令和 5 年度茨城大学工学部情報工学科

卒業研究論文

RAG における小説データベースの Chunk Size と Overlap Size と Embedding モデルの効果

所属　情報工学科

著者　阿部晃弥 (20T4001L)

指導教員　新納浩幸教授

令和 6 年 2 月 2 日 (金)

令和 5 年度茨城大学工学部情報工学科　卒業研究論文

RAG における小説データベースの Chunk Size と Overlap Size と Embedding モデルの効果

著者

阿部晃弥 (20T4001L)

指導教員

新納浩幸教授

論文要旨

近年，自然言語処理の分野では，大規模言語モデルが台頭し，従来の言語モデルをはるかに上回る性能を示している．特に OpenAI 社が提供している ChatGPT は，社会全体に大きな影響を与えた．一方で，大規模言語モデルには，大きな課題も存在し，その一つが不誠実で無意味な回答を生成してしまう，幻覚 (Hallucinations) である．特に「推論の処理を行っても答えられない質問」，つまり「知っていないと答えられない質問」に対する回答には必ず幻覚が含まれると考えられる．しかしそのような質問に対しても正しい回答が生成される場合もある．

本論文では，まず「知っていないと答えられない質問」に対して，どのような要因が幻覚の有無を生じさせているのかを調査するため，「小説のあらすじ」を題材にして，各種大規模言語モデルの生成文を調査し，その特徴を考察する．また，辞書的マッチングを用いて，小説のあらすじ内の幻覚の有無判定を行うことも試みた．その結果，Web 上における検索結果の量に比例して幻覚の発生率が減少し，同一モデルの生成結果を用いた単純な辞書的マッチングの方法によって，60% から 70% の幻覚の判定に成功した．

そして，大規模言語モデルによる幻覚の発生を抑制するための研究について調査し，その中の一つとして，RAG(Retrieval-Augmented Generation) に注目した．これは，文書をベクトル化して保存しておき，正確な回答に必要な文書を prompt に組み込んで生成する手法である．そこで，日本語の小説をデータベースとした RAG において，ベクトル・インデックスの作成の際，Chunk Size，Overlap Size 及び Embedding モデルを変更した場合の Retrieval の結果，回答への影響を検証した．実験の結果，Chunk Size には規則性が見られないが，Overlap Size は大きいほど概ね良い結果が確認できた．また，Chunk Size や Overlap Size の値とは無関係に，Embedding モデル間の性能差が確認された．

目次

表目次

図目次

第 1 章

序論

近年，自然言語処理の分野では，ChatGPT を始めとする大規模言語モデル (Larage Language Model，以下 LLM と略す) が高い性能を示しており，多くの評価指標で従来のスコアを大きく上回る結果を残している．しかし，ある特定の分野の詳しい知識や比較的新しい知識といった学習データに含まれていない情報を扱うことが難しいという問題がある．この問題によって生じる現象が，言語モデルが不適切な回答を生成してしまう Hallucinations(幻覚) である．この問題の単純な解決策として，言語モデルを再度学習させる方法や別のデータを用いて追加学習する方法が考えられる．しかし，LLM は大規模な言語モデルであるという特性上，追加学習が困難であるという性質をもっており，その欠点を補うために，外部知識を様々な形で LLM に組み込むといった研究もおこなわれている．その一つが RAG(Retrieval-Augmented Generation) [1] である．RAG は，外部知識をベクトル・インデックスという形で保存し，入力文をクエリとしてインデックスに問い合わせ，関連する知識を入力文と合わせて LLM に与えることで，外部知識に基づいた生成を行わせることを目的とした手法である．一般的に，外部知識をテキストとして与える場合，テキストを Chunk Size ごとに切り分け，Embedding モデルを用いてベクトル化し，インデックスに格納する形式が用いられる．

本稿ではまず小説のあらすじを題材として幻覚に関する調査を行う．ChatGPT，Bard にあらすじを生成させ，それぞれの小説のあらすじに含まれる幻覚の割合，小説の有名度を測る指標として Web 上の検索件数，検索結果の上位サイトにあらすじを含む割合の情報を用いて関係性を調べる．次に，生成したあらすじに対して単純な辞書的マッチングの手法によって幻覚の判定が可能であるか調査する．

そして本実験では，日本語の小説の内容を外部知識として与えた際，Chunk Size,

Chunk 間の重複を認める Overlap Size が Retrieval の結果，回答へ与える影響について調査を行う．また，Chunk のベクトル化の際には，3 種類の Embedding モデルを利用する．それぞれのモデルで同様の条件で実験を行った場合の結果を比較し，その特徴や生成結果に関して考察を行う．

第 2 章

関連研究

2.1 単語の埋め込み表現

2.1.1 Word2Vec

現在の自然言語処理の分野の基盤となった技術が，単語の分散表現である．単語をベクトルで表現することで, 単語の意味を定量的に把握することを可能にした．Word2Vec [2] は，単語の意味はその周囲の単語によって決定されるという分布仮説に基づいて，単語をベクトル化する手法である．具体的には，2 層のニューラルネットワークによって学習を行い，大量のコーパスから単語の意味を取得する．

Word2Vec には，CBOW と Skip-gram の 2 種類のモデルが存在する．図 2.1 の左のモデルが CBOW である．このモデルは，周囲の単語からターゲットの単語を推測する事を目的としたニューラルネットワークになっている．出力層で得られるターゲットの単語の出現確率と，正解の単語の one-hot ベクトルを比較し，得られる損失関数の値を最小化するように学習する．

図 2.1 の右のモデルが Skip-gram である．このモデルは，一つの単語から周囲の単語を推測する事を目的としたニューラルネットワークになっている．出力層で得られる周囲の単語の出現確率と，正解の単語の one-hot ベクトルを比較し，得られる損失関数の値を最小化するように学習する．損失関数には，一般的に周囲の単語それぞれの損失の合計や平均が用いられる．

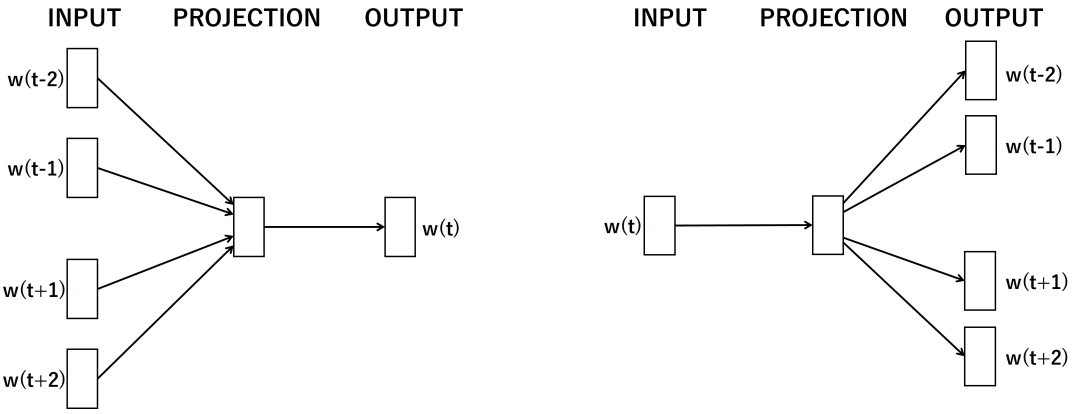

図 2.1: Word2Vec の構造

2.1.2　ELMo(Embeddings from Language Models)

前述の Word2Vec のように一つの単語が対応する一つのベクトルで表現されるような手法には，同じ単語の文脈による意味の違いを表現できないという問題があった．そこで，文脈に応じた単語の埋め込み表現を求めることができる手法として，ELMo [3] が提案された．ELMo は，図 2.2 に示すような双方向の LSTM(Long Short-Term Memory) を用いて学習を行う．事前学習では，次の単語を予測するタスクによって学習を行う．損失関数は，先の単語を見ることができないように，前から予測するモデルと後ろから予測するモデルのそれぞれの出力の損失の合計としている．各単語の埋め込み表現は，学習した LSTM の隠れ層と最初の埋め込みをそれぞれ結合し，加重総和算によって計算することで求めることができる．

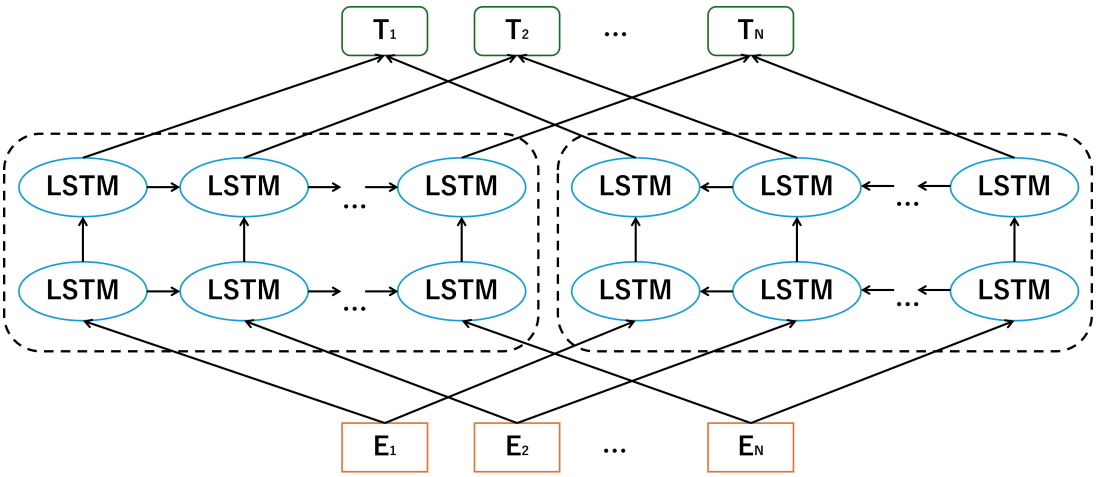

図 2.2: ELMo の構造

2.1.3　GPT

GPT [4] は文章を順に読む片方向に学習する Transformer [5] を用いたモデルである．RNN や CNN を使用せず，Attention のみによって構成された Transformer を用いることで，単語を並列化して処理することが可能となり，訓練時間の大幅な削減が可能になった．言語モデルの事前学習は，直前の k 個の単語を使って対数尤度を最大化させるように次の単語を求める問題により学習を行う．$U = \{u_1, \ldots, u_n\}$ はラベルなしのトークン化されたデータを意味する．

$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta) \tag{2.1}$$

2.1.4　BERT(Bidirectional Encoder Representations from Transformers)

BERT [6] は文章を双方向に学習する Transformer を用いたモデルである．BERT では，双方向の文脈を考慮した事前学習のために，Masked Language Model と Next Sentence Prediction の 2 種類の教師なし学習を行う．

Masked Language Model は，文のトークンの一部を「予測トークン」に置き換え，予測トークンの位置の本来のトークンを予測する問題である．一般的には，文のトークンの 15% を以下のトークンのように置き換える．

- 80% は [Mask] トークンに置き換える．
- 10% はランダムな別のトークンに置き換える．
- 10% は元のトークンとして残す．

Next Sentence Prediction は，入力として与えられた 2 つの文が連続な文であるか予測する問題である．文頭に付与される特殊トークン [CLS] の情報を用いて文脈を判別させ，学習する．

2.2　幻覚に関する調査

自然言語生成 (NLG) は，Transformer ベースの言語モデルなどの配列間深層学習技術の開発により，近年，指数関数的に向上している．しかし，ディープラーニングに基づく

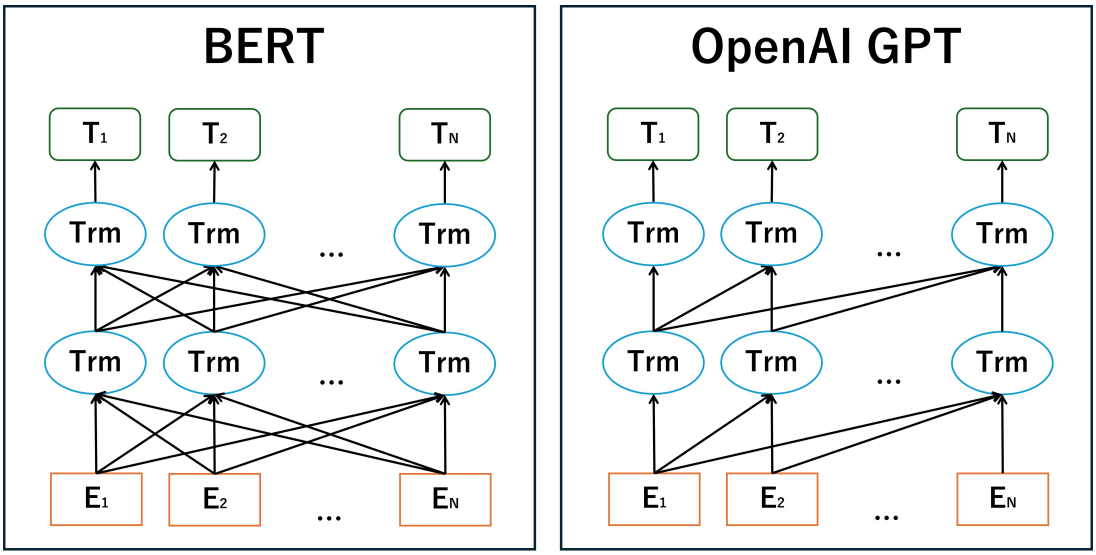

図 2.3: BERT と GPT の構造

生成は，意図しないテキストを生成しやすく，システム性能を低下させ，使用者の期待に応えられないことも明らかになっている．この問題に対処するため，幻覚文の測定と軽減に関する多くの研究が発表されている．それらの幻覚に関する調査を包括的に検討された論文 [7] を参考に，自然言語処理 (NLP) の分野幻覚問題の研究進展と課題について紹介する．

2.2.1　幻覚の定義

「NLG モデルが不誠実で無意味なテキストを生成する」という好ましくない現象に対して用いられる言葉である．この現象が，現実と感じられる非現実的な知覚という心理的な意味で使われる幻覚と同じような特徴を持つため，このような用語が使われるようになった．この問題の本質には，幻覚化したテキストは，不誠実で無意味であるにもかかわらず，流暢で自然な印象を与えることにある．提供される実際のコンテキストに根拠があるように見えるが，実際にはそのようなコンテキストの存在を特定したり検証したりするのは難しいことが問題の解決を妨げている．

2.2.2　幻覚の種類

Intrinsic Hallucinations(**本質的な幻覚**)

ソースコンテンツと矛盾するような生成されたアウトプットのことを指す．

表 2.1: Intrinsic Hallucinations の具体例

ソースコンテンツ	The first vaccine for Ebola was approved by FDA in 2019
アウトプット	The first Ebola vaccine was approved in 2021

Extrinsic Hallucinations(**外在的な幻覚**)

ソースコンテンツから検証できない生成されたアウトプット，つまりソースから支持も矛盾もできない出力のことを指す．注目すべきは，外在的な幻覚は事実上正しい外部情報からのものである可能性があるため，常に誤りであるとは限らないことである．このような情報の検証不可能な側面が事実上の安全性の観点からリスクを高めるため，外在的な幻覚は注意深く扱われている．

2.2.3　幻覚の原因

データ

データによって幻覚が起こる主な原因は source-reference(target) の乖離によるものである．

大規模なデータセットを収集する場合，いくつかの研究では，ソースとターゲットとして実際の文や表を発見的な手法によって選択し，ペアリングする．この発見的な手法による選択のため，ターゲットとなるリファレンスには，ソースでサポートできない情報が含まれている可能性がある．

また，オープンドメインの対話システムは，ユーザの入力や対話履歴，知識ソースに存在しない関連する事実を提示する．．とが許容される．これにより，対話生成の魅力と多様性を向上させることができる．しかしこのような特性は，必然的に外在的幻覚につながることが判明している．

訓練と推論

言語モデル中のエンコーダは，入力されたテキストを理解し，意味のある表現にエンコードする役割を担っている．そのため，欠陥のあるエンコーダが幻覚の程度に影響を与える可能性がある．エンコーダが学習データの相関関係を間違って学習すると，入力と乖離した誤った生成を行う可能性がある．

そして，言語モデル中のデコーダが幻覚に寄与する可能性も示唆されている．デコーダが符号化された入力ソースの間違った部分に注目することで，誤った関連付けをしてしまい，2 つの類似したエンティティの間で事実が混在する生成をもたらす．デコード方式の設計が，最も確率の高いトークンを選ぶのではなく，上位 k 個のサンプルからサンプリングすることで意図的に「ランダム性」を持たせ，生成の多様性を高める場合，幻覚コンテンツを含む可能性を高めていると推測されている．

また，訓練時と推論時のデコーディングの不一致である露出バイアスも原因となる．デコーダは，教師による強制的な最尤推定 (MLE) 学習で訓練する．この学習では，デコーダは直前までのトークンを基に次のトークンを予測するように学習する．しかし，推論生成の際に，モデルはそれ自身が以前に生成した履歴シーケンスを条件として次のトークンを生成する．このような不一致は，特に対象となる配列が長くなった場合に，ますます誤った生成を招くことがある．

最後に，パラメトリックな知識への偏りによる幻覚である．大規模なコーパスに対するモデルの事前学習は，モデルのパラメータに知識を記憶させる．大規模な事前学習済みモデルは，汎化性と網羅性を提供する上で強力だが，このようなモデルが提供された入力よりもパラメトリックな知識を優先することが判明している．つまり，入力ソースからの情報ではなく，パラメトリックな知識で出力を生成することを優先するモデルは，出力に幻覚を生み出しやすい．

2.3　幻覚を低減する研究

幻覚を低減する研究の例として，NPH と RAG を紹介する．本稿では RAG について取り上げ，基本的な RAG の検索部分の設定パラメータを調整することで生じる性能の差異を調べている．

2.3.1　NPH(Nueral Path Hunter)

NPH [8] は，大規模言語モデルによって生成された応答に対して，外部知識である知識グラフによって修正を行い，応答の精度を向上させる手法である．NPH の全体像を表 2.4 に示す．NPH は，幻覚の原因となりそうなトークンを特定し，クエリ信号を用いて K ホップサブグラフ上を検索し，正確なデータに修正する．このモデルは，幻覚の判別

を行う Critic モジュールと，幻覚の修正を行う Retrieval モジュールからなる．

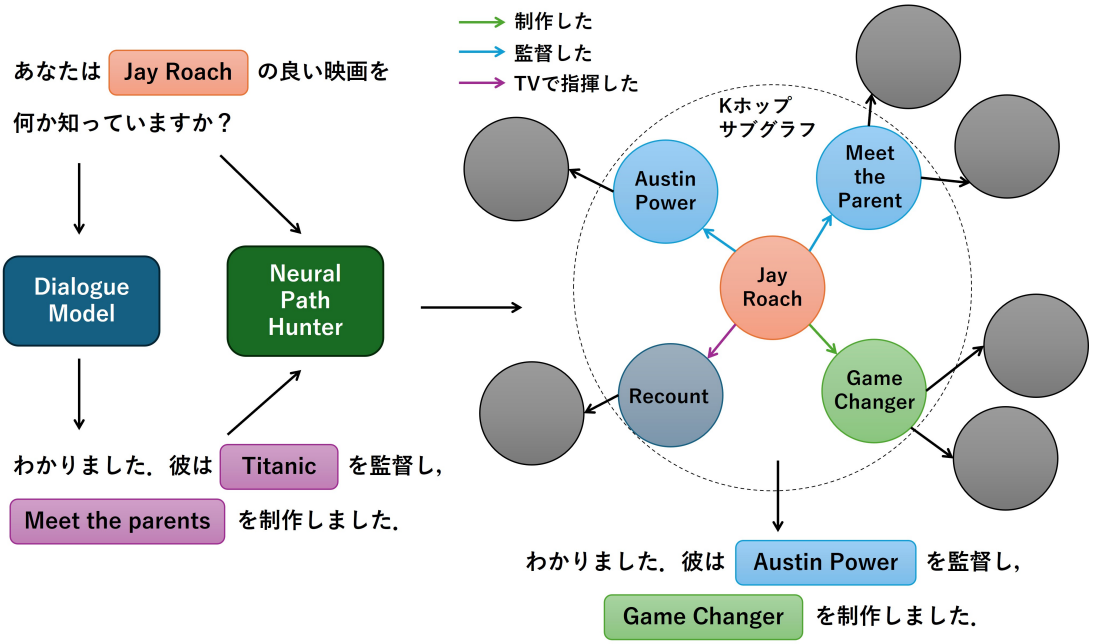

図 2.4: Neural Path Hunter の全体像

知識グラフ

有向辺をもつ三つ組 $t = < [SBJ], [PRE], [OBJ] >$ の集合によって形成される．$[SBJ], [OBJ] \in V$ は名詞を指し，主語と目的語を表すノードである．$[PRE] \in R$ は主語と目的語の関係を表す述語である．

Critic モジュール

幻覚の原因となるトークンを判別するために，幻覚の判別モジュールを訓練する．モジュールの学習のため，各トークンの位置で幻覚か否かの二値ラベルを予測するシーケンスラベリングタスクとして問題を扱う．

Retrieval モジュール

Critic モジュールでマスクされた表現を受け入れ，これらのトークンの文脈表現を構築し，サブグラフによってより信頼性の高いエンティティを検索する．

2.3.2　RAG

RAG [1] は，事前学習されたパラメトリック・メモリとノンパラメトリック・メモリを組み合わせたモデルである．ノンパラメトリック・モデルは Query Encoder と Document Index からなる Retriever である．パラメトリック・メモリは Retriever から受け取った Document を用いて文章を生成する Generator である．Retriever が入力に応じて Document を提供することで，Generator は知識に基づいた生成を行うことが可能になる．RAG の構造の全体像を図 2.5 に示す．

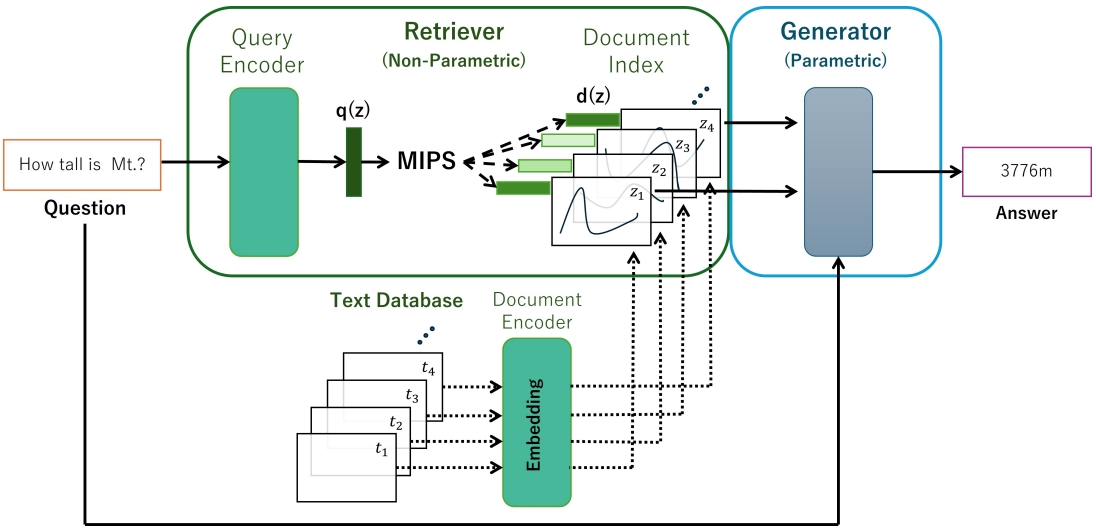

図 2.5: RAG の全体像

この構造の利点として，Document Index を作成する際のデータベースに変更を加えることで，知識を直接拡張したり修正したりすることができ，アクセスされた知識を検査したり解釈したりすることができることが挙げられる．これは，事前学習モデルの課題であった知識の拡張，修正が困難であったり，幻覚 (Hallucinations) を生み出す可能性があるという点に関して対処できることを示す．論文内では，事前学習モデルのみのベースラインと比較して，より具体的で多様かつ事実に基づいた生成結果を示している．外部の知識源にアクセスしなければ実行することが合理的に期待できないような，知識集約的なタスクに関して当時の最先端の結果を達成した．

また，幻覚に対する解決策として，RAG の研究は広く行われており，様々な工夫が施された手法が数多く提案されている．具体的な手法の例として，HyDE と Self-RAG について以下に紹介する．しかしながら，本稿はこのような拡張した手法を提案するもの

ではなく，基本的な RAG の検索部分の設定パラメータを調整することで，性能にどの程度の差異が生じるかを調べている．

HyDE

HyDE [9] は，RAG の検索手法に対して工夫を加えたモデルとなる．RAG では，与えられたクエリをベクトル化し，ベクトル・インデックスに問い合わせることで関連文章を取得する．一方 HyDE では，最初に与えられたクエリを命令追従型の言語モデル (e.g. InstructGPT) に渡し，そのクエリに回答できるような仮説のドキュメントを生成する．そして，生成した仮説のドキュメントをベクトル化し，ベクトル・インデックスに問い合わせることで関連文章を取得する．与えられたクエリに回答できるようなドキュメントは，実際の関連文章に似たドキュメントになるという考えを利用したモデルとなっている．論文内の実験では，様々なタスクと言語において，Fine-tuning された検索に匹敵する強力な性能を示した．

Self-RAG

RAG の検索部以外を改良した手法として，Self-RAG [10] がある．RAG は，回答するために外部知識が不要である質問に対しても，Retrieval した文章を提供し，回答の生成を行う．よって，言語モデルの汎用性を低下させ，回答の精度の低下につながる可能性がある．そこで Self RAG [10] では，Refrection Token を導入し，この token に外部知識の必要性を判断させるという手法である．具体的には，最初に入力文に検索が必要か判断し，必要なら関連文書を検索し，取得した文書の関連性，出力の支持度，出力の有用性を評価し，回答の精度の向上を目指す．論文内の実験では，様々なタスクにおいて，最新の LLM や検索補強モデルを大幅に上回る性能を示した．

第 3 章

小説あらすじに関する幻覚の調査

3.1 調査の背景・目的

大規模言語モデルは，事前学習により得られた知識を基に入力に対する出力を生成する．つまり，「推論の処理を行っても答えられない質問」すなわち「知っていなければ答えられない質問」に対する回答には必ず幻覚が含まれると考えられる．その一方で，そのような質問に対しても幻覚を含まない，正しい回答が生成される場合もある．「知っていないと答えられない質問」に対して，どのような要因が幻覚の有無を生じさせているのかを明らかにすることは，今後の LLM の改良にとって重要である．そこで，本調査では「知っていないと答えられない質問」として「小説のあらすじ」を題材にして，各種 LLM の生成文を調査する．

3.2 調査の方法

3.2.1 生成モデル

小説のあらすじの生成には，OpenAI 社から提供されている ChatGPT[*1]及び Google から提供されている Bard[*2]を用いた．

[*1] https://openai.com/chatgpt

[*2] https://bard.google.com/chat

ChatGPT

回答の生成には GPT-3.5 を用いた．調査時期は 2023 年 7 月から 8 月にかけてであり，回答の生成には Web 上の知識は使われない．

Bard

調査時期は 2023 年 7 月から 9 月にかけてである．ChatGPT と異なり，回答の生成には Web 上の知識を用いている．

3.2.2　対象の小説

太宰治，村上春樹，青山七恵の 3 名の作品を調査の対象とする．それぞれの作家について，最も有名な作品に加え，ランダムに抽出した 3 作品または 4 作品についてを調査の対象とする．なお，最も有名な作品は，ChatGPT に対してそれぞれの作家の最も有名な作品名について回答を生成させ，10 回の生成結果のうち，回答数が最も多い作品とする．

3.2.3　回答の生成

3.2.2 節の対象の小説について，3.2.1 節の生成モデルを用いてあらすじを生成させた．生成モデルに与えるプロンプトは，生成モデル問わず，**「*[作家名]* の小説である *[作品名]* のあらすじを一文で教えてください」**で統一した．小説のあらすじは，各生成モデルごとに，各作品について 20 回ずつ生成させた．このとき，会話履歴がモデルの生成結果に影響しないようにするため，回答の生成ごとに会話をリセットした．

3.2.4　生成文の評価

3.2.3 節で生成した小説のあらすじに幻覚が含まれるか評価した．評価は執筆者自身による主観評価を用いる．

3.3　幻覚の調査結果

3.2 節で述べた通り，大規模言語モデルが生成する小説のあらすじについて幻覚が含まれるか調査を行った．それぞれの調査対象の小説について，生成したあらすじに幻覚が含

まれるかをまとめた結果を表 3.1 に示す．ChatGPT の結果に注目すると，太宰治の「人間失格」，村上春樹の「ノルウェイの森」といった有名な作品の幻覚の割合は低いが，それ以外の作品はほとんどの生成結果に幻覚が含まれていることがわかる．一方で Bard の結果に注目すると，全ての作品で ChatGPT より幻覚の割合が低い．この結果から，回答に知識が必要な質問に対しては，Web 上の知識を使うことができるモデルであれば，幻覚を含まない回答が生成できるが，外部知識を扱えないモデルは幻覚を含む回答が多くなると考えられる．

表 3.1: あらすじに含まれる幻覚の割合の調査結果

作家名	作品名	ChatGPT	Bard
太宰治	人間失格	0.35	0.13
	女生徒	1.00	0.51
	畜犬談	0.80	0.45
	千代女	1.00	0.83
	フォスフォレッスセンス	1.00	0.30
村上春樹	ノルウェイの森	0.40	0.37
	1963/1982 年のイパネマ娘	0.85	0.58
	4 月のある晴れた朝に 100% の女の子に出会うことについて	1.00	0.68
	スプートニクの恋人	1.00	0.68
青山七恵	ひとり日和	1.00	0.26
	すみれ	1.00	0.17
	わたしの彼氏	1.00	0.80
	みがわり	1.00	0.59

3.4　大規模言語モデルが生成する小説あらすじの特徴

次に，Web 上の検索結果と幻覚の割合の関係を調査する．始めに Google Chrome を用いて，それぞれの調査対象の小説について，検索件数と検索結果の上位サイトにあらすじが含まれているかを調べた．その結果を表 3.2 に示す．

表 3.2: 検索件数と上位 n サイトにあらすじが含まれる割合

作家名	作品名	検索件数	Top5	Top10	Top20
太宰治	人間失格	128000	1.00	0.90	0.70
	女生徒	83700	1.00	0.80	0.70
	畜犬談	1430	1.00	0.90	0.60
	千代女	988	0.40	0.30	0.20
	フォスフォレッスセンス	17700	0.20	0.10	0.05
村上春樹	ノルウェイの森	30700	1.00	1.00	0.75
	1963/1982 年のイパネマ娘	543	0.6	0.30	0.15
	4 月のある晴れた朝に 100% の女の子に出会うことについて	2710	0.60	0.80	0.50
	スプートニクの恋人	4050	0.60	0.70	0.60
青山七恵	ひとり日和	18200	0.20	0.30	0.20
	すみれ	9120	0.20	0.30	0.15
	わたしの彼氏	46100	0.40	0.50	0.25
	みがわり	4480	0.20	0.10	0.10

3.4.1　検索件数と幻覚の割合の関係

最初に，Web 上の検索件数と生成文の幻覚の割合の関係を明らかにするため，グラフを作成した．ChatGPT による生成文の幻覚の割合と，検索件数の関係を図 3.1，Bard による生成文の幻覚の割合と，検索件数の関係を図 3.2 に示す．図中の近似直線から，ChatGPT と Bard には，検索件数との間に相関関係があると考えられる．事実として，ChatGPT による生成文の幻覚の割合と，検索件数の相関係数は*-0.549*，Bard による生成文の幻覚の割合と，検索件数の相関係数は*-0.389* となっており，弱い負の相関関係にあるといえる．つまり，Web 上の検索件数が多い小説ほど，幻覚を含まないあらすじを生成できる．

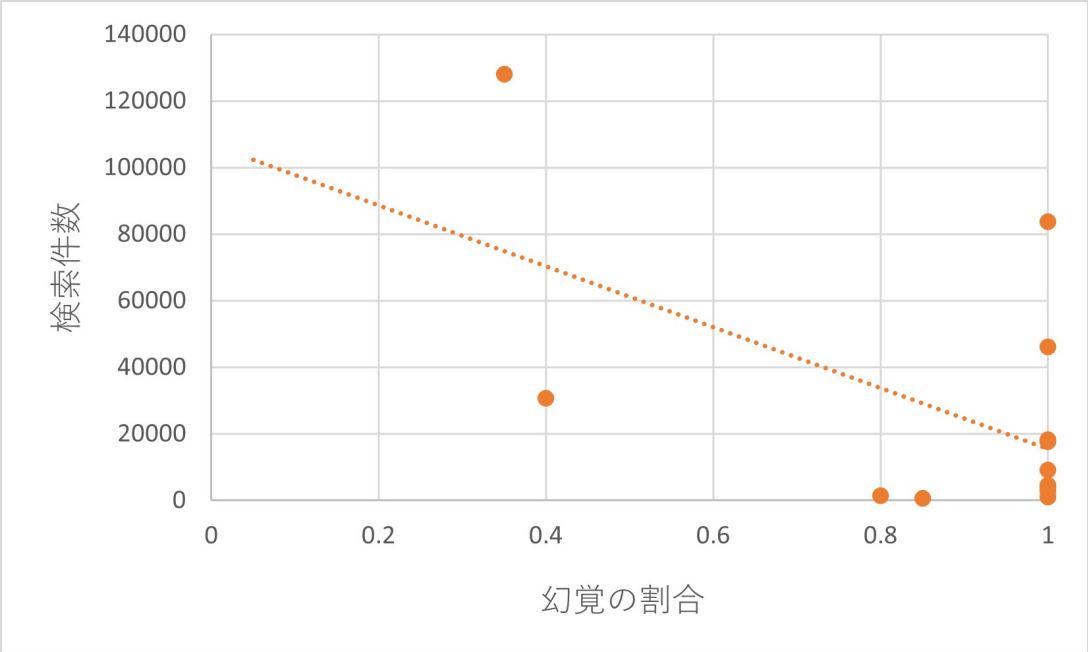

図 3.1: 検索件数と ChatGPT の幻覚の割合の関係

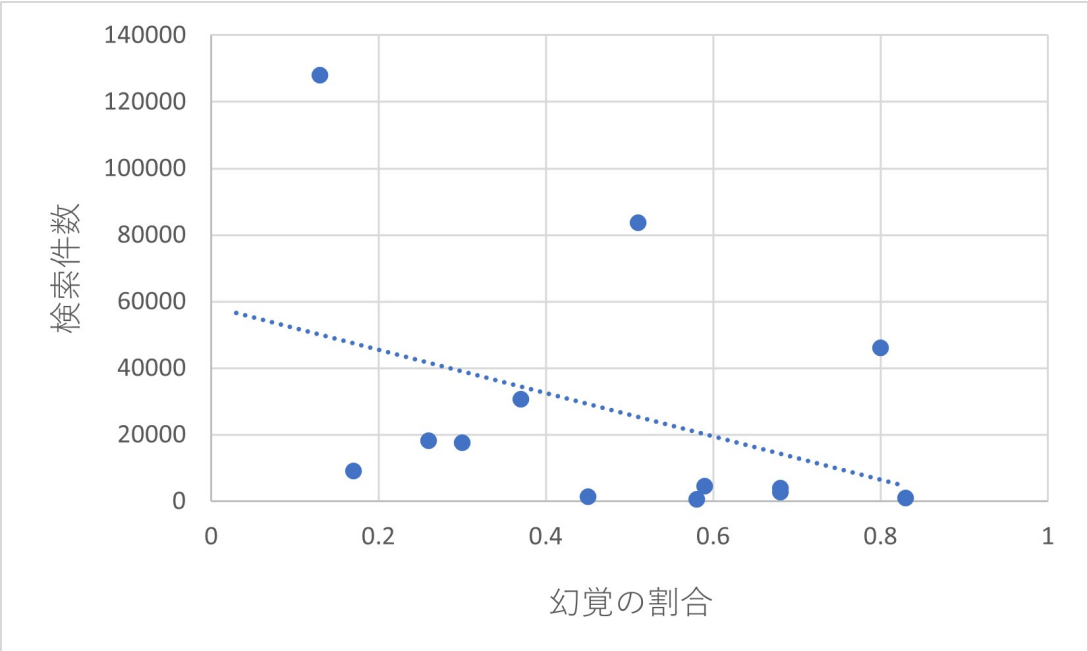

図 3.2: 検索件数と Bard の幻覚の割合の関係

3.4.2　検索結果上位サイトと幻覚の割合の関係

次に，Web 上での検索結果の上位サイト内に小説のあらすじが含まれている割合と，生成文の幻覚の割合の関係を明らかにするため，グラフを作成した．検索結果の上位の 5

つのサイトにあらすじを含む割合と ChatGPT の生成文の幻覚の割合の関係を図 3.3，検索結果の上位の 5 つのサイトにあらすじを含む割合と Bard の生成文の幻覚の割合の関係を図 3.4 に示す．

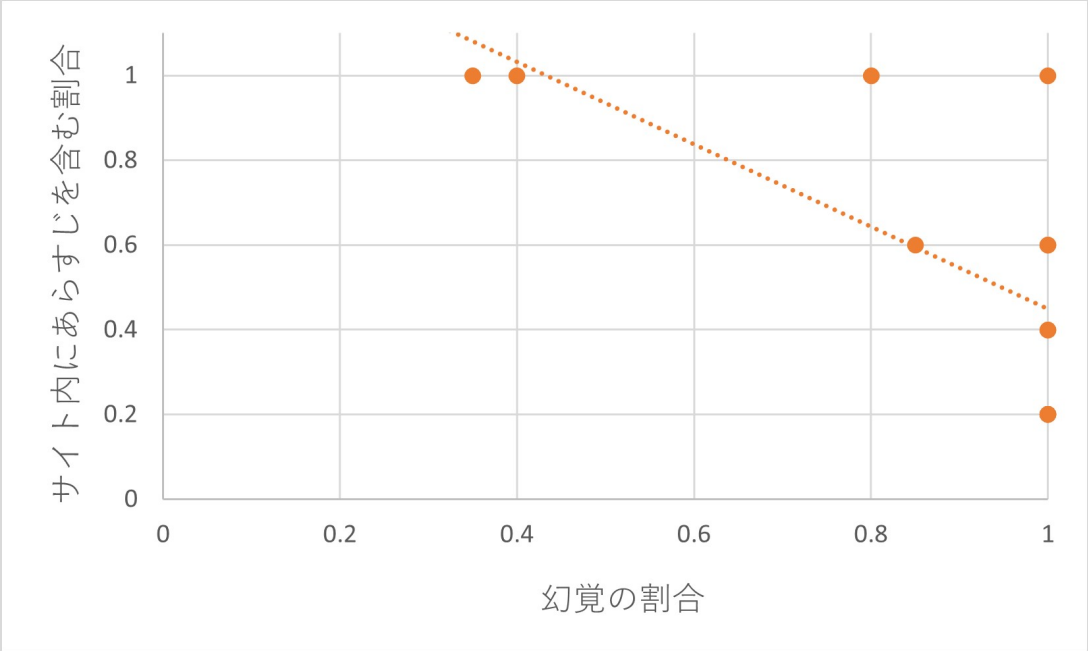

図 3.3: Top5 のサイトのあらすじを含む割合と ChatGPT の幻覚の割合の関係

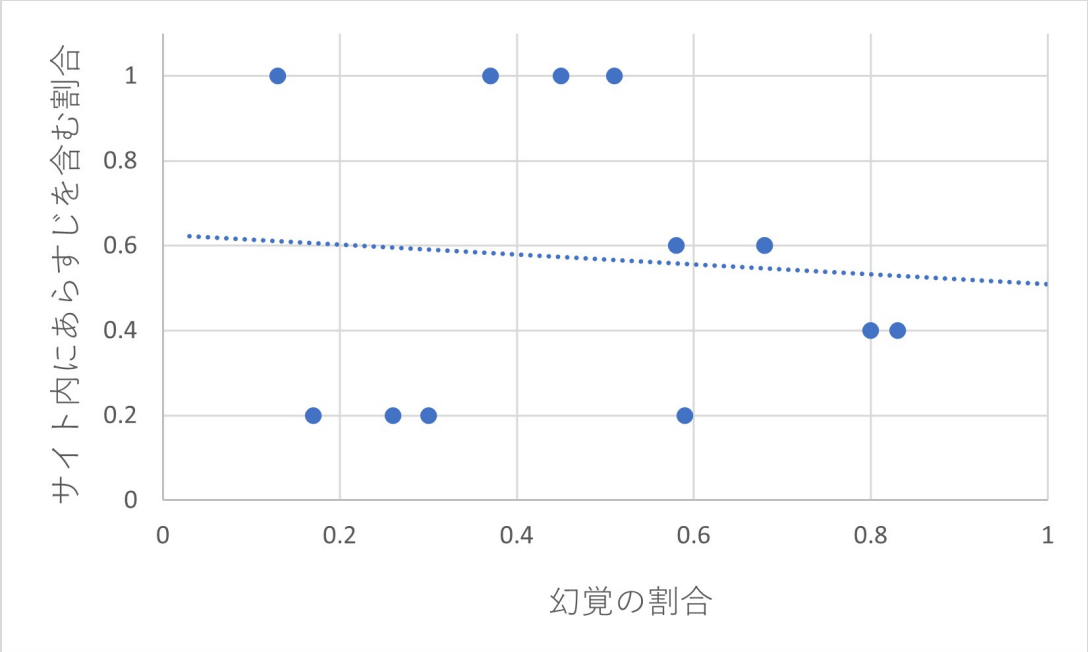

図 3.4: Top5 のサイトのあらすじを含む割合と Bard の幻覚の割合の関係

検索結果の上位 5 サイトにあらすじを含む割合と ChatGPT の生成文の幻覚の割合の

相関係数は*-0.673* であり，図中の近似直線からも負の相関関係がうかがえる．一方で，検索結果の上位 5 サイトにあらすじを含む割合と Bard の生成文の幻覚の割合の相関係数は*-0.080* であり，相関関係は全くないことがわかった．

そして，Web 上での検索結果の上位の 10 個のサイトにあらすじを含む割合と ChatGPT の生成文の幻覚の割合の関係を図 3.5，Web 上での検索結果の上位の 10 個のサイトにあらすじを含む割合と Bard の生成文の幻覚の割合の関係を図 3.6 に示す．

こちらも，検索結果の上位 10 サイトにあらすじを含む割合と ChatGPT の生成文の幻覚の割合の相関係数は*-0.609* であり，図中の近似直線からも負の相関関係があるといえる．検索結果の上位 10 サイトにあらすじを含む割合と Bard の生成文の幻覚の割合の相関係数は*-0.065* であり，相関関係は全くない．

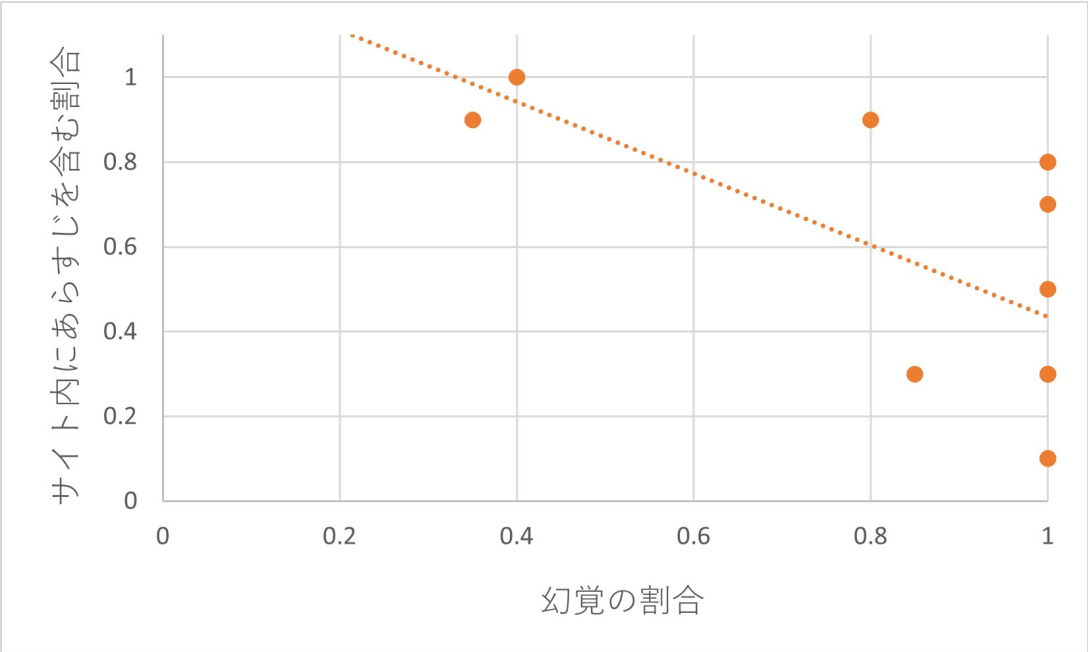

図 3.5: Top10 のサイトのあらすじを含む割合と ChatGPT の幻覚の割合の関係

最後に，Web 上での検索結果の上位の 20 個のサイトにあらすじを含む割合と ChatGPT の生成文の幻覚の割合の関係を図 3.7，Web 上での検索結果の上位の 20 個のサイトにあらすじを含む割合と Bard の生成文の幻覚の割合の関係を図 3.8 に示す．

検索結果の上位 20 サイトにあらすじを含む割合と ChatGPT の生成文の幻覚の割合の相関係数は*-0.598* であり，図中の近似直線からも負の相関関係が読み取れる．検索結果の上位 20 サイトにあらすじを含む割合と Bard の生成文の幻覚の割合の相関係数は*-0.096* であり，相関関係はみられない．

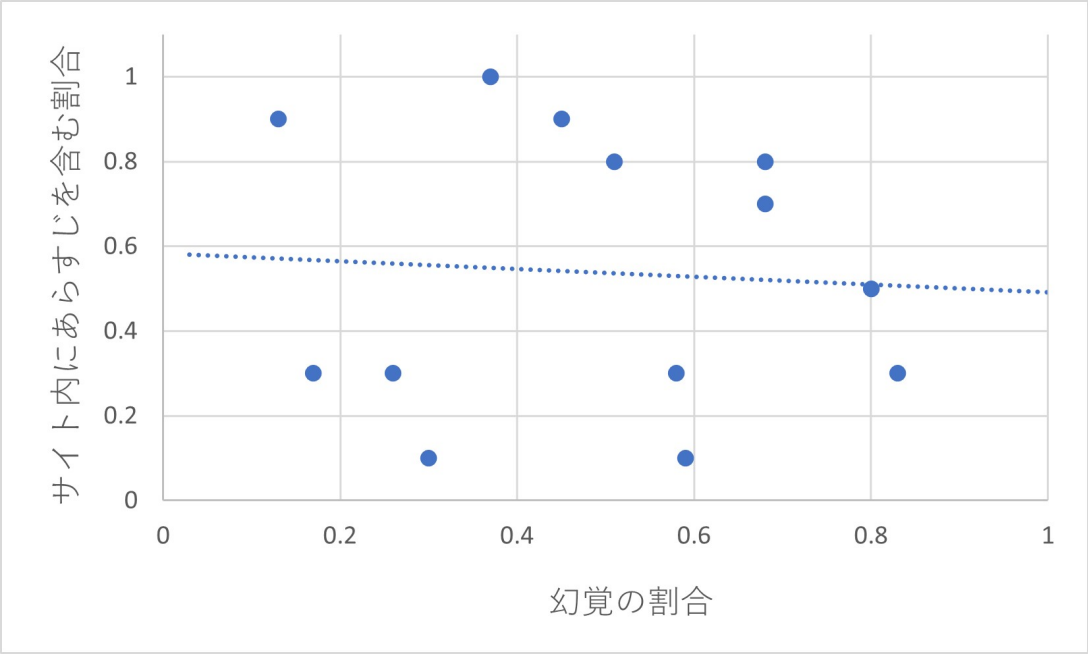

図 3.6: Top10 のサイトのあらすじを含む割合と Bard の幻覚の割合の関係

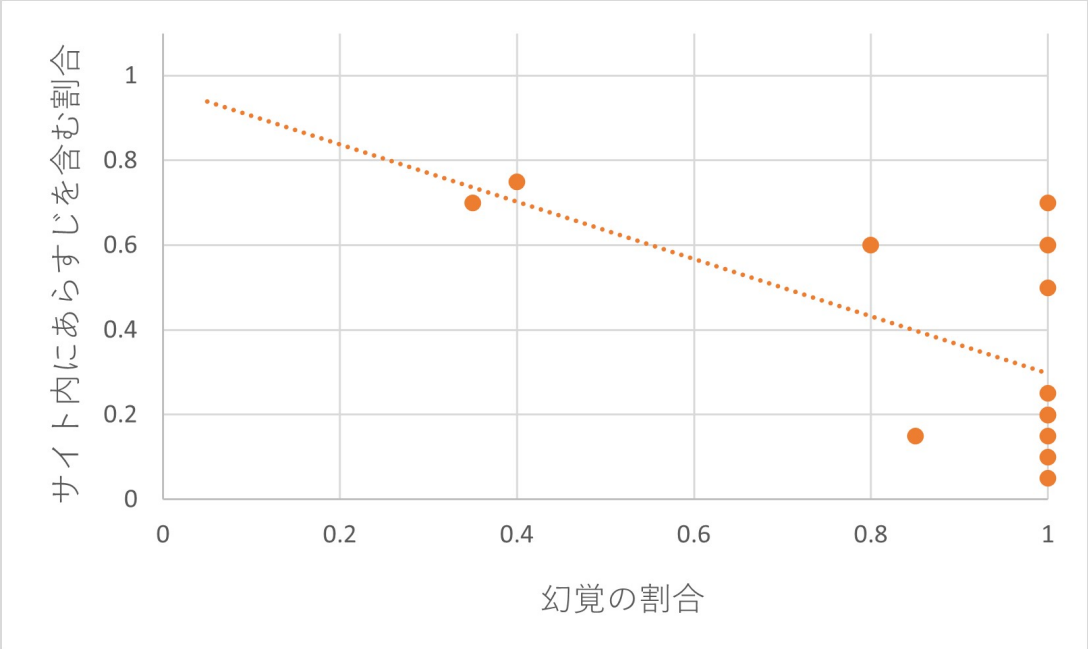

図 3.7: Top20 のサイトのあらすじを含む割合と ChatGPT の幻覚の割合の関係

以上の結果から，Web 上の検索結果の上位サイトにあらすじを含む割合と生成文の幻覚の割合に関しては，ChatGPT の生成文の場合は負の相関関係がみられ，あらすじを含むサイトが多いほど幻覚が少ないが，Bard の生成文の場合は，条件のサイト数を変更しても相関関係は全くみられなかった．

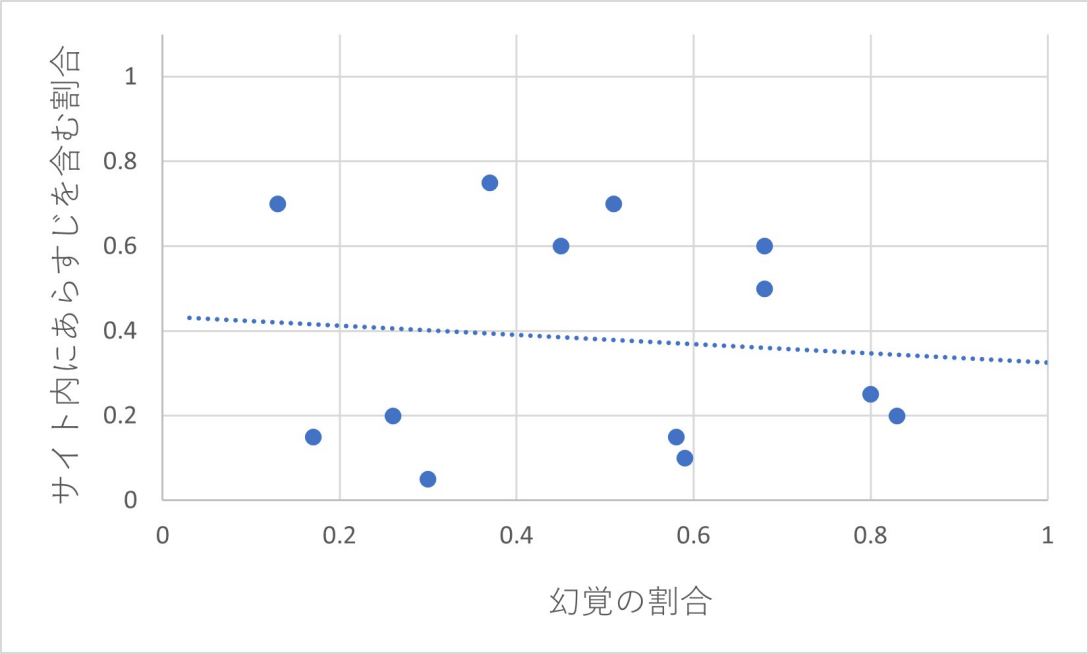

図 3.8: Top20 のサイトのあらすじを含む割合と Bard の幻覚の割合の関係

3.4.3　小説あらすじの特徴について

ChatGPT に関しては，検索件数，検索結果の上位サイトにあらすじを含む割合のどちらにも生成文に幻覚を含む割合と相関関係がみられた．この点は，ChatGPT が Web 上の知識を参照できないため，有名作品の知識しか扱えないことが原因と考えられる．つまり，有名な作品ほど作品の紹介サイトや，小説の情報を扱うサイトが多いため，検索件数は増え，同時にあらすじを紹介するサイト数も増える．その結果，どちらにも相関関係がみられたのではないだろうか．

その一方で Bard に関しては，検索件数には生成文に幻覚を含む割合と相関関係がみられたが，検索結果の上位サイトにあらすじを含む割合とは相関関係がみられなかった．上位サイトにあらすじを含む割合に関して相関がみられない原因として，Bard は Web 上の知識を参照できるため，どの作品であったとしてもある程度の知識を扱えることが考えられる．そして，検索件数が多いほど，より信頼性の高い情報を選定できるため，検索件数との間には相関関係がみられたのではないか．

3.5　辞書的マッチングによる幻覚の有無判定

単純な方法でも幻覚の判定が行えるかを調査するため，辞書的マッチングによる幻覚の有無判定を行った．幻覚が含まれない生成文には，類似する単語が頻出するのではないかという考えに基づいて，同じ単語の出現数をカウントし，幻覚を判定する．

3.5.1　幻覚の判定手法

この手法では，2 つの生成文を用いて幻覚の判定を行う．手順は以下の通りである．

1. 判定に用いる 2 つの生成文を用意する．
2. それぞれの文から janome の形態素解析により，名詞を抽出する．
3. 2 つの文に共通する名詞，片方の文にのみ出現する名詞の数をそれぞれカウントする．
4. 3 でカウントした名詞のうち，2 つの文に共通する名詞の割合と幻覚判定用の閾値を比較する．閾値より共通の名詞の割合が高い場合は幻覚を含まず，閾値より共通の名詞の割合が低い場合は幻覚を含む文章と判定する．

判定に用いる 2 つの生成文は，どちらも幻覚を含む文章，またはどちらも幻覚を含まない真実の文章とする．判定に用いる文の組み合わせと，閾値による判定の詳細については表 3.3 に示す．

表 3.3: 生成文の組み合わせと閾値の判定

	真実と真実の文	真実と幻覚の文	幻覚と真実の文	幻覚と幻覚の文
文 1	真実	真実	幻覚	幻覚
文 2	真実	幻覚	真実	幻覚
閾値より高い	正解	判定しない	判定しない	不正解
閾値より低い	不正解	判定しない	判定しない	正解

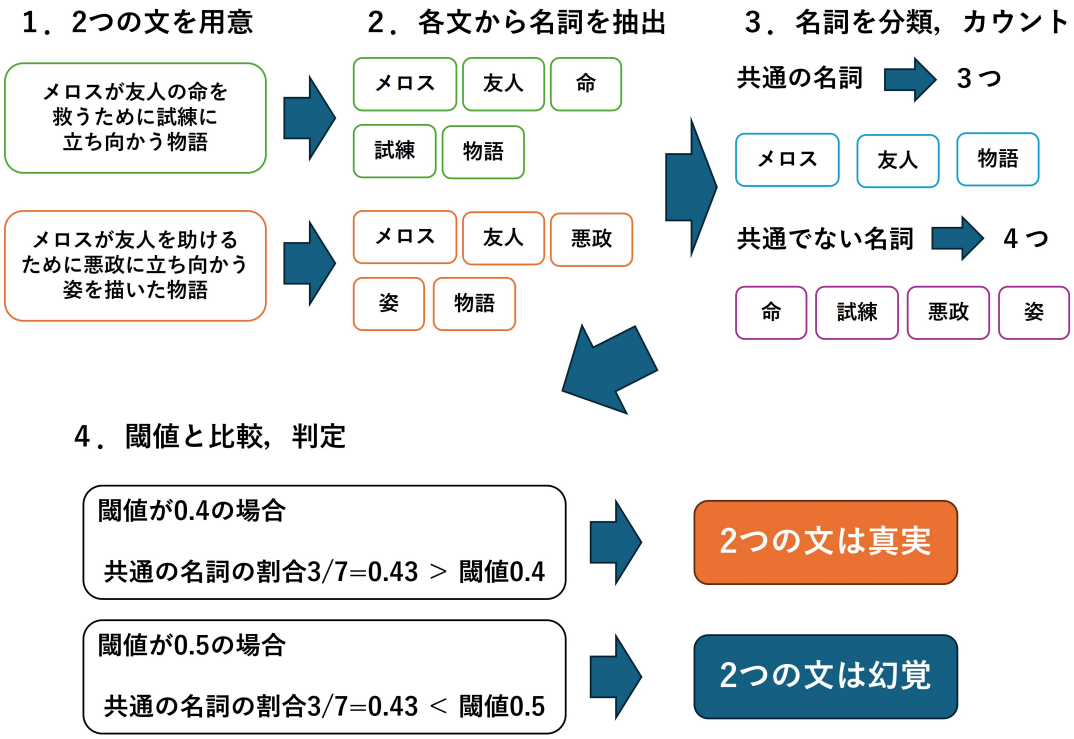

図 3.9: 幻覚の判定手法

3.5.2　幻覚の判定結果

3.5.1 節の手法を用いて，生成文同士の辞書的マッチングによる幻覚の判定を行った．判定のための閾値を 0.01 から 0.01 ずつ増加させた場合の，真実の生成文同士の判定，幻覚の生成文同士の判定の正解率及びそれらの平均の正解率について調査した．生成文については，ChatGPT の 2 つの生成文，Bard の 2 つの生成文，ChatGPT と Bard の生成文の 3 通りについて判定を行った．ChatGPT の生成文を 2 つ用いた場合の幻覚の判定結果を図 3.10 に示す．平均正解率に着目すると，最も高い正解率は 0.7 程度であり，幻覚の判定に成功しているといえる．

Bard の生成文を 2 つ用いた場合の幻覚の判定結果を図 3.11 に示す．平均正解率に着目すると，最も高い正解率は 0.62 から 0.63 程度であり，ChatGPT の生成文を 2 つ用いた場合に成功率は劣るが，幻覚の判定が若干可能であるといえる．

ChatGPT の生成文と Bard の生成文を用いた場合の幻覚の判定結果を図 3.12 に示す．平均正解率に着目すると，平均正解率がほとんどの閾値で 0.5 を下回っており，幻覚の判定が全く行えていない．

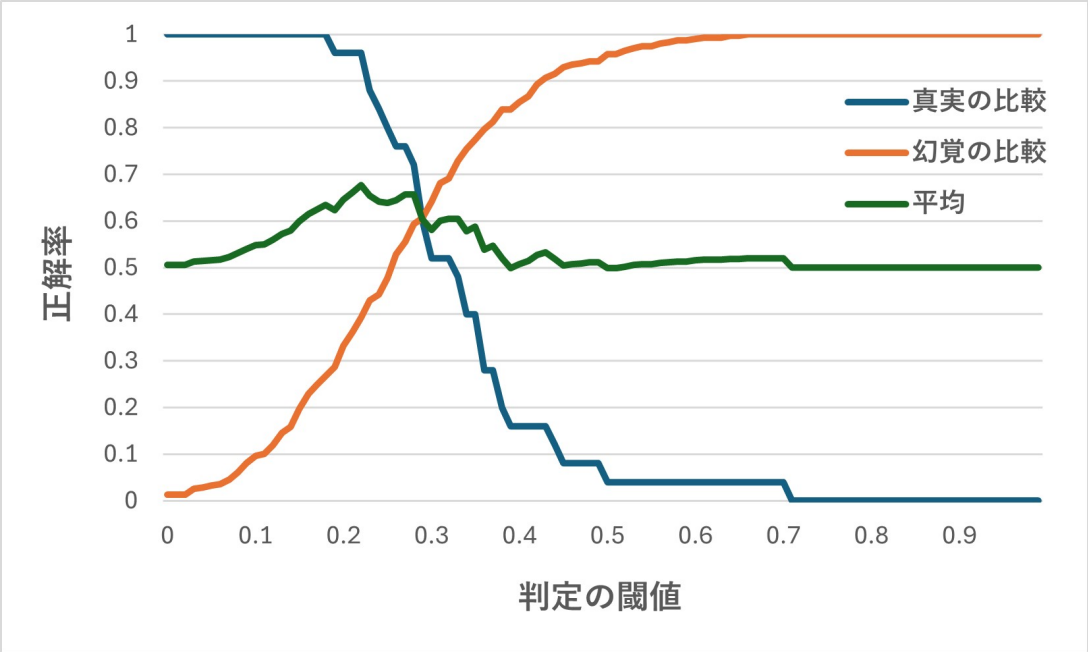

図 3.10: ChatGPT の生成文の比較

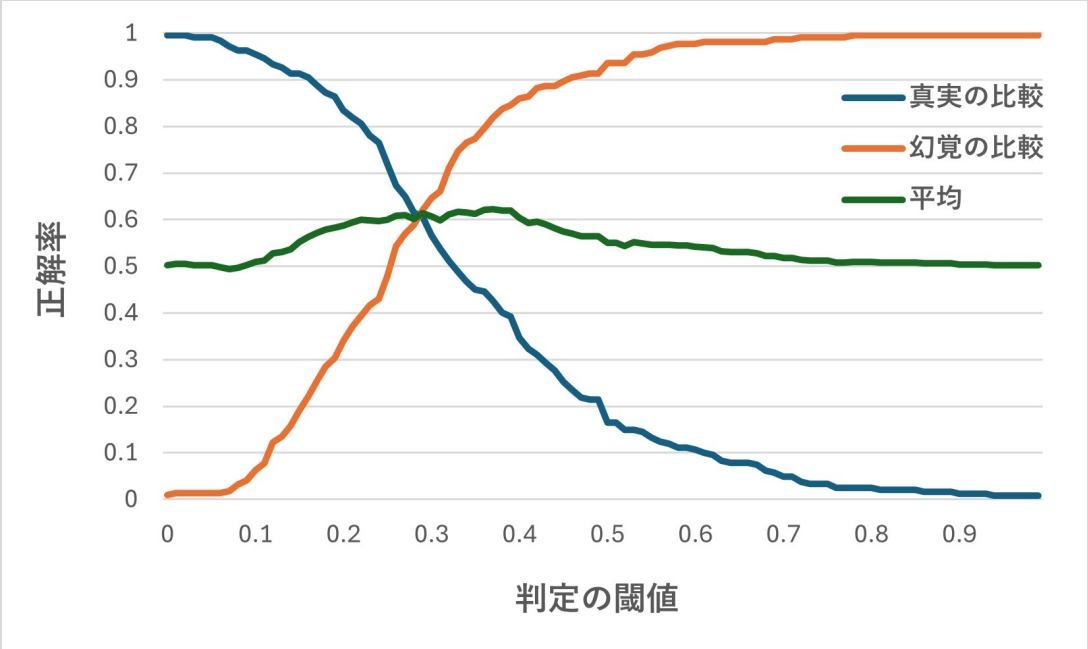

図 3.11: Bard の生成文の比較

以上の結果から，同種の生成モデルを用いた場合，単純な辞書的マッチングによる手法で生成した小説のあらすじの幻覚を判定することがある程度可能であることがわかった．一方で，異なる生成モデルが生成したあらすじについて，辞書的マッチングで幻覚を判定することは不可能であることがわかった．この原因として，各生成モデルの出現する単

語の特徴が異なる点，生成文の長さが異なり，一方の名詞の割合に影響された点が考えられる．よって，同じ意味を持つ単語を共通の名詞と扱うことで一定の改善効果が見込めると考える．

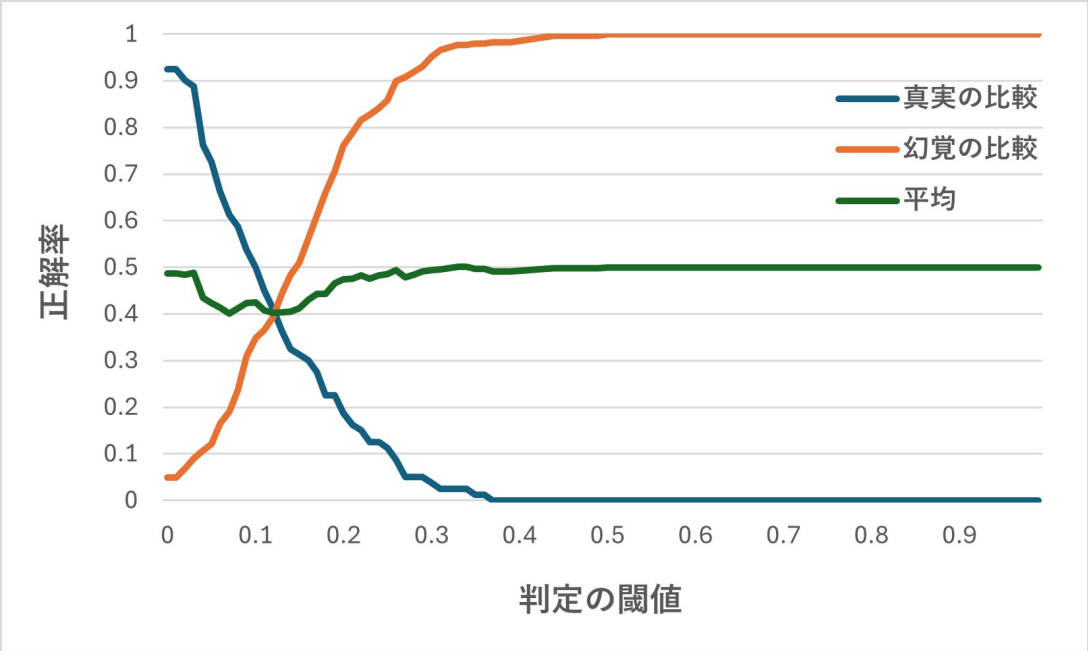

図 3.12: ChatGPT と Bard の生成文の比較

第 4 章

RAG を用いた実験

QA タスク用の RAG を実装し，Chunk の作成時のパラメータ，Retrieval に用いる Embedding モデルを変更することによる Retrieval の結果，回答の生成に与える影響について調査する実験を行った．

4.1 実験の背景・目的

第 3 章の調査の結果から，Bard のような Web 上の検索結果を考慮した出力が可能な LLM はマイナーな小説のあらすじもある程度生成することができることがわかった．そこで本実験では，Bard のように外部の知識を取得して利用できる RAG を用いて，よりマイナーな知識である，小説の内容に関する質問の回答の調査を行う．

RAG は，LLM から得られない知識をデータベースから検索して得るという手法である．このため RAG の研究は主に検索手法を扱ったものが多い．本実験はこのような拡張した手法ではなく，基本的な RAG の検索部分の設定パラメータを調整することで，性能にどの程度の差異が生じるかを調べている．この点から本実験と最も関連が深いのは論文 [11] である．そこでは LLM の検索部の影響を系統的に調査している．また，本実験ではクエリが（マイナーな）小説の内容に関するものであり，LLM 内に回答の知識がないことを前提としている．

4.2　実験用データセット

4.2.1　実験用データベース

青空文庫[*1]に公開されている太宰治の小説「女生徒」,「千代女」本文を，句点で分割してデータベースを作成した．なお，作品名と作者名を識別するため，一文毎に文頭に「小説名（作者名）：」を付与した．実験用データベースの例を表 4.1 に示す．

表 4.1: 実験用データベースの例

”女生徒 (太宰治): パチッと眼がさめるなんて、あれは嘘だ。”
”女生徒 (太宰治): 濁って濁って、そのうちに、だんだん澱粉が下に沈み、少しずつ上澄が出来て、やっと疲れて眼がさめる。”
”女生徒 (太宰治): 朝は、なんだか、しらじらしい。”
”女生徒 (太宰治): 悲しいことが、たくさんたくさん胸に浮かんで、やりきれない。”
”女生徒 (太宰治): いやだ。”
”女生徒 (太宰治): いやだ。”
”女生徒 (太宰治): 朝の私は一ばん醜い。”
”女生徒 (太宰治): 両方の脚が、くたくたに疲れて、そうして、もう、何もしたくない。”
”女生徒 (太宰治): 熟睡していないせいかしら。”
”女生徒 (太宰治): 朝は健康だなんて、あれは嘘。”
”女生徒 (太宰治): 朝は灰色。”
”女生徒 (太宰治): いつもいつも同じ。”
”女生徒 (太宰治): 一ばん虚無だ。”

4.2.2　実験用テストデータ

RAG の評価を行うために，質問，対応する本文，正答からなる評価用のデータセットを作成した．この質問は，太宰治の小説「女生徒」の内容に関するものであり，正答は，質問の正解となる．また，対応する本文は，それぞれの質問に正確に回答するために必要となる本文の文章となる．この質問，対応する本文，正答の組を 15 組用意し，実験のテスト，評価に用いた．本テストデータは付録 A に記した．

[*1] https://www.aozora.gr.jp/

4.3　RAG の設定

本実験の RAG は LangChain を用いて実装した．実験の詳細な設定に関して，以下に示す．また，プログラム本文は付録 B に記した．

4.3.1　Chunk の作成

Chunk の作成には，4.2.1 章の実験用データベースを用いた．Chunk Size は実験ごとに変更し，100 トークンから 1000 トークンまで 100 トークンずつの 10 種類である．Overlap Size も実験ごとに変更し，Chunk Size の 0%，25%，50% の 3 種類である．よって，作成する Chunk は 30 種類となる．また，Chunk の最小単位は小説の本文一文とする．

4.3.2　Index の作成

4.3.1 章で作成した Chunk を埋め込みモデルに与えて Index を作成した．埋め込みモデルには，HuggingFace で提供されている paraphrase-multilingual-mpnet-base-v2[*2]，intfloat の multilingual-e5-large[*3]，OpenAI 社から提供されている API から text-embedding-ada-002[*4]の 3 種類を用いる．

4.3.3　Retrieval

4.3.2 章で作成した Index を用いて Retrieval を行う．具体的には，質問に関連する Chunk を Index から取得し，回答の生成の際に prompt に埋め込んで使用する．今回の実験では，Index から取得する Chunk 数を，質問とのコサイン類似度上位 4 つとした．

[*2] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2
[*3] https://huggingface.co/intfloat/multilingual-e5-large
[*4] https://openai.com/blog/new-and-improved-embedding-model

4.3.4　Generation

回答の生成には，LINE 社が公開しているモデル japanese-large-lm-3.6b-instruction-sft[*5]を使用した．使用する prompt は表 4.2 の通りである．

表 4.2: 生成に用いる prompt

以下のコンテキストを使用して回答を生成してください。
——
コンテキスト: context
——
質問: question
回答:

4.4　実験結果

実験用データセットで作成したデータベースを実装した RAG に与え，評価用データの 15 個の質問に回答させた．それぞれの質問には，4 つの Chunk が Retrieval され，回答生成に用いられた．そして，Retrieval された Chunk に実際に質問に関連する文章が含まれているか，また，回答が正確であるかを評価した．この実験を，各 Chunk Size，各 Overlap Size，各 Embedding モデルに応じて行った．

Retrieval 成功数，回答の成功数に関して，paraphrase-multilingual-mpnet-base-v2 の結果を表 4.3，intfloat/multilingual-e5-large の結果を表 4.4，text-embedding-ada-002 の結果を表 4.5 に示す. [*6]各モデルの Overlap Size ごとの Retrieval 成功数及び回答の成功数の平均値に注目すると，どのモデルに関しても Overlap Size が大きくなるほど，ほとんどの数値が増加している．よって，日本語の小説をデータベースとした RAG の場合，Overlap Size が大きくなるほど，より良い Retrieval，生成結果が得られる可能性があると考えられる．

[*5] `https://huggingface.co/line-corporation/japanese-large-lm-3.6b-instruction-sft`

[*6] OvSz は Overlap Size，CkSz は Chunk Size，RetS は Retrieval の成功数，正答は質問に対する回答の正答数を示す．また，平均値は各 Overlap Size に対する Chunk Size 全体の Retrieval の成功数と回答の正答数の平均値を示す．

表 4.3: paraphrase-multilingual-mpnet-base-v2 の実験結果

OvSz	CkSz	RetS	正答	CkSz	RetS	正答
0%	100	4	2	600	3	1
	200	4	4	700	5	0
	300	6	2	800	2	1
	400	6	3	900	3	1
	500	3	1	1000	5	0
平均値					4.1	1.5
25%	100	4	2	600	3	1
	200	6	4	700	4	2
	300	7	3	800	6	3
	400	5	2	900	0	0
	500	4	2	1000	2	0
平均値					4.1	1.9
50%	100	4	1	600	6	2
	200	7	3	700	5	3
	300	7	4	800	6	1
	400	6	4	900	6	1
	500	6	2	1000	6	2
平均値					5.9	2.3

また，Retrieval 成功数に関して，Overlap Size ごとの比較をグラフで表した．Overlap Size 0% を図 4.1，Overlap Size 25% を図 4.2，Overlap Size 50% を図 4.3 に示す．それぞれのグラフでは，Chunk Size の大きさという観点では規則性は見られず，多くの Chunk Size で緑色の Multilingual-e5-large が良い性能を示しており，赤色の paraphrase-multilingual-mpnet-base-v2 が劣った性能を示している．ここで，今回の実験は限定的な状況のため，一概に Multilingual-e5-large が優秀なモデルであり，paraphrase-multilingual-mpnet-base-v2 が劣ったモデルであるとはいえない．しかしながら，一つのタスクに限定すると，Chunk Size や Overlap Size とは無関係に Embedding モデルの性能差があらわれることがわかった．

表 4.4: intfloat/multilingual-e5-large の実験結果

OvSz	CkSz	RetS	正答	CkSz	RetS	正答
0%	100	7	3	600	9	4
	200	9	5	700	9	3
	300	8	2	800	5	1
	400	7	3	900	10	6
	500	11	6	1000	6	0
平均値					8.1	3.3
25%	100	8	3	600	9	5
	200	9	3	700	8	1
	300	9	5	800	7	4
	400	10	6	900	6	5
	500	9	5	1000	11	1
平均値					8.6	3.8
50%	100	12	4	600	10	6
	200	7	4	700	10	4
	300	10	5	800	8	3
	400	7	4	900	8	3
	500	11	4	1000	8	0
平均値					9.1	3.7

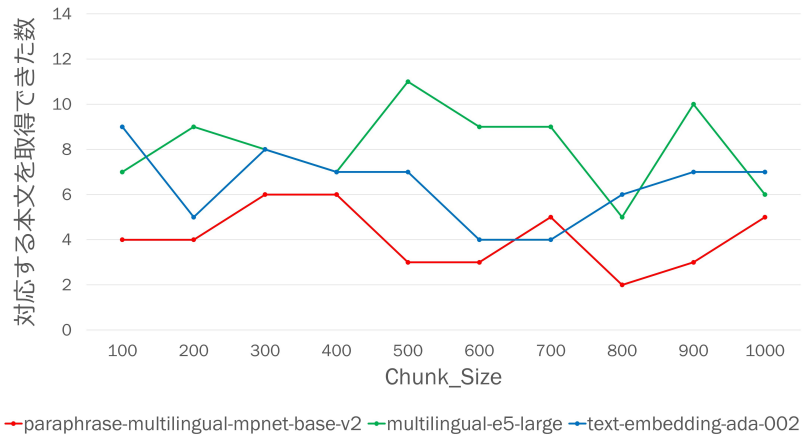

図 4.1: Overlap Size 0% の場合の各モデルの比較

表 4.5: text-embedding-ada-002 の実験結果

OvSz	CkSz	RetS	正答	CkSz	RetS	正答
0%	100	9	5	600	4	2
	200	5	3	700	4	2
	300	8	4	800	6	2
	400	7	2	900	7	2
	500	7	2	1000	7	0
平均値					6.4	2.4
25%	100	10	7	600	5	2
	200	8	5	700	6	1
	300	8	4	800	7	2
	400	5	3	900	4	0
	500	7	3	1000	3	0
平均値					6.3	2.7
50%	100	9	5	600	4	2
	200	7	4	700	6	4
	300	7	4	800	6	2
	400	9	4	900	7	5
	500	7	3	1000	8	0
平均値					7.0	3.3

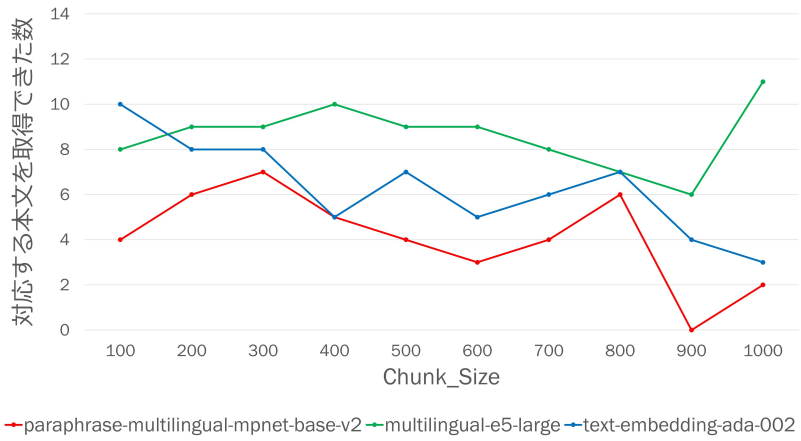

図 4.2: Overlap Size 25% の場合の各モデルの比較

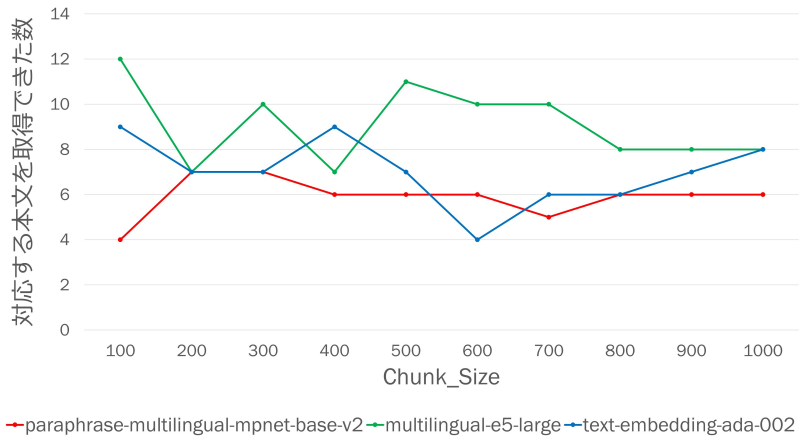

図 4.3: Overlap Size 50% の場合の各モデルの比較

第 5 章

考察

事前学習で得られなかった知識に関して，LLM が正確な回答を生成するには，正しい Retrieval が必要であると考えられる．つまり，RAG においては Retrieval の成功数と質問に対する回答の正答数には，相関関係があると考えられる．そこで，4.4 章で得られた実験結果について，Chunk Size，Overlap Size の組ごとの Retrieval の成功数と回答の正答数について，相関係数を調査し，さらに濃度分布図に表した．作成した濃度分布図を図 5.1 に示す．調査した結果，相関係数は *0.645* であり，Retrieval の成功数と回答の正答数について相関が示された．また，図 5.1 から，視覚的にも相関関係があることがうかがえる．

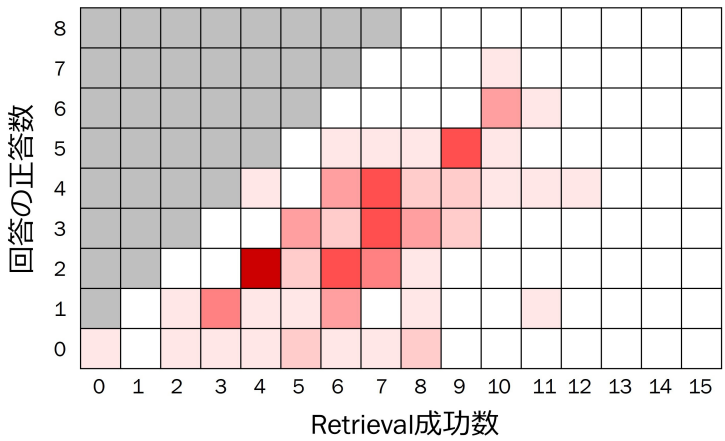

図 5.1: Retrieval 成功数と正答数の関係

次に，テストデータの質問ごとに各モデルの Retrieval 成功数を表にあらわした．その

結果を表 5.1 に示す.[*1] この表から，それぞれの質問についての傾向が読み取れる.

質問 1，2 に関しては，いずれのモデルであってもほとんど Retrieval できていない. その原因の一つとして，Retrieval したい情報の前後が全く異なる文脈であるため，情報が埋もれてしまっていることが考えられる．また，その他の原因として，似通った意味の情報が本文中に多く存在し，上位 4 つに Retrieval できていないということが考えられる.

そして，一部のモデルでは Retrieval が可能だが，一部のモデルではほとんど Retrieval できないというようにモデルによる得手不得手な質問が存在している．例えば，質問 5 は paraphrase-multilingual-mpnet-base-v2 では Retrieval できているが，他の 2 つのモデルでは Retrieval できていない．一方で質問 12 は，intfloat/multilingual-e5-large では Retrieval できているが，他の 2 つのモデルでは Retrieval できていない.

表 5.1: 各質問の Retrieval 成功回数

QzNo.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
mpnet	0	0	3	12	8	9	17	22	22	20	20	1	0	8	0
e5	1	2	13	14	3	28	24	12	29	26	26	21	28	20	11
ada	3	2	11	22	1	21	24	18	24	14	13	2	13	18	11

さらに，各 Chunk Size，各 Overlap Size，各 Embedding モデルでそれぞれの質問が Retrieval できたかどうかを図に示した．その一例として，質問 7 の Retrieval の結果を図 5.2 に示す．この図の赤く囲んだ部分に注目すると，Overlap Size 25% の場合，Chunk Size 600 から 900 にかけて不自然に Retrieval に失敗していることがわかる．よって，クエリに応じて情報が埋めにくい Chunk Size，Overlap Size が存在すると考えることができる.

[*1] QzNo. は質問番号を示す．mpnet，e5，ada は各 Embedding モデルを示し，それぞれ paraphrase-multilingual-mpnet-base-v2，intfloat/multilingual-e5-large，text-embedding-ada-002 である.

Chunk Size	Overlap Size								
	multilingual-mpnet			intfloat/e5-large			embedding-ada-002		
	0%	25%	50%	0%	25%	50%	0%	25%	50%
Chunk_size 100				●	●	●	●	●	●
Chunk_size 200	●	●	●	●	●	●		●	●
Chunk_size 300	●	●	●	●	●	●	●	●	●
Chunk_size 400		●	●		●	●		●	
Chunk_size 500		●	●	●	●	●	●	●	●
Chunk_size 600			●	●		●	●		●
Chunk_size 700				●		●	●		●
Chunk_size 800			●	●			●		●
Chunk_size 900	●		●	●		●	●	●	●
Chunk_size 1000	●	●	●	●	●	●	●	●	●

図 5.2: 質問 7 の Retrieval の結果

第 6 章

結論

本研究では調査として，日本語の小説のあらすじを LLM に生成させ，幻覚が発生するか否かを評価した．それぞれの小説の Web 上の検索件数，上位のサイト内にあらすじを含む割合について，幻覚を含む生成文の割合との相関関係を調査した．その結果，ChatGPT の生成文には，検索件数，上位サイト内のあらすじ含有率のどちらにも負の相関関係がみられ，Bard の生成文には，検索件数のみに相関関係がみられた．また，単純な辞書的マッチングによるあらすじの幻覚判定は，同一の LLM の生成文同士の判定に対して一定の効果が認められた．

加えて本研究では，LangChain から作成した RAG を用いて，日本語の小説として太宰治の作品をデータベースとして Chunk Size，Overlap Size，Embedding モデルをそれぞれ変更した際の影響について調査を行った．そして，Retrieval の成功数，小説の内容に関した質問に対する回答の正答数という観点から評価を行った．その結果，Chunk Size に決まった規則性は見られないこと，Overlap Size が大きいほど RAG の性能が向上すること，Embedding モデル間の性能の差は Overlap を変更しても変わらないことがわかった．

謝辞

本研究は国立国語研究所の共同研究プロジェクト「テキスト読み上げのための読みの曖昧性の分類と読み推定タスクのデータセットの構築」及び JSPS 科研費 23K11212 の助成を受けています．また，本研究を進めるにあたって，多くのご指導を頂いた指導教員の新納浩幸教授に感謝いたします．そして，日常の議論を通して多くの知識，示唆を頂いた新納研究室の皆様に感謝します．

参考文献

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[3] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[4] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[7] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, Vol. 55, No. 12, p. 1–38, March 2023.

[8] Nouha Dziri, Andrea Madotto, Osmar Zaiane, and Avishek Joey Bose. Neural path hunter: Reducing hallucination in dialogue systems via path grounding, 2021.

[9] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022.

[10] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

[11] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation, 2023.

付録

A　実験用テストデータ

4.2.2 章で述べた RAG の実験に用いた評価用のテストデータを表 A.1,A.2 に示す.

B　RAG のプログラム

LangChain を用いた RAG のソースコードを B.1 に示す.

ソースコード B.1: rag.py

```
import torch

from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain
from langchain.agents import ZeroShotAgent, Tool, AgentExecutor,
    load_tools

from langchain.llms.huggingface_pipeline import HuggingFacePipeline
from transformers import AutoModelForCausalLM, AutoTokenizer,
    pipeline

from langchain.document_loaders import TextLoader
from langchain.indexes import VectorstoreIndexCreator
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.text_splitter import TextSplitter
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.text_splitter import CharacterTextSplitter

from langchain.embeddings import OpenAIEmbeddings

import os
```

```
20
21  # 使用する API KEY に書き換える
22  os.environ["OPENAI_API_KEY"] = "api-key"
23
24  # 利用可能であれば cuda を利用する
25  device_num = -1 # CPU
26  if torch.cuda.is_available():
27    device_num = 0 # cuda:0
28
29  # 生成モデルとして LINE 社のモデルを利用する
30  model_id = "line-corporation/japanese-large-lm-3.6b-instruction-sft"
31
32  # データベースとして青空文庫の太宰治の「女生徒」,「千代女」の本文を使用
        する
33  loader = TextLoader('./drive/MyDrive/rag/aozora_sentence.txt',
        encoding="utf-8")
34
35  # Embedding モデルを選択し，Chunk Size ，Overlap Size を設定する
36  #index = VectorstoreIndexCreator(embedding= HuggingFaceEmbeddings(
        model_name="paraphrase-multilingual-mpnet-base-v2"),text_splitter
         = CharacterTextSplitter(separator = "\n", chunk_size = 100,
        chunk_overlap = 0)).from_loaders([loader])
37  index = VectorstoreIndexCreator(embedding= OpenAIEmbeddings(),
        text_splitter = CharacterTextSplitter(separator = "\n",
        chunk_size = 100, chunk_overlap = 0)).from_loaders([loader])
38  #index = VectorstoreIndexCreator(embedding= HuggingFaceEmbeddings(
        model_name="intfloat/multilingual-e5-large"),text_splitter =
        CharacterTextSplitter(separator = "\n", chunk_size = 100,
        chunk_overlap = 0)).from_loaders([loader])
39
40  # 生成モデルを設定する
41  tokenizer = AutoTokenizer.from_pretrained(model_id)
42  model = AutoModelForCausalLM.from_pretrained(model_id)
43  pipe = pipeline(
44      "text-generation", model=model, tokenizer=tokenizer,
            max_new_tokens=512, device=device_num, torch_dtype=torch.
            float16
45  )
46  hf = HuggingFacePipeline(pipeline=pipe)
47
48  from langchain.chains import RetrievalQA
```

```
49
50  # 質問用のリストを用意し，質問を追加する
51  quiz = []
52
53  quiz.append("太宰治の小説である「女生徒」の主人公のいとこの名前は？")
54  quiz.append("太宰治の小説である「女生徒」の主人公の一番好きな子の名前
        は？")
55  quiz.append("太宰治の小説である「女生徒」の主人公の家に来たお客さんの名
        前は？")
56  quiz.append("太宰治の小説である「女生徒」に登場する美しい青色の似合う先
        生の名前は？")
57  quiz.append("太宰治の小説である「女生徒」で描かれているのは何月何日？")
58  quiz.append("太宰治の小説である「女生徒」で「私」が可愛がったペットの名
        前は？")
59  quiz.append("太宰治の小説である「女生徒」で主人公が今年初めて食べたもの
        は？")
60  quiz.append("太宰治の小説である「女生徒」でお昼御飯のときに出たお話は？
        ")
61  quiz.append("太宰治の小説である「女生徒」で午後の図画の時間に「私」をモ
        デルにしたのは？")
62  quiz.append("太宰治の小説である「女生徒」の主人公が机の上に飾っている花
        の名前は？")
63  quiz.append("太宰治の小説である「女生徒」で主人公が朝、掃除をしながら
        歌った曲は？")
64  quiz.append("太宰治の小説である「女生徒」で、「私」が新聞で一番楽しいと
        感じている内容は？")
65  quiz.append("太宰治の小説である「女生徒」で、主人公がおととしの夏休みに
        遊びに行った場所は？")
66  quiz.append("太宰治の小説である「女生徒」で「私」が見たいと思っている映
        画は？")
67  quiz.append("太宰治の小説である「女生徒」で、「私」の隣の席の人の名前
        は？")
68
69  # prompt のテンプレートを作成する
70  template = """以下のコンテキストを使用して回答を生成してください。
71
72  -----コンテキスト
73  : {context}
74  -----質問
75  : {question}回答
76  : """
77
78  # 各質問に答える
79  for i in range(15):
80    print("\nquiz" + str(i+1))
81
```

```
82      # 質問に関連する文書を上位 4 つ取得する
83      results = index.vectorstore.similarity_search(quiz[i], k=4)
84      context = "\n".join([document.page_content for document in results
            ])
85      print(f"{context}")
86
87      # 質問用の prompt を作成する
88      prompt = PromptTemplate(template=template, input_variables=["
            context", "question"]).partial(context=context)
89
90      # チェーンを作成し，質問に答える
91      llm_chain = LLMChain(prompt=prompt, llm=hf)
92      print(llm_chain.run(quiz[i]))
```

表 A.1: テスト用の質問と対応する本文と正答 (No.1～No.10)

No.	質問の内容	対応する本文	正答
1	太宰治の小説である「女生徒」の主人公のいとこの名前は何ですか？	一通は、私へ、いとこの順二さんから。	順二
2	太宰治の小説である「女生徒」の主人公の一番好きな子の名前は何ですか？	私は、親類中で、いや、世界中で、一ばん新ちゃんを好きだ。	新ちゃん
3	太宰治の小説である「女生徒」の主人公の家に来たお客さんの名前は何ですか？	きょうのお客様は、ことにも憂うつ。大森の今井田さん御夫婦に、ことし七つの良夫さん。先刻、今井田が来ていたときに、お母さんを、こっそり恨んだことを、恥ずかしく思う。今井田さん、おかえりになる。	今井田さん御夫妻
4	太宰治の小説である「女生徒」に登場する美しい青色の似合う先生の名前は何ですか？	けさの小杉先生は綺麗。私の風呂敷みたいに綺麗。美しい青色の似合う先生。山中、湖畔の古城に住んでいる令嬢、そんな感じがある。	小杉先生
5	太宰治の小説である「女生徒」で描かれているのは何月何日の話ですか？	けさから五月、そう思うと、なんだか少し浮き浮きして来た。	5 月 1 日
6	太宰治の小説である「女生徒」で「私」が可愛がったペットの名前は何ですか？	ジャピイと、カア（可哀想な犬だから、カアと呼ぶんだ）と、二匹もつれ合いながら、走って来た。二匹をまえに並べて置いて、ジャピイだけを、うんと可愛がってやった。縁側に腰かけて、ジャピイの頭を撫でてやりながら、目に浸みる青葉を見ていると、情なくなって、土の上に坐りたいような気持になった。あんまり面白くて、この木をゆすぶって、ポタポタ落としたら、ジャピイ夢中になって食べはじめた。ばかなやつ。茱萸を食べる犬なんて、はじめてだ。急に、歯ぎしりするほどジャピイを可愛くなっちゃって、シッポを強く掴むと、ジャピイは私の手を柔かく噛んだ。	ジャピイ
7	太宰治の小説である「女生徒」で主人公が今年初めて食べたものは何ですか？	ことし、はじめて、キウリをたべる。	キウリ
8	太宰治の小説である「女生徒」でお昼御飯のときに出たお話は何ですか？	お昼御飯のときは、お化け話が出る。ヤスベエねえちゃんの、一高七不思議の一つ、「開かずの扉」には、もう、みんな、きゃあ、きゃあ。それからまた、ひとしきり恐怖物語にみなさん夢中。それから、これは怪談ではないけれど、「久原房之助」の話、おかしい、おかしい。	お化け話，「開かずの扉」，恐怖物語，久原房之介

表 A.2: テスト用の質問と対応する本文と正答 (No.9〜No.15)

No.	質問の内容	対応する本文	正答
9	太宰治の小説である「女生徒」で午後の図画の時間に「私」をモデルにしたのは誰ですか？	午後の図画の時間には、皆、校庭に出て、写生のお稽古。伊藤先生は、どうして私を、いつも無意味に困らせるのだろう。きょうも私に、先生ご自身の絵のモデルになるよう言いつけた。三十分間だけ、モデルになってあげることを承諾する。すこしでも、人のお役に立つことは、うれしいものだ。けれども、伊藤先生と二人で向かい合っていると、とても疲れる。それからずいぶん自分を買いかぶっているのですよ、ああ、こんな心の汚い私をモデルにしたりなんかして、先生の画は、きっと落選だ。美しいはずがないもの。いけないことだけれど、伊藤先生がばかに見えてしようがない。	伊藤先生
10	太宰治の小説である「女生徒」の主人公が机の上に飾っている花の名前は？	お部屋へ戻って、机のまえに坐って頬杖つきながら、机の上の百合の花を眺める。	百合
11	太宰治の小説である「女生徒」で主人公が朝、掃除をしながら歌った曲は？	お掃除しながら、ふと「唐人お吉」を唄う。普段、モオツァルトだの、バッハだのに熱中しているはずの自分が、無意識に、「唐人お吉」を唄ったのが、面白い。蒲団を持ち上げるとき、よいしょ、と言ったり、お掃除しながら、唐人お吉を唄うようでは、自分も、もう、だめかと思う。	唐人お吉
12	太宰治の小説である「女生徒」で、「私」が新聞で一番楽しいと感じている内容は？	新聞では、本の広告文が一ばんたのしい。	新聞の本の広告文
13	太宰治の小説である「女生徒」で、主人公がおととしの夏休みに遊びに行った場所は？	おととしの夏休みに、北海道のお姉さんの家へ遊びに行ったときのことを思い出す。苫小牧のお姉さんの家は、海岸に近いゆえか、始終お魚の臭いがしていた。	北海道，苫小牧
14	太宰治の小説である「女生徒」で「私」が見たいと思っている映画は？	お風呂から上がって、私と二人でお茶を飲みながら、へんにニコニコ笑って、お母さん何を言い出すかと思ったら、「あなたは、こないだから『裸足の少女』を見たい見たいと言ってたでしょう？　そんなに行きたいなら、行ってもよござんす。そのかわり、今晩は、ちょっとお母さんの肩をもんで下さい。働いて行くのなら、なおさら楽しいでしょう？」もう私は嬉しくてたまらない。「裸足の少女」という映画も見たいとは思っていたのだが、このごろ私は遊んでばかりいたので、遠慮していたのだ。	「裸足の少女」
15	太宰治の小説である「女生徒」で、「私」の隣の席の人の名前は？	キン子さんは、全く無性格みたいで、それゆえ、女らしさで一ぱいだ。学校で私と席がお隣同士だというだけで、そんなに私は親しくしてあげているわけでもないのに、お寺さんのほうでは、私のことを、あたしの一ばんの親友です、なんて皆に言っている。	キン子