はプロンプトに「比較項目」を明確に指定する必要があった．要素削除の場合は，残りの要素の文言に変更がないはずの箇所も，要素の内容変更有と判断されることもあった．削除の場合は文章のブロックのずれが，内容が異なると判断されるのかもしれない．

また要素の追加・削除の場合は，文書①文書②においてそれぞれ一方にのみ要素の記載がある状態になるが，文書②で要素の削除がある場合は，文書①で「追加されている項目」という表現で出力されることもある．プロンプトの情報不足の可能性はあるが，常に文書②が変更された文書と判断されるわけではなく，結果の読み取り方も工夫が必要であると思われる．

4．2．3　差分抽出における生成 AI 活用についての考察

差分抽出の変更の判断を生成 AI で行うメリットとしては，抽出した変更箇所に対する影響範囲の考察が加えられていることがあげられる．例えば「品質管理」を「品質保証」という用語を置き換えた実験では，結論のサマリとして「特に『品質管理』と『品質保証』の用語の違いは，プロジェクトの品質に対するアプローチの違いの変化を示唆している可能性があります」といった文言が補足されることがある．単純な比較ではない気付きを得られる可能性がある．

単純な差分ツールでは変更前と変更後の差異がある部分が抜き出されて示されるが，変更があった内容の意味に言及したサマリコメントは，この変更が正しいものか，より深く人間が考察するためのアシスト情報としては利用できるのではないかと考える．例えば文書①文書②とでは「レビューのタイミングが異なる」「プロジェクトの管理手法や運用に影響を与える可能性がある」といったアドバイスが出力される．反対に単純な変更を網羅して確認を求めるならば，当然のことながら従来の差分ツールを利用するほうが確実である．本稿では生成 AI で差分比較したが，比較そのものは差分ツールで行い，生成 AI はその出力結果をインプットとして要約し，文書の品質の傾向を提示する方法もあると思われる．

5.　まとめ

5.1 結論と今後の課題

本検証では既存の差分ツールをそのまま置き換えるような利用方法はできないものの，検証範囲を指定して二文書間の差異を出力しその比較内容のサマリを出力する期待が持てることは確認できた．今後は下記のような文書間の比較をさらに検証しより有効な活用方法を探っていきたい．

- 同一の意味で異なる表現の同一性・ずれの確認（用語の揺れを含む）
- 形容詞，動詞，形容動詞等の類似語の違いによる推論された意味の齟齬
- 抽象度の異なる記述の同一性・相違点の抽出．特にソフトウェア開発における要件定義・機能設計・詳細設計・テスト設計などの開発設計文書やマニュアル・手順書などの保守運用文書間の比較
- 時系列やプロセスフローを伴う矛盾点の抽出
- 三つ以上の文書間比較や形式・フォーマットの異なる文書間差異の吸収
- 意図的に変更している個所の比較対象としての除外

また生成 AI や生成 AI を組み込んだシステムを用いることで，人間が行う作業にかかる工数やコスト削減を期待されているが，反対に生成 AI を用いることで新たにコストが発生する．成果物への脆弱性の混入・著作権侵害などの法律上の問題への対応，生成 AI の実行時間と結果の妥当性の証明にかかる時間，モデル品質を維持するための学習コスト，こうった課題も含めた費用対効果を確認してく必要がある．

5.2 今後の展望

レビューなど品質に関連した研究や商用サービスもますます増えてくると思われるが，本格的に業務として活用されてくると出力結果に対する生成 AI の決定プロセスの透明性や説明責任の必要性がますます増してくると思われる．著作権やファクトチェックを行うサービスなども提供されてきておりそのような動向にも注目していきたい．

参考文献

[1] 森崎 修司，間違いだらけの設計レビュー 第３版，日経 BP，2023
[2]北里 竜，片桐 汐駿，馬場 大輔，星野 智彦，ソフトウェアレビューにおける生成 AI 活用の研究～ChatGPT が欠陥検出と指摘伝達をアシスト～ソフトウェアレビューにおける生成 AI 活用の研究 ～ChatGPT が欠陥検出と指摘伝達をアシスト～，ソフトウェア品質シンポジウム，2034
[3] 和田 卓人，やっとむ，森崎 修司，生成 AI の得意と不得意を知って開発の仕事に役立てよう～最新研究事例から見えてきたこと～，Developers Summit 2024 Summer，24-A-9，2024
[4] 政府 CIO ポータル，https://cio. go. jp/