クエリ要約における 疑似ハルシネーションの作成と検出

横山 正秋[1]　嶋田 和孝[2]

概要：クエリ (特定の話題や内容に関する質問) に対応する要約を作成するタスクをクエリ要約という．近年，大規模言語モデルの発達により，クエリ要約の生成は容易になっているが，ハルシネーションを含む．モデルの出力するハルシネーションは誤った理解の原因となるため，検出し取り除くことが求められる．要約対象文書において言及のないトピックのクエリに対し，クエリ要約を作成した場合は必ず不正確なものとなる．このような言及の有無に基づいたハルシネーションデータは存在しない．そのため，要約対象文書に言及のないトピックのクエリ要約を疑似ハルシネーションとしたデータを作成する．さらに，作成したデータに対して検出実験を行い，本データに対する学習の有効性を確認する．

キーワード：クエリ要約，ハルシネーションデータ，ハルシネーション検出

Creation and Detection of Pseudo-Hallucination in Query-forcsed Summarization

Abstract: The task of generating a summary responding to a query is called query-focused summarization. In recent years, the development of Large Language Models has facilitated the generation of query-focused summaries, but they contain hallucinations. The hallucinations generated by models cause misinterpretation, so they must be detected and removed. If a query-focused summary is generated for a query on a topic that is not mentioned in the document being summarized, it will inevitably be inaccurate. There is no such hallucination data based on the existence of such mentions. Therefore, we create pseudo-hallucination data for query-focused summarization of topics that are not mentioned in the document. Furthermore, we will perform detection experiments on the created data to evaluate the effectiveness of learning on this data.

Keywords: Query-Focused Summarization，Hallucination Data，Hallucination Detection

1. はじめに

近年，大規模言語モデルの発達は著しく，様々なタスクにおいて活用が広がっている．要約タスクにおいても，大規模言語モデルの利用によって流暢な要約を容易に作成可能となっている．しかし，大規模言語モデルにはハルシネーション (Hallucination: 幻覚) と呼ばれる，不正確または無意味なテキストを生成する性質がある [1]．要約タスクにおいて，要約対象文書と要約の内容が異なる場合にハルシネーションとなる．Falke ら [2] は要約システムを分析し，生成された要約の 25%に誤りがあることを示している．大規模言語モデルの出力は，ハルシネーションの場合であっても流暢で自然なテキストとなるため，ハルシネーションの検出が困難である．

クエリ要約 (Query-Focused Summarization) はクエリ (特定の話題や内容に関する質問) に対する要約を生成するタスクである．クエリ要約は，要約対象文書と特定の内容や話題に関するクエリを入力として，クエリに対応した内容の要約を生成する．特に議事録など，多岐にわたる話題が含まれ長大である場合にクエリ要約は有効であり，内容の理解が容易になる．しかし，生成されたクエリ要約が要

[1] 九州工業大学 大学院情報工学府
Department of Creative Informatics, Kyushu Institute of Technology 680-4 Kawazu, Iizuka, Fukuoka 820-8502, JAPAN
[2] 九州工業大学 大学院情報工学研究院 知能情報工学研究系
Department of Artificial Intelligence, Kyushu Institute of Technology 680-4 Kawazu, Iizuka, Fukuoka 820-8502, JAPAN

約対象文書の内容と異なり，ハルシネーションとなる場合がある．ハルシネーションは誤った理解に繋がり，特に議事録の場合は意思決定や情報収集に悪影響を及ぼす可能性がある．そのため，クエリ要約のハルシネーションをもれなく検出することが求められる．

クエリ要約のハルシネーション検出モデルの学習やテストにはハルシネーションのデータが必要である．クエリ要約におけるハルシネーションデータは少ない．また既存のデータは大規模言語モデルにハルシネーションの生成を指示するものが多く，実際のハルシネーションと異なる可能性がある．そのため，既存の議論対話コーパスである Kyutech コーパス [3] を対象にハルシネーションを含むクエリ要約をを生成し，データを作成する．Kyutech コーパスには，対話ごとにトピックタグが付与されている．正確なクエリ要約のクエリは付与されているトピックタグのものである．これに対して，言及されていないトピックタグに対するクエリ要約は，コーパスにおいて誤った内容であり，ハルシネーションである．このような Kyutech コーパスのトピックタグの有無を利用することで，ハルシネーション生成の指示をすることなくハルシネーションデータを作成可能である．ただし，トピックタグの有無によってハルシネーションデータを作成しており，実際に内容が正確か不正確かを確認していない．そのため，作成されるデータは疑似ハルシネーションデータとなる．

本研究の全体の流れを図 1 に示す．まず，図 1 の上部に示すように，Kyutech コーパスを対象にクエリ要約における疑似ハルシネーションデータを作成する．クエリをトピックタグから作成，大規模言語モデルによってクエリ要約を生成し，トピックタグの有無によってハルシネーションラベルを付与する．さらに，図 1 の下部に示すように，作成データを対象とした検出モデルを構築し，ハルシネーションを検出する．検出モデル全体を学習データによってファインチューニングする提案モデルと出力層のみを学習する比較モデルとの精度差を検証する．ハルシネーションをもれなく取り除く必要性とハルシネーションと非ハルシネーションの検出のバランスを考慮し評価を行う．

2. 関連研究

2.1 クエリ要約

クエリ要約タスクは Dang ら [4] において提案され，様々な研究が行われている．QMSum[5] はクエリ要約の英語データセットとして一般的に用いられている．QMSum はマルチドメインの会議におけるクエリ要約のデータセットであり，AMI コーパスや ICSI コーパス，委員会会議を対象にクエリ要約を作成している．QMSum におけるクエリの作成には一般的なクエリスキーマリスト (General Query Schema List) と特徴的なクエリスキーマリスト (Specific Query Schema List) を導入している．これらのスキーマに

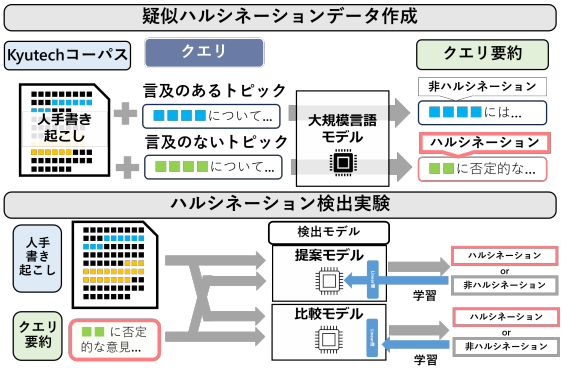

図 1 疑似ハルシネーションデータ作成と検出実験の流れ．青のトピックはコーパスにおいて言及されているが，緑のトピックは言及がないため緑のクエリ要約はハルシネーションである．検出実験では，要約対象文書とクエリ要約がモデルに入力され，ハルシネーションか否かを出力する．検出モデル全体を学習データによってファインチューニングする提案モデルと出力層のみを学習する比較モデルを比較する．

対して，話者やコーパスにアノテーションされたトピックとサブトピックを挿入することでクエリを構成している．また作成されたクエリに対して，人手でクエリ要約を作成している．本研究では，トピックタグが付与されている既存データセットを用いるが，クエリは含まれていない．そのため，QMSum のクエリスキーマリストをクエリテンプレートとして利用し，クエリ作成を行う．

2.2 要約におけるハルシネーション

Manez ら [6] は抽象要約タスクにおけるハルシネーションを定義し，ハルシネーションを内在的 (Intrinsic) と外在的 (Extrinsic) に分類している．要約対象に含まれる内容と明らかに矛盾するクエリ要約の場合，内在的なハルシネーションである．内在的なハルシネーションは，明らかな矛盾を発見することで検出可能である．これに対して，要約対象文書において言及のない内容を含む場合，外在的なハルシネーションである．外在的なハルシネーションの検出には，要約対象文書全体を網羅的に確認する必要がある．文書の網羅的な確認は，文書全体を理解したうえで要約内容への言及の有無判断が必要であり，人手で検出することは困難である．本研究では外在的なハルシネーションに着目し，クエリのトピックが要約対象文書に含まれていない場合を考える．

3 行要約データセット [7] は本文と 3 行の要約文から成り，岩本ら [8] は不正確な要約文を生成し，日本語における不正確な要約文のデータセットを自動構築している．文の反意化，固有名詞の入れ替え，数字の入れ替え操作により不正確な要約文を生成した JFactCC，要約文と最も関連している本文中の 1 文を削除することにより要約文との整合性を破綻させる JSumFC，そして GPT-4 を用いた過大表現による誤り例の生成とその拡張からなる JExIS を提案

している．これらは疑似的なハルシネーションデータといえる．本研究では，ハルシネーションを課題とするにあたり，通常の要約タスクではなくクエリ要約タスクに注目している．

2.3 クエリ要約におけるハルシネーション

HaluEval[9] は，大規模言語モデルのためのハルシネーション評価ベンチマークデータセットを提案している．HaluEval では，QA，対話，要約の各タスクについて ChatGPT に対しハルシネーション生成の指示や人手によるハルシネーションのラベリングを行っている．本研究では，生成におけるハルシネーションの指示を行わないことからより実際に近いハルシネーションを生成できると考えられる．また，人手ではなくクエリの内容への言及の有無からラベリングを行うことで疑似ハルシネーションデータを作成している．

TofuEval[10] は，MediaSum データセットと MeetingBank データセットを対象にハルシネーションを作成している．特に大規模言語モデルによるトピックの特定と，そのトピックを利用した要約の作成を行っている．作成した要約の内容が事実整合性，関連性，完全性に適合しているかを人手によってアノテーションしている．さらに，作成されたデータセットについて，大規模言語モデルによる推定も行っている．本研究では，既存のトピックに対して，含まれていないトピックのクエリ要約も作成することで，誤った要約を作成しラベルを付与している．

3. 疑似ハルシネーションデータの作成

3.1 Kyutech コーパス

データの作成には，Yamamura ら [3] によって作成された Kyutech コーパス[*1]を対象とする．Kyutech コーパスは，大学生及び大学院生 4 名の参加者 (A，B，C，D) による架空のショッピングモールに関する 9 対話のデータセットである．9 対話は 4 つの設定 (設定 1〜4) に対応し，設定 1 のみ 3 対話，設定 2〜4 は 2 対話である．

Kyutech コーパスの対話の例を表 1 に示す．書き起こされた発話に対して，参加者ラベルに加えて発話に対する付加的な最大 3 個のトピックタグが付与されている．全 28 種のトピックタグは各発話に対し主タグ，副タグ 1，副タグ 2 として最大 3 個が付与されている．

3.2 作成手法

疑似ハルシネーションデータの作成手法の流れを図 2 に示す．本節では，まずクエリの作成方法を示し，次に対応したクエリ要約の生成方法，そしてクエリ要約に対するハルシネーションのラベル付与方法について示す．

[*1] http://www.pluto.ai.kyutech.ac.jp/ shimada/resources.html

表 1 Kyutech コーパスの対話の例．
発話に対して発話者とトピックタグが付与されている．

発話者	発話	主タグ	副タグ 1	副タグ 2
C	あと，家族連れなら和食の定食屋がいいと思う	People	Exist3	-
A	家族向けと考えると，もうちょっとリーズナブルな方が	People	Price	-
B	定食もいいけど，値段が高い寿司も家族向けかも	People	Exist3	CandX

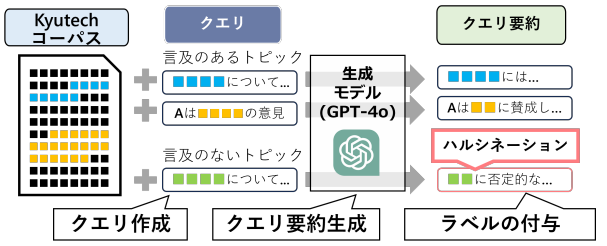

図 2 疑似ハルシネーションデータの作成手法の流れ．Kyutech コーパスを対象に，クエリの作成とクエリ要約生成を行い，ハルシネーションラベルを付与する．

3.2.1 クエリ作成

クエリ作成の流れを図 3 に示す．クエリの作成は，QMSum[5] を参考とし，クエリテンプレートを日本語に翻訳し利用する．クエリの作成においては，クエリテンプレートに対して，トピックや参加者を挿入することでクエリを作成する．QMSum には 18 種類のテンプレートがあるが，要約作成が可能か判別のできないもの[*2]は除外し，9 種類のテンプレートを利用する．使用した 9 つのテンプレートを表 2 に示す．作成したテンプレートに対して，[参加者] には A，B，C，D を挿入する．[トピック] には Kyutech コーパスのうち，対話に設定されたタグを挿入する．[参加者] と [トピック] の複数の組み合わせを持つクエリに対しても同様に挿入する．

3.2.2 クエリ要約の生成

図 4 にクエリ要約作成の流れを示す．クエリ要約の生成には大規模言語モデルの 1 つである，ChatGPT の API を使用する[*3]．ChatGPT は流暢な文章を生成可能であり，プロンプトを用いて生成内容の指示を行うことが可能である．クエリには 3.2.1 節で作成したクエリを使用する．クエリ要約生成には，記号の除去などの前処理を行った対話の人手書き起こしとクエリを入力する．出力では，クエリ

[*2] 「[トピック] について議論する際、グループ/委員会はなぜそのようなことをすることにしたのか？」や「[トピック] について議論するとき、なぜ [参加者] は [サブトピック] のことを考えたのか。」など
[*3] GPT-4o-2024-05-13

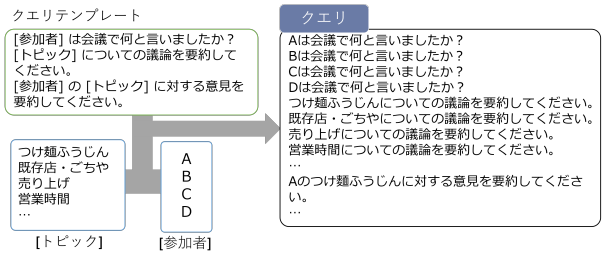

図 3 クエリ作成の流れ．
クエリテンプレートに対してトピックと参加者を挿入する．

表 2 使用したクエリテンプレート．

[参加者] は会議で何と言いましたか？
会議の結論/決定は何でしたか？
会議の目的は何でしたか？
[トピック] についての議論を要約してください。
[参加者] の [トピック] に対する意見を要約してください。
[トピック] に関する利点と欠点は何でしたか？
[参加者] は [トピック] について何を考えましたか？
[トピック] を議論する際、[参加者] は [トピック] について何を考えましたか？
[参加者] と [参加者] は [トピック] について何を議論しましたか？

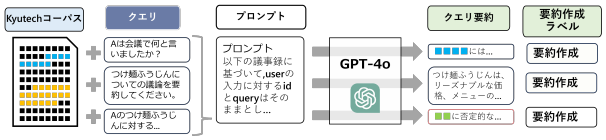

図 4 クエリ要約生成の流れ．人手書き起こしとクエリをプロンプトとして入力する．クエリ要約と要約作成ラベルを出力する．

要約に加えて，そのクエリに対して要約があると判断したか否かのラベルについても同時に出力するように指示する．これはクエリ要約の生成において「要約なし」や「本文中に言及がありません」など多様な出力が含まれると考えられ，要約を作成したものを区別するために要約作成ラベルを出力する．API の使用コストの観点から，1 度のリクエストに 30 件のクエリを入力し，要約と言及ラベルの出力を行う．

プロンプトを表 3 に示す．入力する Kyutech コーパスの人手書き起こし (議事録) とクエリを入力して，クエリ要約と要約作成ラベルを追加して出力するように指示している．

3.2.3 ハルシネーションラベルの付与

図 5 にハルシネーションラベルの付与の流れを示す．3.2.2 節にて作成したクエリ要約のうち，本文に言及がないものを疑似ハルシネーションとしてラベル付けする．ラベルはハルシネーションまたは非ハルシネーションである．ハルシネーションラベルは，クエリに含まれる参加者やトピックタグの内容について，要約対象文書に言及があるか否かから付与される．

3.1 節に示した Kyutech コーパスのトピックタグのうち，

表 3 クエリ要約の生成プロンプト．

{” role”：” system”，” content”：” 以下の議事録に基づいて， user の入力に対する id と query はそのままとし， [クエリ要約](query に対応する要約テキスト)， [要約作成ラベル](要約が存在すれば 1 を， 要約が存在しなければ 0 をもつ整数型) にあたる部分を 生成し,id,query,[クエリ要約],[要約作成ラベル] を持つ 辞書型の リスト形式のままで回答してください”}，
{” role”：” system”，” content”：[要約対象文書]}，
{” role”：” user”，” content”：[30 件のクエリ]}

少なくとも 1 発話にトピックタグの付与されている発話がある場合を考える．このとき，この対話においてトピックの内容への言及があるため，クエリ要約の作成は可能でありラベルは非ハルシネーションとなる．図 5 の例においては，要約対象文書全体が 3 発話であると仮定する．“定食屋についての議論を要約してください。” のクエリについては “候補店・定食” タグが付与されているため言及があり，非ハルシネーションである．参加者の条件のみをもつクエリについても同様に，参加者の発話が少なくとも 1 つ存在する場合は言及があり，ラベルは非ハルシネーションとなる．“A は会議で何と言いましたか？” のクエリは A の発話が存在するため，非ハルシネーションである．しかし，1 発話にもトピックタグが付与されていない場合には，対話内では言及がないと考えられる．付与されていないトピックタグのクエリに対してクエリ要約を作成した場合には，本来存在しない内容を要約していると考えられるため，ハルシネーションとなる．参加者とトピックが条件となるクエリに対しては，特定の参加者の発話に特定のトピックタグが付与されているかを確認し，ラベルを付与している．図 5 の例において，“A の定食屋にに対する議論を要約してください。” のクエリに対応するハルシネーションラベルを考える．このとき，A の発話はコーパスに存在するが，“候補店・定食” のトピックタグの付与された A の発話は存在しない．そのため，該当のクエリについては言及がないとされ，ハルシネーションとしてラベル付けする．これ以外の 3.2.1 節で作成したクエリにおける複数の条件を持つクエリについても同様である．

3.3 作成結果

表 4 に作成した疑似ハルシネーションデータの作成結果を示す．表の行方向はトピックの言及の有無であり，列方向は ChatGPT の出力におけるクエリ要約生成の有無である．トピックタグをもとに作成されたクエリは 5,329 件であった．人手書き起こしと作成したクエリを入力する ChatGPT はクエリ要約の出力が期待されたが，2,177 件のクエリ要約を生成していない．これらのデータはハルシネーションか否かの対象ではないため除外した．クエリ要

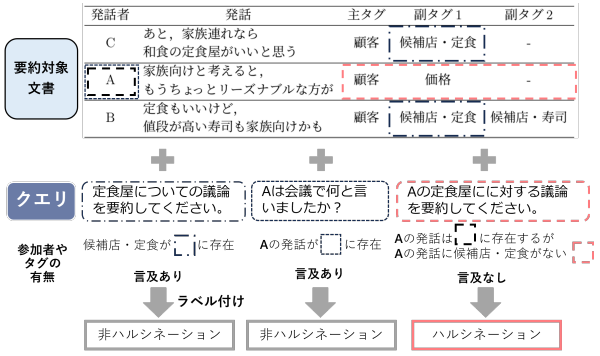

図 5 ハルシネーションラベル付与の流れ．全体が 3 発話と仮定しタグの有無からハルシネーションラベルを付与する例を示す．

表 4 疑似ハルシネーションデータ作成結果．

	クエリ要約作成	クエリ要約作成なし
トピック言及あり	2,491	788
トピック言及なし	**661**	1389
計	3,152	2,177

約を作成した 3,152 件のうち，トピックに言及のある非ハルシネーションは 2,491 件であり，トピックに言及がないハルシネーションは 661 件であった．

表 5 に本手法で作成した疑似ハルシネーションデータの例を示す．要約対象の人手書き起こしの内容について，3.2.1 節で作成したクエリに対応したクエリ要約を作成している．“A の売り上げ” は参加者 A の発話かつ “売り上げ” のトピックタグが付与された発話が存在する．これは言及があるため非ハルシネーションのラベルが付与された．これに対して，“B の既存店・ミスター K” のクエリについて，この参加者とトピックタグに一致する発話が存在しなかった．そのため，言及がないとされハルシネーションのラベルが付与された．表 5 に示す疑似ハルシネーションデータの例では，ハルシネーションであるかに関わらず，クエリ要約において流暢な要約を作成している．そのため，クエリ要約単体によってハルシネーションか否かを判断することは難しいと考えられる．これは，クエリ要約におけるハルシネーション検出の難しさを表し，要約対象文書との内容比較が求められると考えられる．

作成した疑似ハルシネーションデータにおいて，言及があると判断された非ハルシネーションの実際の内容正確性は確認していない．今後はこの非ハルシネーション側の内容正確性の確認が求められる．

4. ハルシネーション検出実験

作成した疑似ハルシネーションデータセットを対象に，検出モデルの学習及びテストに使用した検出実験を行う．

図 6 に本実験の流れを示す．要約対象文書と疑似ハルシネーションデータのクエリ要約を入力として，ハルシネーションか否かを検出する．Transformer ベースモデルである BERT[11] は，事前学習に加えて特定のタスクに対する

表 5 疑似ハルシネーションデータの例．下記のクエリ要約のように，元となる本文がなければ，クエリと生成されたクエリ要約だけではそれがハルシネーション否かは判断できない．これは，本タスクの難しさを表している．

クエリ	クエリ要約	ラベル
A は売り上げについて何を考えましたか？	A は売り上げについて、特に女性客や家族連れをターゲットにすることで売り上げを伸ばせると考え、次に入れる店舗の選定に役立てようとしました。	非ハルシネーション
B は既存店・ミスター K について何を考えましたか？	B は既存店・ミスター K について、特にその売り上げや客層を分析し、次に入れる店舗の選定に役立てようとしました。	ハルシネーション

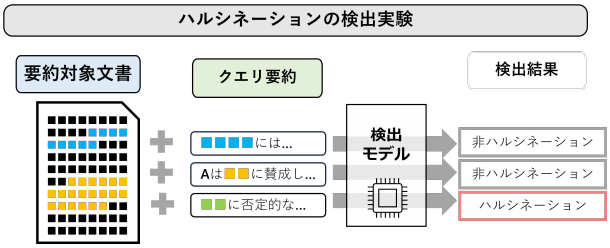

図 6 検出実験におけるハルシネーション検出の流れ．検出モデルは要約対象文書とクエリ要約を入力し，ハルシネーションか否かを出力する．

追加学習を行うことで，タスクに特化し高い性能を示す．しかし，事前学習された BERT モデルは入力として処理可能な最大トークン長に 512 トークンの制限がある．クエリ要約のトークン長は一般的にこれより短く処理可能であるが，要約対象文書はこの制限より長いため直接処理することができない．このため，本実験では要約対象文書の処理を BERT とは異なるモデルを用いて Embedding(埋め込みベクトル) を取得する．この Embedding は要約対象文書全体ごとに取得される．クエリ要約は BERT の処理によって Embedding を取得する．これらの Embedding を用いて検出モデルの構築を行い，ラベルのフィードバックによる学習を行う．本実験では，疑似ハルシネーションデータを学習に用いた交差検証を行う．また，フィードバック範囲を変化させたモデルの実験も行い，比較する．

4.1 OpenAI Embedding

OpenAI Embedding[*4]は，OpenAI 社の API によって提供されるテキストの埋め込みベクトル取得である．OpenAI Embedding は，対話のように長い入力に対しても適切に埋め込みを生成できる能力を持ち，最大 8,191 トークンまでの入力を処理可能である．そのため，3 節で構築した疑似ハルシネーションデータの要約対象文書である Kyutech

[*4] https://platform.openai.com/docs/guides/embeddings

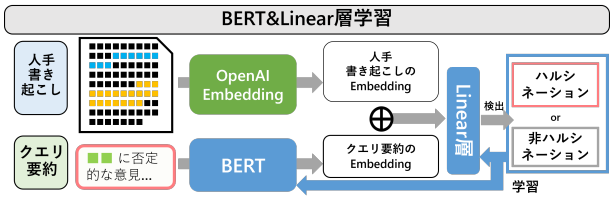

図 7 BERT&Linear 層学習モデルにおける検出の流れ．BERT と Linear 層を学習している．

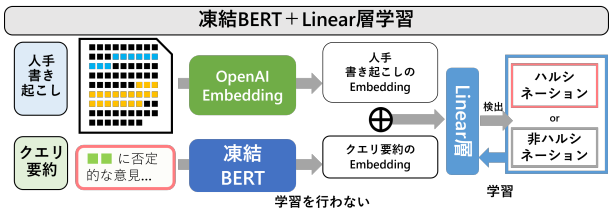

図 8 凍結 BERT+Linear 層学習モデルにおける検出の流れ．凍結 BERT 部分は学習せず，Linear 層のみを学習している．

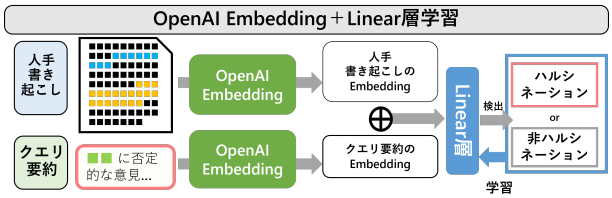

図 9 OpenAI Embedding+Linear 層学習モデルにおける検出の流れ．OpenAI Embedding 部分は学習がなく，Linear 層のみを学習している．

コーパスの対話もほぼ全体の情報を反映した埋め込みを作成することが可能である．疑似ハルシネーションデータの対象の対話を API に入力し，1,536 次元の Embedding を取得する．

4.2 モデル

本節では，ハルシネーションの検出に用いたモデルについて説明する．4.1 節における OpenAI Embedding を利用した 3 つのモデルを構築する．4.3 節では，今回の提案モデルを示す．4.4 節，4.5 節では，提案モデルに対して，学習の有効性を確認する 2 つの比較モデルを示す．

4.3 BERT&Linear 層学習モデル

BERT と OpenAI Embedding を用いた検出モデルの構造を図 7 に示す．要約対象文書の Embedding を OpenAI Embedding によって作成し，クエリ要約の Embedding を BERT によって作成している．一般的な BERT では，Linear 層に BERT の 768 次元の Embedding のみが入力される．これに対して本モデルでは，BERT の 768 次元の Embedding と要約対象文書の Embedding の 1,536 次元の Embedding を結合した 2,304 次元を Linear 層に入力する．両者の埋め込みを結合することで，要約文と要約対象文書間の意味的関係をモデルが学習しやすくすると考えられる．ハルシネーションのラベルを推定し，BERT と Linear 層にフィードバックし学習する．これにより，要約が要約対象文書において言及しているかを高精度に判定し，ハルシネーション検出可能であると考えられる．

4.4 凍結 BERT+Linear 層学習モデル

凍結した BERT と OpenAI Embedding を用いた検出モデルの構造を図 8 に示す．4.3 節の BERT+OpenAI Embedding モデルに対して，BERT の学習が有効であるかを確認するために BERT の学習を行わず，Linear 層のみを学習するモデルである．そのため，BERT はパラメータの更新を行わないよう凍結する．要約対象文書の Embedding を OpenAI Embedding によって作成し，クエリ要約の Embedding を凍結した BERT によって作成している．BERT 部分の凍結により，BERT は 768 次元の Embedding 取得にのみ利用され，推定ラベルがフィードバックされる学習は Linear 層のみとなる．

4.5 OpenAI Embedding+Linear 層学習モデル

OpenAI Embedding のみを用いた検出モデルの構造を図 9 に示す．4.3 節の検出モデルの学習 BERT と，OpenAI Embedding が有効であるかを確認する．要約対象文書の Embedding を OpenAI Embedding によって作成し，クエリ要約の Embedding も OpenAI Embedding によって作成している．本手法では，要約対象文書の Embedding の 1,536 次元とクエリ要約の Embedding の 1,536 次元の Embedding を結合した 3,072 次元を Linear 層に入力する．推定ラベルのフィードバックによる学習は Linear 層のみとなる．

4.6 実験設定

本実験では Kyutech コーパスの疑似ハルシネーション 9 対話に対して，学習:検証:テスト=7:1:1 として交差検証を行う．

本実験では，東北大学によって公開されている BERT モデル[*5]を用いる．BERT の学習の実験設定は，バッチサイズ 16，学習率 2e-5，損失関数は CrossEntropyLoss を用いている．エポック数について，5 エポックと 20 エポックを比較する．

疑似ハルシネーションデータは，表 4 の通り，ハルシネーションと非ハルシネーションの間にデータ数の不均衡がある．そのため，通常の比率だけでなく，アンダーサンプリングとオーバーサンプリングを学習データに適用する．アンダーサンプリングでは，少数派データであるハルシネーションデータと同数まで，非ハルシネーションデータをランダムに減らす．オーバーサンプリングでは，ハルシネーションデータを非ハルシネーションデータと同数までコピーする．これらについても比較を行う．

[*5] https://github.com/cl-tohoku/bert-japanese

表 6 ハルシネーション検出実験結果．ハルシネーション Recall の最高は BERT&Linear 層学習モデルの 20 エポックアンダーサンプリングのモデルであり，重み付き平均 F1 の最高はオーバーサンプリングのモデルであった．

			ハルシネーション			非ハルシネーション			
モデル	エポック数	サンプリング	Pre	Rec	F1	Pre	Rec	F1	重み付き平均 F1
BERT&Linear 層学習	5	-	**0.58**	0.48	0.53	0.87	0.91	0.89	0.81
	20	-	0.53	0.43	0.48	0.86	0.90	0.88	0.79
	5	アンダー	0.31	0.65	0.42	0.87	0.62	0.73	0.66
	20	アンダー	0.45	**0.72**	**0.55**	0.91	0.76	0.83	0.77
	5	オーバー	0.52	0.57	0.54	0.88	0.86	0.87	0.80
	20	オーバー	0.46	0.51	0.49	**0.93**	0.92	**0.92**	**0.87**
凍結 BERT+Linear 層学習	5	-	0.00	0.00	0.00	0.79	**1.00**	0.88	0.70
	5	アンダー	0.20	0.51	0.28	0.78	0.45	0.57	0.51
	5	オーバー	0.31	0.43	0.36	0.83	0.74	0.78	0.69
OpenAI Embedding+Linear 層学習	5	-	0.00	0.00	0.00	0.79	**1.00**	0.88	0.70
	5	アンダー	0.37	0.43	0.39	0.84	0.80	0.82	0.73
	5	オーバー	0.40	0.63	0.49	0.88	0.75	0.81	0.74

精度として，ハルシネーションと非ハルシネーションについてのそれぞれの Precision，Recall，F1 を示す．さらに，ハルシネーションと非ハルシネーションの重み付き平均の F1 を示す．精度について，もれなくハルシネーションを検出しつつ，検出のバランスも考慮するために，ハルシネーションの Recall と重み付き平均 F1 を重視する．

4.7 実験結果

実験結果を表 6 に示す．BERT と Linear 層を学習する BERT&Linear 層学習モデルにおいて，ハルシネーションの検出の最高 Recall はアンダーサンプリングにおける 20 エポックの場合で 0.72 であった．また，最高の重み付き平均 F1 はオーバーサンプリングにおける 20 エポックの場合で 0.87 であった．エポック数について，一般的にエポック数が増加するほど精度が向上するが，サンプリングごとに傾向が異なった．通常のサンプリングとオーバーサンプリングにおいては，エポック数の増加でハルシネーション検出精度が減少している．これに対して，アンダーサンプリングではハルシネーション検出精度が向上している．これは，アンダーサンプリングによる学習件数の減少が原因と考えられる．

また，サンプリング間の比較では，ハルシネーションをもれなく検出する観点においてはアンダーサンプリングが一定の Recall を達成しているため有効であるといえる．バランスも考慮すると，重み付き平均の観点ではオーバーサンプリングが通常のサンプリングに比べてより高い精度となっている．ハルシネーションの検出 Recall はアンダーサンプリングがより高い精度となっている．ハルシネーションの検出と，重み付き平均の両方を考慮すると，BERT+OpenAI Embedding モデルの 20 エポックかつアンダーサンプリング手法が有効であると考えられる．

凍結した BERT と OpenAI Embedding を用いた凍結 BERT+Linear 層学習モデルは非ハルシネーションに偏った検出結果となっている．サンプリングの適用によって精度向上を達成しているが，BERT&Linear 層学習モデルに及ぶものではない．OpenAI Embedding のみを用いた OpenAI Embedding+Linear 層学習の場合も同様に非ハルシネーション側に精度が偏った結果となった．サンプリングの適用によって一定程度のバランスをとった結果となっているが，BERT&Linear 層学習モデルの精度を超えるものではない．このため，学習データとして疑似ハルシネーションデータを用いる場合，少なくともクエリ要約の Embedding についての学習が有効であると言える．

本実験では，クエリ要約の Embedding の学習は行ったが，要約対象文書の Embedding の学習は行っていない．要約対象文書の適切な Embedding を学習することや，Linear 層に入力する追加素性によってさらなる精度向上が考えられる．また，本実験ではクエリ要約と要約対象文書のみを組み合わせているが，クエリそのものの情報は利用していない．クエリそのものの Embedding を取得し Linear 層への入力に追加することで，ハルシネーションを検出精度向上が考えられる．

5. おわりに

本研究では，クエリ要約タスクを対象として，疑似ハルシネーションデータの作成を行った．また，作成データを対象としたハルシネーションを検出するハルシネーション検出実験を行った．

疑似ハルシネーションデータの作成においては，既存の Kyutech コーパスを用いた．Kyutech コーパスに対して与えられているトピックタグを用いて，言及の有無を判別した．QMSum をもとにしたクエリテンプレートとトピック

タグを利用しクエリを作成した．Kyutech コーパスの人手書き起こしと作成したクエリをもとに，ChatGPT を用いてクエリ要約を作成した．トピックタグの有無から，ハルシネーションか否かのラベルを付与し，クエリ要約を作成した 3,152 件のうちハルシネーションが 661 件であった．今後は自動トピック生成と組み合わせることでより多くのコーパスを対象に疑似ハルシネーションデータを作成することが考えられる．また，言及した内容についてクエリ要約を生成した場合の正確性については確認していない課題がある．

ハルシネーションの検出実験においては，作成したデータを学習に利用したモデルの構築を行った．特に，要約対象文書を OpenAI Embedding によって処理し，クエリ要約を BERT モデルに学習させるモデルの BERT&Linear 層学習モデルが特に有効であると確認された．本研究では，要約対象文書とクエリ要約のみを入力としている．今後はクエリの情報もモデルの学習に使用することで，より高精度にハルシネーションを検出できると考えられる．

謝辞

本研究は科研費 23K11368 の一部です．

参考文献

[1] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. Vol. 55, pp. 1–38, 2023.

[2] Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2214–2220, 2019.

[3] Takashi Yamamura, Kazutaka Shimada, and Shintaro Kawahara. The Kyutech corpus and topic segmentation using a combined method. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)*, pp. 95–104, 2016.

[4] Hoa Trang Dang. DUC 2005: Evaluation of Question-Focused Summarization Systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pp. 48–55, 2006.

[5] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5905–5921, 2021.

[6] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, 2020.

[7] Tomonori Kodaira and Mamoru Komachi. The Rule of Three: Abstractive Text Summarization in Three Bullet Points. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, 2018.

[8] Keisuke Iwamoto and Kazutaka Shimada. Dataset Construction and Verification for Detecting Factual Inconsistency in Japanese Summarization. In *2024 16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 243–248, 2024.

[9] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464, 2023.

[10] Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. TofuEval: Evaluating Hallucinations of LLMs on Topic-Focused Dialogue Summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4455–4480, 2024.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.