L-022

ポストモーテム作成を支援するインシデント振り返り自動生成方式の検討
Study of an automatic incident look-back generation method to support postmortem creation

和田 清美† 増田 峰義†
Kiyomi Wada Mineyoshi Masuda

1. はじめに

ポストモーテムとはシステム障害対応の振り返りと再発防止策を記録するために書かれるドキュメントである。ポストモーテムを作成するためには、インシデントチケットの他に関係各署による原因究明、対処の判断と実行結果から、ポストモーテムに必要な項目に関する情報を取得して作成する。ポストモーテム作成には時間とスキルが必要であるが、技術者不足のため工数削減とスキル不要化が課題である。そこで、ポストモーテム作成のインシデント振り返り部分を生成 AI で支援する方式を提案する。これにより、ポストモーテム作成工数を削減できるため、インシデント対応後すぐに再発防止策の検討を開始することで、システムの安定運用に貢献できる。

2. ユースケースと課題

2.1 インシデント管理での生成 AI 活用ユースケース

始めに、インシデント対応で使用されるチケット管理システムについて説明する。チケット管理システムは、IT サービス運用業務のインシデント対応の他、ユーザからの問合せ対応、システムのリリース・保守作業などをチケット情報として一元管理し、IT サービスに関する情報共有と連携を実現する。また、チケット管理システムにはシステム運用業務に関する様々な種類のチケットが日々蓄積されていくため、これらのチケットを活用してポストモーテムを作成することができる。そこで、チケット管理に生成 AI アプリを組み込んだシステムを提案する。

図 1 はインシデント対応のためのチケット管理に生成 AI アプリを組み込んだシステム概要である。図 1 において、生成 AI アプリは大規模言語モデル Large language Models(LLM)を活用して、障害発生により作成されたインシデントチケットからポストモーテムのインシデント振り返り(以下ポストモーテムとする)を生成する。

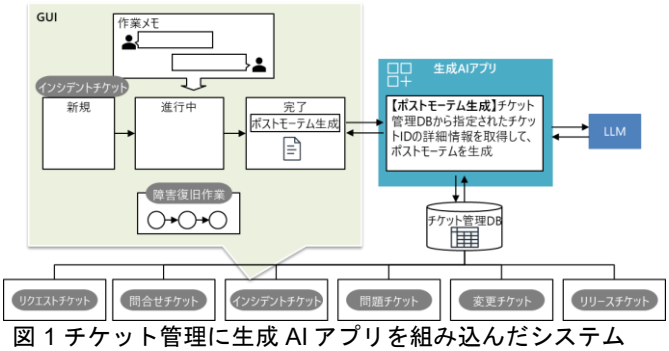

図 1 チケット管理に生成 AI アプリを組み込んだシステム

2.2 課題

従来技術[1]の LLM によるインシデントの影響範囲を評価し要約を作成するシステム「OASIS」は、クラウドシステムに障害が発生すると、顧客への迅速な通知、問題の軽減、障害の解決のため、①障害の影響範囲を自動的に評価し②人間が読める要約を作成する。関連インシデントから障害の全容を把握するため、インシデント間の関係(相関)を自動的に見つけてそれらを集約して要約を作成する。

「OASIS」で扱うインシデントは定型化されたイベント情報である。一方、図 1 のインシデント対応時の作業メモは試行錯誤のプロセスであり、障害の原因や復旧につながる情報と不要な情報が混在している。このため、インシデント対応状況を正確に把握し、ポストモーテム生成の元となる情報を含む要約を生成しなければならない。

また、図 1 で 1 つのインシデントチケットだけでポストモーテムに必要な情報を全て持っていない場合、ポストモーテムに必要な情報を持つ関連チケットを検出して、この情報を含めてポストモーテムを作成する。関連付けはワークフローから作成された場合を除き、人が手動で関連付けをしなければならないため手間がかかる。ポストモーテムに必要な情報を持つ関連チケットを検索するとき、全文検索でキーワードの一致度で関連性ありと判定されたチケットが、ポストモーテム作成に必要な情報を持つチケットでない可能性がある。ポストモーテム作成に不要な情報を含めると、ポストモーテムのドラフト生成の精度が低下する可能性がある。

以上より課題は、チケット内の紆余曲折による不要な情報を無視することと、ポストモーテムに関連するチケットを正しく選別することである。

3. 提案方式

図 2 はポストモーテム生成の課題を解決する提案方式である。提案方式は、チケット情報と作業メモ要約と Site Reliability Engineering(SRE)[2]推奨のポストモーテム必須項目(a)を生成 AI のテキストに特化した LLM に与えて、ポストモーテムのドラフトを生成する。また、ポストモーテムの不足項目に対して検索対象チケット種別を指定して関連チケットを検索するために、検索条件にポストモーテム項目別検索対象チケット (b)を指定する。例えば、ユーザ影響が不明の場合は検索対象チケット種別は問合せチケットとする。

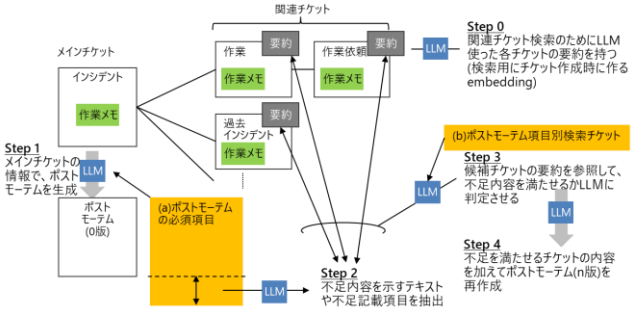

図 2 ポストモーテム生成方法

†株式会社 日立製作所, Hitachi, Ltd.

1 つめの課題であるチケット内の不要な情報を無視することは、本提案方式の作業メモの要約とポストモーテム必須項目(a)を指定することで実現できる。作業メモの要約は関係者間のやりとりで不要な情報を除去でき、ポストモーテム必須項目(a)を指定することで、LLM は必要な情報を選択して作文するからである。

2 つめの課題であるポストモーテムに関連するチケットを正しく選別することは、本提案方式のポストモーテム項目別検索対象チケット(b)がポストモーテム不足項目を補完するチケットを絞り込み、意味検索することで実現できる。LLM は検索結果のチケットの要約が、ポストモーテム不足項目を満たすかどうかを意味内容で判断するからである。

ポストモーテム生成手順を以下に示す。前提として関連チケットの意味検索をするため、Step0 で登録済みチケットを要約してベクトル化し、検索用 DB に格納する。

Step 1 でメインチケットから、作業メモを要約し、ポストモーテム必須項目(a)に従ってポストモーテムを生成する。

Step 2 でポストモーテム必須項目(a)を満たさない項目を抽出する。

Step3 で前記 Step2 で抽出した不足項目に対応する検索条件(b)で関連チケットを検索し、検索結果のチケット情報の要約がポストモーテム必須項目(a)を満たすかどうかをベクトル類似度で判定する。

Step 4 で追加のチケット情報を含めてポストモーテムを再作成する。

4. 実験方法

作業メモの要約とポストモーテム必須項目(a)を指定することで、1 つめの課題であるチケット内の不要な情報を無視できることを検証する。

図 3 は AWS 環境に構築した検証システム構成である。チャットに入力されたプロンプトを生成 AI アプリ(Agent)に渡し、チケット管理プログラムにアクセスして LLM(基盤モデル)が要約した結果を回答する。Amazon Bedrock の基盤モデル[3]は Anthropic 社の Claude 2.1 を使用する。Claude 2.1 は多言語対応で日本語の応答精度が高く、最大トークン数が 200k で、要約や Q&A、コンテンツ作成や推論に優れている。Agent には、エージェント向けの指示と API を呼び出してタスクを実行するための Action Group がある。Action Group にはチケット管理プログラムの Application Programming Interface(API)を呼び出す関数と API を定義した OpenAPI スキーマ[4]がある。

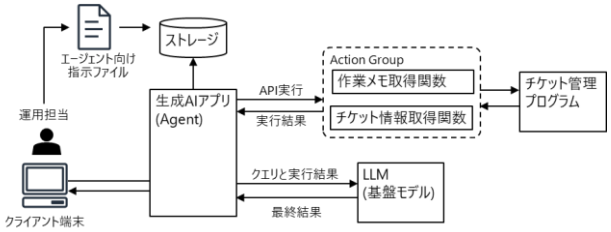

図 3 検証システム構成

エージェントの処理は、チケット管理プログラムのチケットの作業メモの要約、ポストモーテム生成の 2 種類とした。エージェント向け指示ファイルには、ポストモーテム生成の必須項目と内容を定義した。

検証データは Gitlab の Issues「2017-06-15: Pages Outage Post Mortem」[5]を使用した。Gitlab Pages のデプロイ後に該当ページが表示されない障害発生から原因調査と復旧対策までのタイムライン情報が含まれており、これに基づいて、作業メモを含むチケット情報を作成した 。

実験方法はチケット情報からポストモーテムを生成する際、ポストモーテム項目指定および作業メモ要約の有無で生成されるポストモーテムの内容を比較する。

5. 結果と考察

図 4 は、①ポストモーテム項目指定なし、②ポストモーテム項目指定ありで作業メモ要約なし、③ポストモーテム項目指定ありで作業メモ要約ありに対するポストモーテム生成結果である。①は黄色の部分がインシデントに関係ない情報であった。また、作業メモに沿ってインシデント対応の流れが説明されており、根本原因や発生要因を読み取る必要があった。②は黄色の部分が作業メモの紆余曲折のやりとりで不要な情報であった。③は要点がまとまっており、影響、根本原因、発生要因、対処が正確であった。

以上より、作業メモの要約とポストモーテム必須項目を指定することで、不要な情報を無視し、ポストモーテムに必要な情報を適切に選んで生成できることを検証できた。

①ポストモーテム項目指定なし 作業メモ要約なし	②ポストモーテム項目指定あり 作業メモを要約なし	③ポストモーテム項目指定あり 作業メモを要約あり
rsyncを使用して残りのページデータの移行を開始しましたが、ダウンタイムを最小限に抑えるためにrsyncの完了を待たずにカットオーバーを実行しました。その後、fleet全体でマウントポイントを更新しましたが、gitlab-pagesデーモンが再起動されていなかったため、ページはまだ古いNFSサーバーから提供され続けていました。fleet全体でgitlab-pagesデーモンを再起動した後、問題は解決しました。今後同様の事態を避けるために、マウントポイントの変更後には関連サービスの再起動を確実に行う必要があります。	【概要】 CIジョブでデプロイステージが成功した後もGitLab Pagesが更新されない問題が発生。rsyncによるデータ移行、マウントポイントの変更などを実施したが、ページが更新されない状態が続いた。gitlab-pagesデーモンを再起動することで問題が解決した。	【概要】 CIジョブ成功後もGitLab Pagesが更新されない問題が発生しました。rsyncとgitlab-pagesデーモンの再起動により解決しました。
	【影響】 ユーザに古いコンテンツが表示され続ける状態が発生	【影響】 ユーザに古いページが表示され続ける影響がありました。
	【根本原因】 gitlab-pagesデーモンがマウントポイント変更後に再起動されていなかった	【根本原因】 gitlab-pagesデーモンがマウントポイント変更後に再起動されなかったことが原因です
	【発生要因】 マウントポイント変更時の再起動漏れ	【発生要因】 マウントポイント変更作業後のgitlab-pagesデーモン再起動が漏れたことがきっかけです。
	【対処】 gitlab-pagesデーモンの再起動	【対処】 gitlab-pagesデーモンをfleet全体で再起動しました。

図 4 ポストモーテム生成結果

6. おわりに

本報告では、チケット管理に生成 AI アプリを組み込んだシステムを構築し、インシデントチケットと作業メモからポストモーテム作成 のインシデント振り返り部分を生成 AI で支援できることを確認した。

商標について

Amazon Web Services は米国およびその他の国における Amazon Technologies, Inc.の登録商標である。

参考文献

[1] P Jin, S Zhang, M Ma, H Li, Y Kang, L Li, Y Liu, B Qiao, C Zhang, P Zhao, S He, F Sarro, Y Dang, Q Lin, “Assess and Summarize: Improve Outage Understanding with Large Language Models”, arXiv:2305.18084v1 [cs.SE] 29 May 2023
[2] B Beyer, C Jones, J Petoff, N Murphy, Site Reliability Engineering, O’Reilly
[3] https://docs.aws.amazon.com/ja_jp/bedrock/latest/userguide/models-supported.html
[4] https://www.openapis.org/
[5] https://gitlab.com/gitlab-com/gl-infra/production-engineering/-/issues/2040