検索拡張生成 知識集約型NLPタスク

パトリック•ルイス† ‡、イーサン•ペレス、
アレクサンドラ•ピクトゥス†、ファビオ•ペトローニ†、ウラジミール•カルプキン†、ナマン•ゴヤル†、ハインリヒ•キュトラー†、
マイク•ルイス†、ウェンタウ•イー†、ティム•ロックテッシェル† ‡、セバスチャン•リーデル† ‡、ドゥウェ•キーラ†
† Facebook AI Research; ‡ユニバーシティ•カレッジ•ロンドン; ニューヨーク大学;
plewis@fb.com

抽象的な

大規模な事前学習済み言語モデルは事実の知識を蓄積できることが示されている
パラメータを最適化し、下流のNLPタスクで微調整すると最先端の結果を達成します。しかし、知識にアクセスして正確に操作する能力はまだ限られており、したがって知識集約型タスクでは、そのパフォーマンスは
タスク固有のアーキテクチャに遅れをとっています。さらに、
意思決定と世界知識の更新は未解決の研究課題である。明示的な非パラメトリックモデルへの
微分可能なアクセスメカニズムを備えた事前学習済みモデル
メモリはこれまで抽出下流タスクについてのみ研究されてきた。
検索強化型生成のための汎用微調整レシピを探る
（RAG）は、言語生成のために事前に訓練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す。パラメトリックメモリが
メモリは事前学習済みのseq2seqモデルであり、非パラメトリックメモリは密な
事前学習済みのニューラルリトリーバーでアクセスしたWikipediaのベクトルインデックス。2つのRAG定式を比較します。1つは同じ検索された文章を条件とします。
生成されたシーケンス全体にわたって、そして異なるパッセージを使用できる別のもの
トークンごとに。幅広い知識集約型NLPタスクでモデルを微調整して評価し、3つのオープンドメインQAタスクで最先端の技術を確立しました。
パラメトリックseq2seqモデルとタスク固有の取得と抽出を上回る
アーキテクチャ。言語生成タスクでは、RAGモデルは
最先端のパラメトリックのみの言語よりも、より具体的で、多様で、事実に基づいた言語です。
seq2seq ベースライン。

1 はじめに

事前学習済みのニューラル言語モデルは、データからかなりの量の詳細な知識を学習できることが示されている[47]。パラメータ化された
暗黙の知識ベース[51, 52]。この発展はエキサイティングですが、このようなモデルには欠点もあります。メモリを簡単に拡張または修正できない、
予測が不正確になり、「幻覚」を引き起こす可能性がある[38]。パラメトリックと
非パラメトリック記憶（すなわち、検索ベースの記憶） [20, 26, 48]は、これらの問題のいくつかに対処することができる。
知識を直接修正•拡張することができ、アクセスした知識を
検査され解釈された。REALM [20]とORQA [31]は、最近導入された2つのモデルであり、
マスク言語モデル[8]と微分可能リトリーバーを組み合わせた研究は有望な結果を示している。

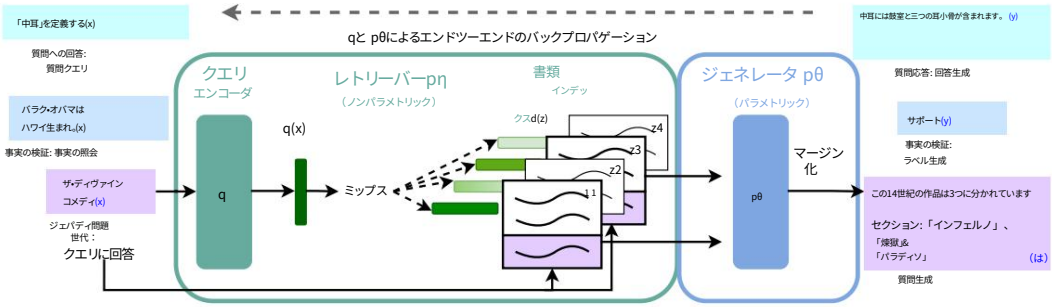

図 1: 私たちのアプローチの概要。事前トレーニング済みのリトリーバー (クエリ エンコーダー + ドキュメントインデックス) と事前トレーニング済みの seq2seq モデル (ジェネレーター) を組み合わせ、エンドツーエンドで微調整します。クエリ x に対して、最大内積探索 (MIPS) を使用して、上位 K 個のドキュメントziを検索します。 最終的な予測y については、 z を潜在変数として扱い、異なるドキュメントが与えられた場合の seq2seq 予測を周辺化します。

しかし、これまではオープンドメインの抽出型質問応答しか研究してきませんでした。ここでは、パラメトリックとノンパラメトリックのハイブリッドメモリを「NLP の主力」であるシーケンスツーシーケンス (seq2seq) モデルに導入します。

私たちは、検索拡張生成 (RAG) と呼ばれる汎用微調整アプローチを通じて、事前トレーニング済みのパラメトリック メモリ生成モデルに非パラメトリック メモリを付与します。

我々は、パラメトリックメモリが事前トレーニング済みの seq2seq トランスフォーマーであり、ノンパラメトリックメモリが事前トレーニング済みのニューラルリトリーバーでアクセスされる Wikipedia の稠密ベクトルインデックスである RAG モデルを構築します。これらのコンポーネントを、エンドツーエンドでトレーニングされた確率モデルに組み合わせます (図 1)。リトリーバー (Dense Passage Retriever [26]、以下 DPR) は、入力に応じて条件付けされた潜在文書を提供し、seq2seq モデル (BART [32])は、これらの潜在文書を入力とともに条件付けして出力を生成します。潜在文書は、出力ごと(すべてのトークンに対して同じ文書が使用されていると想定) またはトークンごと (異なるトークンに対して異なる文書が使用されている)のいずれかで、トップ K 近似を使用してマージナライズされます。T5 [51]や BART と同様に、RAG は任意の seq2seq タスクで微調整でき、ジェネレーターとリトリーバーの両方が共同で学習されます。

これまでにも、メモリネットワーク[64, 55]、スタック拡張ネットワーク[25] 、メモリ層[30] など、特定のタスクのためにゼロからトレーニングされた非パラメトリックメモリでシステムを強化するアーキテクチャを提案する広範な研究が行われてきました。対照的に、私たちはパラメトリックメモリコンポーネントと非パラメトリックメモリコンポーネントの両方が事前にトレーニングされ、広範な知識が事前にロードされている設定を検討します。重要なのは、事前トレーニングされたアクセスメカニズムを使用することで、追加のトレーニングなしで知識にアクセスする機能が存在することです。

私たちの研究結果は、知識集約型タスク（外部の知識源にアクセスしなければ人間が実行することは期待できないタスク)において、パラメトリックメモリとノンパラメトリックメモリを生成と組み合わせることの利点を強調しています。私たちのRAGモデルは、オープンなNatural Questions [29]、 WebQuestions [3] 、CuratedTrec [2]で最先端の結果を達成し、 TriviaQA [24]では特殊な事前トレーニング目標を使用する最近のアプローチを大幅に上回っています。これらは抽出タスクであるにもかかわらず、制約のない生成は以前の抽出アプローチよりも優れていることがわかりました。

知識集約型生成については、MS-MARCO [1]と Jeopardy 問題生成の実験を行い、 BART ベースラインよりも事実に基づいた、具体的で多様な応答をモデルが生成することを発見しました。FEVER [56] の事実検証では、強力な検索監視を使用する最先端のパイプライン モデルの 4.3% 以内の結果を達成しました。最後に、世界の変化に応じてモデルの知識を更新するために、ノンパラメトリック メモリを置き換えることができることを実証しました。1

2つの方法

我々は、入力シーケンスxを使用してテキスト文書zを検索し、それをターゲットシーケンスyを生成する際の追加のコンテキストとして使用するRAGモデルを調査します。図1に示すように、モデルは2つのコンポーネントを活用します。(i)クエリxが与えられた場合にテキストパッセージの分布（上位K個を切り捨てた)を返す、パラメータηを持つリトリーバーpη (z |x)と、(ii)パラメータ化されたジェネレーターpθ(yi |x, z, y1:i－1)です。

1RAGを使った実験を実行するコードは、HuggingFace Transformers Library [66]の一部としてオープンソース化されており、 https://github.com/huggingface/transformers/blob/master/examples/rag/で見つけることができます。RAGモデルのインタラクティブなデモはhttps://huggingface.co/rag/で見つけることができます。

θは、前のi－1個のトークンy1:i－1、元の入力x、および取得したパッセージzのコンテキストに基づいて現在のトークンを生成します。

リトリーバーとジェネレーターをエンドツーエンドでトレーニングするために、取得したドキュメントを潜在変数として扱います。
我々は、生成されたテキストの分布を生成するために、異なる方法で潜在文書を周辺化する 2 つのモデルを提案します。1 つのアプローチである RAG-Sequence では、モデルは同じ文書を使用して各ターゲット トークンを予測します。2 番目のアプローチである RAG-Token では、異なる文書に基づいて各ターゲット トークンを予測できます。以下では、両方のモデルを正式に紹介し、 pηおよびpθコンポーネント、およびトレーニングとデコードの手順について説明します。

2.1 モデル

RAGシーケンスモデルRAGシーケンスモデルは、同じ検索された文書を使用して完全なシーケンスを生成します。技術的には、検索された文書を単一の潜在変数として扱い、トップK近似を介してseq2seq確率p(y|x)を取得するために周辺化されます。具体的には、上位Kの文書がリトリーバーを使用して検索され、ジェネレーターは各文書の出力シーケンス確率を生成し、次に周辺化されます。

$$\text{pRAGシーケンス}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_{i=1}^{N} p_\theta(y_i|x,z,y_{1:i-1})$$

RAG トークン モデルRAG トークン モデルでは、ターゲット トークンごとに異なる潜在ドキュメントを描画し、それに応じてマージナライズすることができます。これにより、ジェネレーターは回答を生成する際に複数のドキュメントからコンテンツを選択できます。具体的には、リトリーバーを使用して上位 K 個のドキュメントが取得され、ジェネレーターは各ドキュメントの次の出力トークンの分布を生成してからマージナライズし、次の出力トークンでプロセスを繰り返します。正式には、次のように定義します。

$$\text{pRAGトークン}(y|x) \approx \prod_{i=1}^{N} \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y_i|x,z,y_{1:i-1})$$

最後に、ターゲット クラスを長さ 1 のターゲット シーケンスと見なすことで、RAG をシーケンス分類タスクに使用できることに注意してください。この場合、RAG シーケンスと RAG トークンは同等になります。

2.2 レトリーバー：DPR

検索コンポーネントpη(z|x)はDPR [26]に基づいています。DPRはバイエンコーダアーキテクチャに従います: $d(z) = \text{BERT}_d(z)$,

$$p_\eta(z|x) \propto \exp d(z)\, q(x) \text{であり、} \quad q(x) = \text{BERT}_q(x)$$

d(z) はBERTBASEドキュメント エンコーダ[8]によって生成されたドキュメントの稠密な表現であり、 q(x) は同じく BERTBASE に基づくクエリ エンコーダによって生成されたクエリ表現です。事前確率 pη(z|x) が最も高いk 個のドキュメントzのリストであるtop-k(pη(·|x))を計算することは、最大内積探索 (MIPS) 問題であり、近似的に線形時間未満で解決できます[23]。DPRの事前トレーニング済みバイエンコーダを使用して、リトリーバーを初期化し、ドキュメント インデックスを構築します。このリトリーバーは、TriviaQA [24]の質問とNatural Questions [29] への回答を含むドキュメントを検索するようにトレーニングされました。ドキュメント インデックスをノンパラメトリック メモリと呼びます。

2.3 ジェネレータ: BART

ジェネレーターコンポーネントpθ(yi |x, z, y1:i－1)は、任意のエンコーダー-デコーダーを使用してモデル化できます。私たちは、400Mのパラメーターを持つ事前トレーニング済みのseq2seqトランスフォーマー[58]であるBART-large [32]を使用します。BARTから生成するときに入力xと取得されたコンテンツzを組み合わせるには、それらを単純に連結します。BARTは、ノイズ除去目的とさまざまなノイズ関数を使用して事前トレーニングされています。これは、さまざまな生成タスクで最先端の結果を取得しており、同等のサイズのT5モデル[32]よりも優れています。以降、BARTジェネレーターパラメーターθをパラメトリックメモリと呼びます。

2.4 トレーニング

どのような文書が検索されるべきかを直接指示することなく、検索コンポーネントと生成コンポーネントを共同でトレーニングします。
入力/出力ペア（xj 、 yj)の微調整トレーニングコーパスが与えられた場合、

周辺対数尤度を確率的に最小化する、 Adam [28]による勾配降下法です。トレ じ − log p(yj |xj)は、各ターゲットの負の
ーニング中にドキュメントエンコーダBERTdを更新すると、REALMが事前トレーニング中に行うようにドキュメントインデックスを定期的に更新する必要があるため、コストがかかります[20]。
このステップは強力なパフォーマンスには必要ないと判断し、ドキュメント エンコーダー (およびインデックス) を固定したまま、クエリ エンコーダー BERTqと BART ジェネレーターのみを微調整します。

2.5 デコード

テスト時に、RAG-Sequence と RAG-Token では、 arg maxy p(y|x) を近似するための異なる方法が必要です。

RAGトークンRAGトークンモデルは、標準的な自己回帰seq2seq生成器として考えることができます。 (yi |x, y1:i − 1) = pη(zi |x)pθ(yi |x, zi , y1:i − 1)遷移確
は、p θ 率: p θ z∈top-k(p(· |x)) (yi |x, y1:i − 1) を標準的なビームデコーダーに挿入します。 デコードするに

RAGシーケンスRAG シーケンスの場合、尤度p(y|x) は従来のトークンごとの尤度に入らないため、単一のビーム検索では解決できません。代わりに、各ドキュメント z に対してビーム検索を実行し、各仮説をpθ(yi |x, z, y1:i − 1) を使用してスコア付けします。これにより、仮説Yのセットが生成されますが、その一部はすべてのドキュメントのビームに出現していない可能性があります。仮説yの確率を推定するには、 yがビームに出現しない各ドキュメントzに対して追加の順方向パスを実行し、ジェネレータの確率にpη(z|x)を乗算してから、マージナルのビーム全体の確率を合計します。このデコード手順を「徹底的なデコード」と呼びます。出力シーケンスが長くなると、 |Y |が大きくなり、多くの順方向パスが必要になる場合があります。より効率的なデコードを行うには、さらに近似してpθ(y|x, zi) ≈ 0にすることができます。ここで、 y はx、 ziからのビーム検索中に生成されません。

これにより、候補セットYが生成された後、追加のフォワード パスを実行する必要がなくなります。このデコード手順を「高速デコード」と呼びます。

3つの実験

我々は、幅広い知識集約型タスクで RAG の実験を行っています。すべての実験で、ノンパラメトリックな知識ソースとして単一の Wikipedia ダンプを使用しています。Lee ら[31]およびKarpukhin ら[26] に従い、 2018 年 12 月のダンプを使用しています。各 Wikipedia 記事は、 100 語ずつのばらばらのチャンクに分割され、合計 2100 万のドキュメントが作成されます。ドキュメント エンコーダーを使用して各ドキュメントの埋め込みを計算し、FAISS [23]と階層的ナビゲート可能なスモール ワールド近似を使用して単一の MIPS インデックスを構築し、高速検索を実現しています[37]。トレーニング中は、各クエリの上位 k 個のドキュメントを取得します。トレーニングではk ∈ {5, 10} を考慮し、開発データを使用してテスト時間にはkを設定しました。次に、各タスクの実験の詳細について説明します。

3.1 オープンドメイン質問応答

オープンドメインの質問応答（QA)は、知識集約型タスクの重要な実世界アプリケーションであり、一般的なテストベッドです[20]。質問と回答を入力と出力のテキストのペア（x、y)として扱い、回答の負の対数尤度を直接最小化することで RAG をトレーニングします。RAG を、検索されたドキュメントから抽出された範囲で回答が抽出され、主に非パラメトリックな知識に依存する一般的な抽出型 QA パラダイム[5、7、31、26]と比較します。また、RAG のように回答を生成しますが、検索を利用せず、代わりに純粋にパラメトリックな知識に依存する「クローズドブックQA」アプローチ[52]とも比較します。一般的なオープンドメイン QA データセットとして、Natural Questions（NQ） [29]、 TriviaQA（TQA） [24]、 WebQuestions（WQ） [3] 、CuratedTrec（CT） [2]の 4 つを検討します。 CTとWQは小さいため、DPR [26]に従い、CTモデルとWQモデルをNQ RAG モデルで初期化します。先行研究[31, 26]と同じトレーニング/開発/テスト分割を使用し、Exact Match (EM)スコアを報告します。TQAについては、T5 [52]と比較するために、TQA Wikiテストセットでも評価します。

3.2 抽象的な質問応答

RAGモデルは、単純な抽出QAを超えて、自由形式の抽象的なテキスト生成で質問に答えることができます。知識集約的な設定でRAGの自然言語生成（NLG)をテストするために、 MSMARCO NLGタスクv2.1 [43]を使用します。タスクは質問、質問ごとに検索エンジンから取得した10のゴールドパッセージ、および取得したパッセージから注釈を付けた完全な文の回答で構成されています。提供されたパッセージは使用せず、質問と回答のみを使用して処理します。

MSMARCO はオープンドメインの抽象 QA タスクです。MSMARCO には、 「カリフォルニア州ボルケーノの天気は?」など、ゴールド パッセージにアクセスしないと参照回答と一致する方法で回答できない質問がいくつかあるため、ゴールド パッセージを使用しないとパフォーマンスは低くなります。
また、一部の MSMARCO の質問は、Wikipedia だけでは回答できないことにも注意してください。この場合、 RAG はパラメトリック知識に頼って、適切な回答を生成できます。

3.3 ジェパディ問題生成

RAG の非 QA 環境での生成能力を評価するために、オープン ドメインの質問生成を研究します。通常は短くて簡単な質問で構成される標準的なオープン ドメイン QA タスクの質問を使用するのではなく、Jeopardy の質問を生成するという、より要求の厳しいタスクを提案します。
ジェパディは、ある実体に関する事実からその実体を推測するという珍しい形式です。
たとえば、「ワールドカップ」は、「1986 年にメキシコは、この国際スポーツ大会を 2 度開催した最初の国となった」という質問に対する答えです。Jeopardy の質問は正確で事実に基づいた記述であるため、回答エンティティを条件とする Jeopardy の質問を生成することは、知識集約型の困難な生成タスクとなります。

SearchQA [10]の分割を使用し、 100K のトレーニング、14K の開発、27K のテスト例を使用します。これは新しいタスクなので、比較のために BART モデルをトレーニングします。 [67] に従い、 SQuAD 調整済みの Q-BLEU-1 メトリック[42] を使用して評価します。Q -BLEU は BLEU の変形であり、一致するエンティティの重みが高く、標準的なメトリックよりも質問生成に関する人間の判断との相関が高くなっています。また、生成の事実性を評価するためのものと、特異性のためのものの 2 つの人間による評価も実行します。事実性を、信頼できる外部ソースによって文が裏付けられるかどうかと定義し、特異性を入力と出力間の高い相互依存性と定義します[33]。ベスト プラクティスに従い、一対比較評価を使用します[34]。評価者には、回答と、 BART から生成された質問と RAG から生成された質問が 2 つ表示されます。次に、質問 A の方が良い、質問 B の方が良い、どちらも良い、どちらも良くない、という4 つの選択肢から 1 つを選択するよう求められます。

3.4 事実の検証

FEVER [56] では、自然言語の主張が Wikipedia によって支持されているか反駁されているか、あるいは判断するのに十分な情報がないかどうかを分類する必要がある。このタスクでは、主張に関連する証拠を Wikipedia から取得し、この証拠に基づいて推論して、主張が真か偽か、あるいは Wikipedia だけでは検証できないかを分類する必要がある。
FEVER は、困難な含意推論タスクを伴う取得問題である。また、生成ではなく分類を処理する RAG モデルの能力を調査するための適切なテストベッドも提供する。FEVERクラス ラベル (支持、反駁、または十分な情報なし) を単一の出力トークンにマッピングし、主張クラスのペアで直接トレーニングする。重要なのは、FEVER に対する他のほとんどのアプローチとは異なり、取得した証拠に対して監督を使用しないことである。多くの実際のアプリケーションでは、取得監督信号は利用できないため、そのような監督を必要としないモデルは、より広範囲のタスクに適用できる。我々は2つのバリエーションを検討した。標準的な3元分類タスク（支持/反論/情報不足)と、 ThorneとVlachos [57]で研究された2元分類タスク（支持/反論)である。どちらの場合も、ラベルの精度を報告する。

4 件の結果

4.1 オープンドメイン質問応答

表 1 は、RAG と最先端のモデルの結果を示しています。4 つのオープンドメイン QA タスクすべてにおいて、RAG は新たな最先端技術を設定しました (TQA の T5 と同等の分割のみ)。RAG は、 「クローズドブック」(パラメトリックのみ) アプローチの生成柔軟性と、「オープンブック」検索ベースのアプローチのパフォーマンスを組み合わせています。REALM や T5+SSM とは異なり、RAG は高価で特殊な「顕著なスパンのマスキング」事前トレーニング[20]なしで強力な結果を実現しています。RAG のリトリーバーは、Natural Questions と TriviaQA で検索監督を使用する DPR のリトリーバーを使用して初期化されることは注目に値します。RAG は、BERT ベースの「クロス エンコーダー」を使用してドキュメントを再ランク付けし、抽出リーダーも使用する DPR QA システムと比較して優れています。RAGは、最先端のパフォーマンスには再ランク付けも抽出リーダーも必要ないことを示しています。

抽出可能な場合でも、回答を生成することにはいくつかの利点がある。回答に関する手がかりはあるものの、回答をそのまま含まない文書でも、正しい回答を生成するのに役立つ可能性がある。これは標準的な抽出アプローチでは不可能であり、

表1: オープンドメインQAテストのスコア。TQAの場合、左の列はオープンドメインQAの標準テストセットを使用し、右の列はTQA-Wikiを使用します。テストセット。詳細については付録Dを参照してください。

	モデル	NQ	TQA	WQ	CT
閉鎖本	T5-11B [52]	34.5	- /50.1	37.4 - /60.5	-
	T5-11B+SSM[52]	36.6	44.7		-
開ける本	レルム [20]	40.4 - / -	DPR [26]	41.5 57.9/ -	40.7 46.8 41.1 50.6
	RAGトークン	44.1	55.2/66.1	45.5	50.0
	RAG配列	44.5	56.8/68.0	45.2	52.2

表 2: 生成と分類のテストスコア。MS-MARCO SotAは[4]、FEVER-3は[68] 、FEVER-2は[57] *ゴールドコンテキスト/証拠を使用します。ゴールドアクセスなしのベストモデルが下線付きで表示されます。

モデル	ジェパディ B-1	ジェパディ QB-1	ジェパディ RL	MSMARCO B-1	MSMARCO ラベル	FVR3 アクセプター	FVR2
ソタ	-	-	49.8*	49.9*	76.8	92.2*	
バート	15.1	19.7	38.2	41.6	64.0	81.1	
ラグトク。	17.3	22.2	40.1	41.5	72.5	89.5	____
RAG シーケンス	14.7	21.4	40.8	44.2	____	____	

文書のより効果的なマージナリゼーションに。さらに、RAGは正しい答えを生成できる。正しい答えが検索された文書にない場合でも、そのような状況では11.8%の精度を達成しました。NQ の場合、抽出モデルではスコアが 0% になります。

4.2 抽象的な質問応答

表2に示すように、RAGシーケンスはOpen MS-MARCO NLGでBARTより2.6Bleu優れています。ポイントと2.6 Rouge-Lポイントを獲得しました。RAGは最先端のモデルのパフォーマンスに近づいており、（i)これらのモデルは、特定の情報を使用して金の通路にアクセスし、参照回答を生成する、（ii)多くの質問はゴールドパッセージなしでは答えられない、そして（iii)Wikipediaだけではすべての質問に答えられるわけではない。表3は生成された回答の一部を示している。定性的に見ると、RAGモデルは幻覚が少なく、事実に基づいた結果を生成することがわかった。テキストをBARTよりも頻繁に修正します。また、RAG世代はBARTよりも多様であることがわかります。BART世代（§4.5を参照）。

4.3 ジェパディ問題生成

表2は、RAG-TokenがJeopardyの質問生成においてRAG-Sequenceよりも優れていることを示しています。どちらのモデルもQ-BLEU-1でBARTを上回る性能を示した。4は452以上の人間による評価結果を示している。BARTとRAGトークンの世代ペア。評価者はBARTの方が事実に基づいたものだったと指摘した。RAGは7.1%のケースでより事実に基づいたものであったが、RAGは42.7%のケースでより事実に基づいたものであった。BARTはさらに17%のケースで事実に基づいており、RAGの有効性を明確に示している。最先端の世代モデルよりもタスクを優先する。評価者はまた、RAG世代の方が大幅に特異性があります。表 3 は各モデルの典型的な世代を示しています。

ジェパディの質問には2つの別々の情報が含まれていることが多く、RAGトークンは複数の文書のコンテンツを組み合わせたレスポンスを生成できるため、最適です。図2は例えば、「太陽」を生成する場合、「太陽」について言及している文書2の事後確率は高い。「日はまた昇る」。同様に、「武器よさらば」が生成されます。興味深いことに、各書籍の最初のトークンが生成された後、ドキュメントの後方部分は平坦化されます。この観察は、ジェネレータが特定の条件に依存せずにタイトルを完成させることができることを示唆している。言い換えれば、モデルのパラメトリック知識はタイトルを完成させるのに十分です。この仮説の証拠を見つけるには、BARTのみのベースラインに部分的なデコード「日曜日。BARTは世代を完了します。 「日はまた昇る」は、「日はまた昇る」の著者による小説です。「日はまた昇る」というタイトルを示す「Also Rises」がBARTのパラメータに保存されています。同様に、BARTは部分的な解読を完了します。 「日はまた昇る」は「A「日はまた昇る」は「武器よさらば」の著者による小説である。この例はパラメトリック記憶と非パラメトリック記憶がどのように連携するか—非パラメトリック要素がパラメトリックメモリに保存された特定の知識を引き出し、生成をガイドします。

4.4 事実の検証

表2はFEVERの結果を示しています。3ウェイ分類では、RAGスコアは4.3%以内です。最先端のモデルは、ドメイン固有のアーキテクチャと複雑なパイプラインシステムであり、RAG では必要とされない、中間回収監督を使用して訓練された実質的なエンジニアリング。

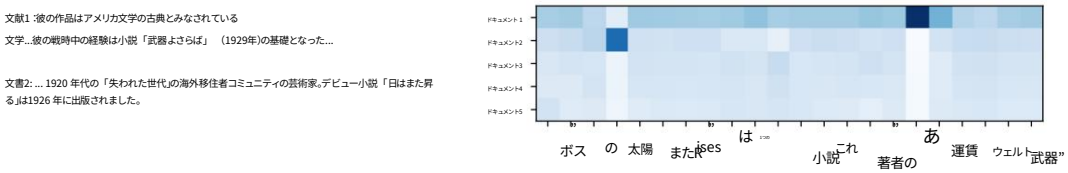

図2: 5つの検索文書を使用したJeopardy生成での入力「Hemingway」に対する各生成トークンのRAGトークン文書事後分布 p(zi |x, yi , y−i) 。文書1の事後分布は「武器よさらば」を生成するときに高く、文書2の事後分布は「日はまた昇る」を生成するときに高くなります。

表 3: 生成タスクの例。RAG モデルは、より具体的で事実に基づいた正確な応答を生成します。「?」は事実上誤った応答を示し、* は部分的に正しい応答を示します。

タスク	入力	モデル生成
MS-マルコ	中間を定義する耳	BART ?中耳は中耳と鼻の間にある耳の部分です。 RAG-T 中耳は鼓膜の内側にある耳の部分です。 RAG-S 中耳には鼓室と 3 つの耳小骨が含まれます。
	スコットランドで必要な通貨	BART スコットランドで必要な通貨はポンドです。 RAG-T ポンドはスコットランドで必要な通貨です。 RAG-S スコットランドで必要な通貨はポンドです。
危険質問世代	ワシントン	BART ?この州は米国で最も多くの郡を抱えている RAG-T アメリカ合衆国大統領にちなんで名付けられた唯一のアメリカの州です。 RAG-S マウント•レーニア国立公園がある州です。 BART
	神聖コメディ	*ダンテのこの叙事詩は、地獄篇、煉獄篇、冥界篇の 3 つの部分に分かれています。 RAG-T ダンテの「地獄篇」はこの叙事詩の最初の部分です。 RAG-S この 14 世紀の作品は、「地獄篇」、「煉獄篇」、「天国篇」の 3 つのセクションに分かれています。

2方向分類については、ThorneとVlachos [57]と比較します。彼らは、ゴールド証拠文を与えられた場合に主張を真か偽かに分類するようにRoBERTa [35]をトレーニングします。RAGは、主張のみが与えられ、独自の証拠を取得しているにもかかわらず、このモデルの2.7%以内の精度を達成しています。

また、RAG によって取得された文書が、FEVER でゴールド証拠として注釈が付けられた文書に対応するかどうかも分析します。RAG によって取得された上位k 個の文書とゴールド証拠注釈の間の論文タイトルの重複を計算します。取得された上位文書は 71% のケースでゴールド論文からのものであり、取得された上位 10 件の論文にゴールド論文が含まれているケースが 90% あることがわかりました。

4.5 追加の結果

世代の多様性セクション4.3では、RAGモデルはJeopardy問題生成においてBARTよりも事実に基づいており、より具体的であることを示しています。多様性を促進するデコードに関する最近の研究[33、59、39]に続いて、異なるモデルによって生成された合計ngramに対する異なるngramの比率を計算することで、世代の多様性も調査します。表5は、RAG-Sequenceの世代がRAG-Tokenの世代よりも多様であり、どちらも多様性を促進するデコードを必要とせずにBARTよりも大幅に多様であることを示しています。

検索アブレーションRAG の重要な特徴は、タスクに関連する情報を検索することを学習することです。
検索メカニズムの有効性を評価するために、トレーニング中に検索者をフリーズさせるアブレーションを実行します。表 6 に示すように、学習された検索によってすべてのタスクの結果が向上します。

RAGの高密度リトリーバーと単語重複ベースのBM25リトリーバー[53]を比較する。ここでは、 RAGのリトリーバーを固定BM25システムに置き換え、 p(z|x)を計算するときにBM25検索スコアをロジットとして使用する。表6に結果を示す。FEVERの場合、BM25が最も優れているが、これはおそらくFEVERの主張がエンティティ中心であり、単語重複ベースの検索に適しているためだろう。微分化可能な検索は他のすべてのタスク、特にそれが非常に重要なオープンドメインQAの結果を改善します。

インデックスのホットスワッピングRAGのような非パラメトリックメモリモデルの利点は、テスト時に知識を簡単に更新できることです。T5やBARTのようなパラメトリックのみのモデルでは、世界が変化するにつれて動作を更新するためにさらにトレーニングが必要です。これを実証するために、DrQA [5]を使用してインデックスを構築します。
2016年12月のWikipediaダンプと、このインデックスを使用したRAGの出力を、主な結果（2018年12月)の新しいインデックスと比較します。私たちは、交代した82人の世界の指導者のリストを作成しました。

表4: ジェパディの人間による評価
質問生成タスク。

	事実の特定性	
BARTの方が良い	7.1%	16.8%
RAGの方が良い	42.7%	37.4%
どちらも良い	11.7%	11.8%
どちらも貧しい	17.7%	6.9%
過半数なし	20.8%	20.1%

表5: 異なるトリグラムと合計トリグラムの比率
生成タスク。

	MSMARCO 危険 QGen	
ゴール	89.6%	90.0%
ド BART	70.7%	32.4%
RAG トークン RAG-	77.8%	46.8%
Seq。	83.5%	53.8%

表 6: 開発セットのアブレーション。FEVER は分類タスクなので、両方の RAG モデルは同等です。

モデル	NQ TQA WQ CT Jeopardy-QGen MSマルコ FVR-3 FVR-2 完全一致 B-1 QB-1 RL B-1	ラベルの精度	
RAG-トークン-BM25 29.7 41.5 RAG-シーケンス-BM25 31.8 44.1	32.1 33.1 17.5 22.3 36.6 33.8 11.1 19.5	55.5 48.4 56.5 46.9	75.1 91.6
RAG-トークン-凍結 37.8 50.1 37.1 51.1 16.7 21.7 RAG-シーケンス-凍結 52.6 11.8 19.6	41.2 52.1 41.8	55.9 49.4 56.7 47.3	72.9 89.4
RAGトークン RAGシーケンス	43.5 54.8 46.5 51.9 17.9 22.6 44.0 55.8 44.9 53.4 15.3 21.5	56.2 49.4 57.2 47.5	74.5 90.6

これらの日付の間に「{役職} は誰ですか?」というテンプレートを使用します (例: 「ペルーの大統領は誰ですか?」)
各インデックスでNQ RAGモデルを照会します。RAGは2016年のインデックスを使用して70%の正解率を達成しました。
2016年の世界リーダーの68%が2018年のインデックスを使用して2018年の世界リーダーの68%が2018年のインデックスを使用しています。
指数は低いです（2018年の指数と2016年のリーダーでは12%、2016年の指数と2018年のリーダーでは4%）。
これは、非パラメトリック メモリを置き換えるだけで RAG の世界知識を更新できることを示しています。

より多くの文書を取得することによる効果モデルは5または10の潜在的文書でトレーニングされる。
文書間でパフォーマンスに大きな違いは見られません。
テスト時に取得する文書の数を調整する柔軟性があり、パフォーマンスに影響を与える可能性があります。
図3（左)は、テスト時により多くの文書を取得すると、単調に改善されることを示しています。
RAGシーケンスのオープンドメインQA結果ですが、RAGトークンのパフォーマンスは10でピークに達します。
図3（右)は、より多くの文書を取得すると、Rouge-Lが高くなることを示しています。
RAG-Token は Bleu-1 を犠牲にして増加しますが、RAG-Sequence の場合、その効果はそれほど顕著ではありません。

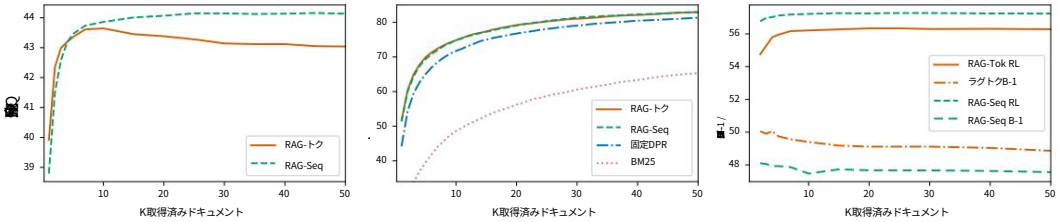

図 3: 左: より多くの文書が検索されたときの NQ パフォーマンス。中央: NQ での検索リコール パフォーマンス。右: より多くの文書が検索されたときの MS-MARCO Bleu-1 と Rouge-L。

5 関連研究

単一タスク検索これまでの研究では、検索によってさまざまなタスクのパフォーマンスが向上することが示されています。
NLPタスクを個別に考えると、オープンドメイン質問応答[5, 29]、
事実確認[56]、事実補完[48]、長文質問応答[12]、 Wikipedia記事
生成[36]、対話[41、65、9、13] 、翻訳[17]、言語モデル[19、27]。
この研究は、検索を個々のタスクに組み込むというこれまでの成功を統合し、単一の
検索ベースのアーキテクチャは、複数のタスクにわたって強力なパフォーマンスを実現できます。

NLPの汎用アーキテクチャNLPタスク用の汎用アーキテクチャに関するこれまでの研究では、検索を使用せずに大きな成功を収めています。事前学習済みの単一の言語モデルは、微調整[49, 8]を行った後、 GLUEベンチマーク[60, 61]のさまざまな分類タスクで優れたパフォーマンスを発揮することが示されています。GPT-2 [50]は後に、左から右への事前学習済みの単一の言語モデルが識別タスクと生成タスクの両方で優れたパフォーマンスを発揮できることを示しました。

さらなる改善のために、BART [32]とT5 [51, 52]は、双方向の注意を活用して識別タスクと生成タスクでより強力なパフォーマンスを実現する、単一の事前トレーニング済みエンコーダーデコーダーモデルを提案しています。私たちの研究は、事前トレーニング済みの生成言語モデルを補強する検索モジュールを学習することにより、単一の統一されたアーキテクチャで可能なタスクの空間を拡大することを目指しています。

学習された検索情報検索において文書を検索する学習については、最近では我々の研究に似た事前学習済みのニューラル言語モデル[44, 26]を用いた重要な研究が行われています。一部の研究では、検索[46]、強化学習[6, 63, 62]、または我々の研究のような潜在変数アプローチ[31, 20]を使用して、質問応答などの特定の下流タスクを支援するために検索モジュールを最適化しています。これらの成功は、単一のタスクで強力なパフォーマンスを達成するために、さまざまな検索ベースのアーキテクチャと最適化手法を活用していますが、我々は単一の検索ベースのアーキテクチャを微調整することで、さまざまなタスクで強力なパフォーマンスを実現できることを示しています。

メモリベースのアーキテクチャ私たちの文書インデックスは、メモリネットワーク[64, 55]に類似した、ニューラルネットワークが注目する大きな外部メモリと見なすことができます。並行作業[14] は、私たちの研究のように生のテキストを取得するのではなく、入力内の各エンティティのトレーニングされた埋め込みを取得することを学習します。他の研究では、事実の埋め込みに注目することで、対話モデルが事実のテキストを生成する能力を向上させています[15, 13]。私たちのメモリの主な特徴は、分散表現ではなく生のテキストで構成されていることであり、これによりメモリは (i) 人間が読み取り可能で、モデルに一種の解釈可能性をもたらし、(ii) 人間が書き込み可能で、文書インデックスを編集することでモデルのメモリを動的に更新できます。このアプローチは知識集約型ダイアログでも使用されており、そこでは、エンドツーエンドの学習された検索ではなくTF-IDF を介して取得されたとはいえ、ジェネレーターは直接取得されたテキストに条件付けられています[9]。

検索と編集のアプローチ私たちの方法は、検索と編集スタイルのアプローチといくつかの類似点があります。検索と編集スタイルのアプローチでは、特定の入力に対して同様のトレーニング入力と出力のペアが検索され、編集されて最終的な出力が提供されます。これらのアプローチは、機械翻訳[18, 22]やセマンティック解析[21] など、多くの分野で成功していることが証明されています。私たちのアプローチには、検索されたアイテムを軽く編集することにあまり重点が置かれておらず、検索された複数のコンテンツからコンテンツを集約すること、潜在的な検索を学習すること、関連するトレーニングペアではなく証拠文書を取得することなど、いくつかの違いがあります。とはいえ、RAG テクニックはこれらの設定でうまく機能する可能性があり、将来有望な研究となる可能性があります。

6 議論

この研究では、パラメトリック メモリとノンパラメトリックメモリにアクセスできるハイブリッド生成モデルを紹介しました。RAG モデルはオープンドメイン QA で最先端の結果を得ることを示しました。人々は純粋にパラメトリックな BART よりも RAG の生成を好み、RAG の方が事実に基づいていて具体的だと感じていることがわかりました。学習した検索コンポーネントを徹底的に調査してその有効性を検証し、検索インデックスをホットスワップして再トレーニングを必要とせずにモデルを更新する方法を示しました。今後の研究では、BART に似たノイズ除去目的または別の目的で、2 つのコンポーネントを最初から共同で事前トレーニングできるかどうかを調査すると有益かもしれません。私たちの研究は、パラメトリック メモリとノンパラメトリック メモリがどのように相互作用し、それらを最も効果的に組み合わせる方法に関する新しい研究の方向性を切り開き、さまざまな NLP タスクに適用される可能性を示しています。

より広範な影響

この研究は、以前の研究に比べて、いくつかの社会的な利点があります。実際の事実に基づく知識 (この場合は Wikipedia) にさらに強く基づいているため、より事実に基づいた世代に対して「幻覚」を起こすことが少なくなり、制御性と解釈性が向上します。RAG は、たとえば医療インデックスを付与してそのトピックに関するオープン ドメインの質問をしたり、人々がより効率的に仕事を行えるように支援したりするなど、社会に直接的な利益をもたらすさまざまなシナリオで使用できます。

これらの利点には、潜在的な欠点も伴います。Wikipedia や潜在的な外部知識源は、おそらく完全に事実に基づいていて、偏見がまったくないということはないでしょう。RAG は言語モデルとして使用できることから、GPT-2 [50]と同様の懸念がここでも当てはまりますが、程度は低いと言えます。たとえば、ニュースやソーシャルメディアで虐待や偽造、誤解を招くコンテンツを生成したり、他人になりすましたり、スパム/フィッシングコンテンツの作成を自動化したりするために使用される可能性があるということです [54]。高度な言語モデルは、今後数十年でさまざまな仕事の自動化につながる可能性もあります[16]。これらのリスクを軽減するために、AI システムを使用して、誤解を招くコンテンツや自動化されたスパム/フィッシングと戦うことができます。

謝辞

著者らは、本論文に対する思慮深く建設的なフィードバックをいただいた査読者の方々、および RAG モデルを実行するためのコードをオープンソース化していただいた HuggingFace に感謝の意を表します。また、著者らは、生産的な議論とアドバイスをいただいた Kyunghyun Cho 氏と Sewon Min 氏にも感謝の意を表します。EP はNSF 大学院研究フェローシップの支援に感謝します。PL は FAIR PhD プログラムによってサポートされています。

参考文献

[1] Payal Bajaj、Daniel Campos、Nick Craswell、Li Deng、Jianfeng Gao、Xiaodong Liu、Rangan Majumder、Andrew McNamara、Bhaskar Mitra、Tri Nguyen、Mir Rosenberg、Xia Song、Alina Stoica、Saurabh Tiwary、Tong Wang。MS MARCO：人間が生成した機械読解データセット。arXiv：1611.09268 [cs]、2016年11月。URL http：//arxiv.org/abs/1611.09268。arXiv：1611.09268 。

[2] Petr BaudiとJan edivy。yodaqaシステムにおける質問応答タスクのモデリング。ヨーロッパ言語のクロス言語評価フォーラム国際会議、 222～228ページ。Springer、2015年。URL https://link.springer.com/chapter/10.1007% 2F978-3-319-24027-5_20。

[3] Jonathan Berant、Andrew Chou、Roy Frostig、Percy Liang。質問と回答のペアからのFreebaseによる意味解析。2013年自然言語処理における経験的手法に関する会議の議事録、1533～1544ページ、ワシントン州シアトル、米国、2013年10月。

計算言語学協会。URL http://www.aclweb.org/anthology/ D13-1160。

[4] Bin Bi、Chenliang Li、Chen Wu、Ming Yan、Wei Wang。Palm：文脈条件付き生成のためのオートエンコーディング＆自己回帰言語モデルの事前トレーニング。ArXiv、abs / 2004.07159、 2020年。URL https://arxiv.org/abs/2004.07159。

[5] Danqi Chen、Adam Fisch、Jason Weston、Antoine Bordes。Wikipediaを読んでオープンドメインの質問に答える。計算言語学協会第55回年次会議議事録（第1巻：長文論文）、1870～1879ページ、バンクーバー、カナダ、2017年7月。計算言語学協会。doi：10.18653/v1/P17-1171。URL https://www.aclweb.org/anthology/P17-1171。

[6] Eunsol Choi、Daniel Hewlett、Jakob Uszkoreit、Illia Polosukhin、Alexandre Lacoste、 Jonathan Berant。長い文書に対する粗から細への質問応答。計算言語学協会第55回年次会議の議事録（第1巻：長文論文）、 209～220ページ、バンクーバー、カナダ、2017年7月。計算言語学協会。doi： 10.18653/v1/P17-1020。URL https://www.aclweb.org/anthology/P17-1020。

[7]クリストファー•クラークとマット•ガードナー。シンプルで効果的な複数段落の読解。arXiv:1710.10723 [cs]、2017年10月。URL http://arxiv.org/abs/1710.10723。 arXiv: 1710.10723.

[8] Jacob Devlin、Ming-Wei Chang、Kenton Lee、Kristina Toutanova。BERT：言語理解のためのディープ双方向トランスフォーマーの事前トレーニング。2019年北米支部計算言語学会会議議事録：人間言語技術、第1巻（長編論文と短編論文）、4171〜4186ページ、ミネソタ州ミネアポリス、2019年6月。計算言語学会。doi：10.18653/v1/N19-1423。

URL https://www.aclweb.org/anthology/N19-1423.

[9]エミリー•ディナン、スティーブン•ローラー、カート•シュスター、アンジェラ•ファン、マイケル•アウリ、ジェイソン•ウェストン。ウィキペディアの魔法使い：知識駆動型会話エージェント。国際学習表現会議、2019年。URL https://openreview.net/forum?id=r1l73iRqKm。

[10] Matthew Dunn、Levent Sagun、Mike Higgins、V. Ugur Guney、Volkan Cirik、Kyunghyun Cho。SearchQA：検索エンジンからのコンテキストで拡張された新しいQ&Aデータセット。arXiv：1704.05179 [cs]、2017年4月。URL http://arxiv.org/abs/1704.05179。 arXiv: 1704.05179.

[11]アンジェラ•ファン、マイク•ルイス、ヤン•ドーフィン。階層的ニューラルストーリー生成。第56回計算言語学会年次会議議事録（第1巻：長編論文）、889〜898ページ、メルボルン、オーストラリア、2018年7月。計算言語学会。doi: 10.18653/v1/P18-1082。URL https://www.aclweb.org/anthology/ P18-1082。

[12]アンジェラ•ファン、ヤシン•ジャーナイト、イーサン•ペレス、デビッド•グランジェ、ジェイソン•ウェストン、マイケル•アウリ。ELI5:長文質問応答。計算言語学協会第57回年次会議の議事録、3558〜3567ページ、フィレンツェ、イタリア、2019年7月。計算言語学協会。doi: 10.18653/v1/P19-1346。URL https://www.aclweb.org/ anthology/P19-1346。

[13]アンジェラ•ファン、クレア•ガーデン、クロエ•ブラッド、アントワーヌ•ボルデス。KNNベースの複合メモリによるトランスフォーマーの拡張、2020年。URL https://openreview.net/forum?id= H1gx1CNKPH。

[14]ティボー•フェブリ、リビオ•バルディーニ•ソアレス、ニコラス•フィッツジェラルド、ウンソル•チョイ、トム•クウィアトコウスキー。エンティティをエキスパートとして: エンティティ監視によるスパースメモリアクセス。ArXiv、abs/2004.07202、2020年。URL https://arxiv.org/abs/2004.07202。

[15] Marjan Ghazvininejad、Chris Brockett、Ming-Wei Chang、Bill Dolan、Jianfeng Gao、Wen tau Yih、Michel Galley。知識に基づくニューラル会話モデル。AAAI人工知能会議、2018年。URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16710 。

[16] Katja Grace、John Salvatier、Allan Dafoe、Baobao Zhang、Owain Evans。AIはいつ人間のパフォーマンスを超えるのか ?AI専門家による証拠。CoRR、abs/1705.08807、2017年。URL http://arxiv.org/abs/1705.08807。

[17] Jiatao Gu、Yong Wang、Kyunghyun Cho、Victor OK Li。検索エンジン誘導ニューラル機械翻訳。AAAI人工知能会議、2018年。URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17282 。

[18] Jiatao Gu、Yong Wang、Kyunghyun Cho、Victor OK Li。検索エンジン誘導ニューラル機械翻訳。第32回AAAI人工知能会議、AAAI 2018、第32回AAAI人工知能会議、AAAI 2018、5133〜5140ページ。AAAIプレス、2018年。第32回AAAI人工知能会議、AAAI 2018;会議日：2018年2月2日から2018年2月7日まで。

[19]ケルビン•グー、橋本辰則、ヨナタン•オーレン、パーシー•リャン。プロトタイプの編集による文章生成。計算言語学協会誌、6:437–450、 2018年。doi: 10.1162/tacl_a_00030。URL https://www.aclweb.org/anthology/Q18-1031。

[20]ケルビン•グー、ケントン•リー、ゾーラ•トゥン、パヌポン•パスパット、ミンウェイ•チャン。レルム:
検索強化言語モデルの事前トレーニング。ArXiv、abs/2002.08909、2020。URL https：
//arxiv.org/abs/2002.08909.

[21]辰徳B橋本、ケルビン•グー、ヨナタン•オーレン、パーシー•S•リャン。あ
構造化された出力を予測するための取得および編集フレームワーク。 S.ベンジオでは、
H. Wallach、H. Larochelle、K. Grauman、N. Cesa-Bianchi、R. Garnett 編著、Advances in Neural Information
Processing Systems 31、10052 ページ～
10062. カランアソシエイツ社、2018年。URL http://papers.nips.cc/paper/
8209-a 構造化出力を予測するための取得および編集フレームワーク。
pdf.

[22]ナビル•ホセイン、マルジャン•ガズヴィニネジャド、ルーク•ゼトルモイヤー。シンプルで効果的な検索•編集•再ランク付
けテキスト生成。第58回米国言語学会年次大会論文集。
計算言語学、2532～2538ページ、オンライン、2020年7月。計算言語学協会。doi: 10.18653/v1/2020.acl-main.228。
URL https://www.aclweb.org/
アンソロジー/2020.acl-main.228.

[23]ジェフ•ジョンソン、マタイス•ドゥーズ、エルベ•ジェグー。 GPUによる10億規模の類似性検索。 arXiv
プレプリントarXiv:1702.08734、2017年。URL https://arxiv.org/abs/1702.08734。

[24]マンダー•ジョシ、ウンソル•チェ、ダニエル•ウェルド、ルーク•ゼトルモイヤー。トリビアQA：大規模
遠隔教師付き読解力チャレンジデータセット。
第55回計算言語学会年次大会（第1巻：長編論文）
1601～1611ページ、カナダ、バンクーバー、2017年7月。計算言語学協会。
doi: 10.18653/v1/P17-1147. URL https://www.aclweb.org/anthology/P17-1147.

[25]アルマンド•ジュランとトーマス•ミコロフ。 スタック拡張型リカレントネットによるアルゴリズムパターン
の推論。第28回国際数学者会議の議事録
ニューラル情報処理システム - 第 1 巻、NIPS'15、190 ～ 198 ページ、ケンブリッジ、マサチューセッツ州、米国、2015
年。MIT プレス。URL https://papers.nips.cc/paper/
5857 スタック拡張リカレント ネットを使用したアルゴリズム パターンの推論。

[26]ウラジミール•カルプキン、バーラス•オグズ、セウォン•ミン、レデル•ウー、セルゲイ•エドゥノフ、ダンチー•チェン、
Wen-tau Yih. オープンドメインの質問応答のための高密度パッセージ検索。arXiv プレプリント
arXiv:2004.04906、2020。URL https://arxiv.org/abs/2004.04906。

[27]ウルヴァシ•カンデルワル、オメル•レヴィ、ダン•ジュラフスキー、ルーク•ゼトルモイヤー、マイク•ルイス。記憶による一般
化：最近傍言語モデル。国際会議
学習表現、2020年。URL https://openreview.net/forum?id=HklBjCEKvH。

[28] Diederik P. KingmaとJimmy Ba. Adam：確率的最適化の方法。Yoshua
ベンジオとヤン•ルカン、編集者、第3回学習表現に関する国際会議、
ICLR 2015、米国カリフォルニア州サンディエゴ、2015 年 5 月 7 ～ 9 日、カンファレンス トラック プロシーディングス、2015。URL
http://arxiv.org/abs/1412.6980。

[29]トム•クウィアトコウスキー、ジェニマリア•パロマキ、オリヴィア•レッドフィールド、マイケル•コリンズ、アンクル•パリク、
クリス•アルバーティ、ダニエル•エプスタイン、イリア•ポロスキン、マシュー•ケルシー、ジェイコブ•デブリン、ケントン•リー、クリス
ティーナ•N•トウタノバ、リオン•ジョーンズ、ミンウェイ•チャン、アンドリュー•ダイ、ジェイコブ
Uszkoreit、Quoc Le、Slav Petrov。自然な質問：質問応答研究のベンチマーク。
計算言語学協会論文集、2019年。URL https://
tomkwiat.users.x20web.corp.google.com/papers/
natural-questions/main-1455-kwiatkowski.pdf を参照してください。

[30]ギヨーム•ランプル、アレクサンドル•サブレイロール、マルク•アウレリオ•ランザート、ルドヴィック•デノワイエ、
Herve Jegou. プロダクトキー付きの大容量メモリレイヤー。H. Wallach、H. Larochelle、
A. Beygelzimer、F. d' Alché-Buc、E. Fox、および R. Garnett 編著、Advances in Neural Information Processing
Systems 32、8548～8559 ページ。Curran Associates、Inc.、2019 年。URL http:
//papers.nips.cc/paper/9061-large-memory-layers-with-product-keys.pdf。

[31]ケントン•リー、ミンウェイ•チャン、クリスティーナ•トゥタノバ。弱教師付き学習における潜在検索
オープンドメインの質問応答。協会第57回年次総会の議事録

計算言語学協会、6086〜6096ページ、フィレンツェ、イタリア、2019年7月。計算言語学協会。doi: 10.18653/v1/P19-1612。URL https://www.aclweb.org/anthology/P19-1612 。

[32] Mike Lewis、Yinhan Liu、Naman Goyal、Marjan Ghazvininejad、Abdelrahman Mohamed、 Omer Levy、Veselin Stoyanov、およびLuke Zettlemoyer。BART：自然言語生成、翻訳、および理解のためのシーケンスツーシーケンス事前トレーニングのノイズ除去。arXivプレプリントarXiv：1910.13461、2019年。URL https://arxiv.org/abs/1910.13461。

[33] Jiwei Li、Michel Galley、Chris Brockett、Jianfeng Gao、Bill Dolan。ニューラル会話モデルのための多様性を促進する目的関数。2016年北米支部計算言語学会会議論文集「人間の言語技術」、110〜119ページ、カリフォルニア州サンディエゴ、2016年6月。計算言語学会。doi: 10.18653/v1/N16-1014。URL https://www.aclweb.org/anthology/ N16-1014。

[34]マーガレット•リー、ジェイソン•ウェストン、スティーブン•ローラー。「Acute-eval：最適化された質問とマルチターン比較による対話評価の改善」ArXiv、abs/1909.03087、2019年。URL https://arxiv.org/abs/1909.03087。

[35] Hairong Liu、Mingbo Ma、Liang Huang、Hao Xiong、Zhongjun He。テキストと音声の埋め込みを組み合わせた堅牢なニューラル機械翻訳。第57回計算言語学会年次会議論文集、3044〜3049ページ、フィレンツェ、イタリア、2019年7月。計算言語学会。doi: 10.18653/v1/P19-1291。URL https://www.aclweb.org/anthology/P19-1291。

[36] Peter J. Liu*、Mohammad Saleh*、Etienne Pot、Ben Goodrich、Ryan Sepassi、Lukasz Kaiser、Noam Shazeer。長いシーケンスを要約してWikipediaを生成する。International Conference on Learning Representations、2018年。URL https://openreview.net/forum? id=Hyg0vbWC-。

[37] Yury A. MalkovとDA Yashunin。階層的なナビゲート可能なスモールワールドグラフを使用した効率的で堅牢な近似最近傍探索。IEEE Transactions on Pattern Analysis and Machine Intelligence、42:824–836、2016年。URL https://arxiv.org/abs/1603.09320。

[38]ゲイリー•マーカス。AIの次の10年：堅牢な人工知能に向けた4つのステップ。arXivプレプリントarXiv：2002.06177、2020年。URL https://arxiv.org/abs/2002.06177。

[39] Luca Massarelli、Fabio Petroni、Aleksandra Piktus、Myle Ott、Tim Rocktäschel、Vassilis Plachouras、Fabrizio Silvestri、Sebastian Riedel。デコード戦略が生成されたテキストの検証可能性に与える影響。arXivプレプリントarXiv：1911.03587、2019年。URL https：//arxiv.org/abs/1911.03587 。

[40] Paulius Micikevicius、Sharan Narang、Jonah Alben、Gregory Diamos、Erich Elsen、David Garcia、 Boris Ginsburg、Michael Houston、Oleksii Kuchaiev、Ganesh Venkatesh、Hao Wu。「混合精度トレーニング」ICLR、2018年。URL https://openreview.net/forum?id=r1gs9JgRZ。

[41] Nikita Moghe、Siddhartha Arora、Suman Banerjee、Mitesh M. Khapra。会話システム構築のための背景知識の活用に向けて。2018年自然言語処理における経験的手法に関する会議の議事録、2322〜2332ページ、ブリュッセル、ベルギー、2018年10〜11月。計算言語学協会。doi: 10.18653/v1/D18-1255。URL https://www.aclweb.org/anthology/D18-1255。

[42] Preksha NemaとMitesh M. Khapra。質問生成システムの評価のためのより良い指標に向けて。2018年自然言語処理における経験的手法に関する会議の議事録、3950〜3959ページ、ブリュッセル、ベルギー、2018年10月〜11月。計算言語学協会。doi: 10.18653/v1/D18-1429。URL https://www.aclweb.org/ anthology/D18-1429。

[43] Tri Nguyen、Mir Rosenberg、Xia Song、Jianfeng Gao、Saurabh Tiwary、Rangan Majumder、 Li Deng。MS MARCO：人間が生成した機械読解データセット。Tarek Richard Besold、Antoine Bordes、Artur S. d'Avila Garcez、Greg Wayne編、Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic

アプローチ2016は、第30回神経情報処理システム年次会議（NIPS 2016）と共催され、2016年12月9日にスペインのバルセロナで開催され、CEURワークショップ議事録の第1773巻に掲載されました。CEUR-WS.org、2016年。URL http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf。

[44] Rodrigo NogueiraとKyunghyun Cho。BERTによるパッセージの再ランキング。arXivプレプリントarXiv:1901.04085、2019年。URL https://arxiv.org/abs/1901.04085。

[45] Myle Ott、Sergey Edunov、Alexei Baevski、Angela Fan、Sam Gross、Nathan Ng、David Grangier、 Michael Auli。fairseq：シーケンスモデリングのための高速で拡張可能なツールキット。 2019年北米支部計算言語学会会議論文集（デモンストレーション）48〜53ページ、ミネソタ州ミネアポリス、2019年6月。計算言語学会。doi：10.18653/v1/N19-4009。URL https://www.aclweb.org/anthology/N19-4009 。

[46] Ethan Perez、Siddharth Karamcheti、Rob Fergus、Jason Weston、Douwe Kiela、Kyunghyun Cho。q&aモデルの説得を学習することで一般化可能な証拠を見つける。2019年自然言語処理の経験的手法に関する会議および第9回自然言語処理に関する国際合同会議（EMNLP-IJCNLP）の議事録、 2402〜2411ページ、中国香港、2019年11月。計算言語学協会。doi：10.18653/v1/D19-1244。URL https://www.aclweb.org/anthology/D19-1244。

[47] Fabio Petroni、Tim Rocktäschel、Sebastian Riedel、Patrick Lewis、Anton Bakhtin、Yuxiang Wu、 Alexander Miller。知識ベースとしての言語モデル ?2019年自然言語処理における経験的手法に関する会議および第9回自然言語処理に関する国際合同会議（EMNLP-IJCNLP）の議事録、2463〜2473ページ、香港、中国、2019年11月。計算言語学協会。doi：10.18653/v1/ D19-1250。URL https://www.aclweb.org/anthology/D19-1250。

[48]ファビオ•ペトローニ、パトリック•ルイス、アレクサンドラ•ピクトゥス、ティム•ロックテッシェル、ウーシャン•ウー、アレクサンダー•H. Miller、Sebastian Riedel。コンテキストが言語モデルの事実予測に与える影響。Automated Knowledge Base Construction、2020年。URL https://openreview.net/forum? id=025X0zPfn。

[49] Alec Radford、Karthik Narasimhan、Tim Salimans、Ilya Sutskever。生成的事前トレーニングによる言語理解の証明、2018年。 https ://s3-us-west-2.amazonaws.com/openai-assets/research-covers/ language-unsupervised/language_understanding_paper.pdf。

私は-
メールアドレス

[50]アレック•ラドフォード、ジェフ•ウー、レウォン•チャイルド、デビッド•ルアン、ダリオ•アモデイ、イリヤ•ランゲージ•モデルは教師なしマルチタスク学習者、2019年。URL Sutskever。 https ://d4mucfpksywv.cloudfront.net/better-language-models/language_ models_are_unsupervised_multitask_learners.pdf。

[51] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、 Yanqi Zhou、Wei Li、Peter J. Liu。統合テキストツーテキスト変換器による転移学習の限界の探究。arXiv e-prints、2019年。URL https://arxiv.org/abs/1910.10683。

[52]アダム•ロバーツ、コリン•ラフェル、ノアム•シャジール。言語モデルのパラメータにはどれだけの知識を詰め込むことができるか ?arXiv e-prints、2020年。URL https://arxiv.org/abs/ 2002.08910。

[53]スティーブン•ロバートソンとヒューゴ•サラゴサ。確率的関連性フレームワーク：Bm25以降。発見。Trends Inf. Retr.、3(4):333–389、2009年4月。ISSN 1554-0669。doi：10.1561/ 1500000019。URL https://doi.org/10.1561/1500000019。

[54]アイリーン•ソライマン、マイルズ•ブランデージ、ジャック•クラーク、アマンダ•アスケル、アリエル•ハーバート•ヴォス、ジェフ•ウー、アレック•ラドフォード、ジャン•ビン•ワン。リリース戦略と言語モデルの社会的影響。ArXiv、abs/1908.09203、2019年。

[55]サインバヤル•スフバータル、アーサー•シュラム、ジェイソン•ウェストン、ロブ•ファーガス。エンドツーエンドのメモリネットワーク。C.コルテス、NDローレンス、DDリー、M.杉山、R.ガーネット編、神経情報処理システムの進歩28、2440-2448ページ。カランアソシエイツ社、2015年。URL http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf.

[56] James Thorne、Andreas Vlachos、Christos Christodoulopoulos、Arpit Mittal。FEVER：事実抽出と検証のための大規模データセット。2018年北米支部計算言語学会会議議事録：人間言語技術、第1巻（長編論文）、809〜819ページ、ルイジアナ州ニューオーリンズ、2018年6月。計算言語学会。doi：10.18653/v1/N18-1074。URL https://www.aclweb.org/anthology/N18-1074。

[57] James H. ThorneとAndreas Vlachos。弾性重み統合による文対分類のモデルバイアス緩和における壊滅的な忘却の回避。ArXiv、abs / 2004.14366、 2020年。URL https://arxiv.org/abs/2004.14366。

[58]アシシュ•ヴァスワニ、ノーム•シェイザー、ニキ•パルマー、ヤコブ•ウスコレイト、ライオン•ジョーンズ、エイダン•N•ゴメス、ウカシュ•カイザー、イリア•ポロスキン。必要なのは注意力だけです。 I. Guyon、UV Luxburg、 S. Bengio、H. Wallach、R. Fergus、S. Vishwanathan、および R. Garnett 編集者、Advances in Neural Information Processing Systems 30、5998 〜 6008 ページ。 Curran Associates, Inc.、2017。URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf。

[59] Ashwin Vijayakumar、Michael Cogswell、Ramprasaath Selvaraju、Qing Sun、Stefan Lee、David Crandall、Dhruv Batra。複雑なシーンの記述を改善するための多様なビーム探索。
人工知能に関する AAAI カンファレンス、2018 年。URL https://www.aaai.org/ocs/index。 php/AAAI/AAAI18/paper/view/17329。

[60]アレックス•ワン、アマンプリート•シン、ジュリアン•マイケル、フェリックス•ヒル、オマー•レヴィ、サミュエル•ボウマン。
GLUE: 自然言語理解のためのマルチタスク ベンチマークおよび分析プラットフォーム。
2018 EMNLP ワークショップ BlackboxNLP: NLP のためのニューラル ネットワークの分析と解釈の議事録、353 〜 355 ページ、ブリュッセル、ベルギー、2018 年 11 月。計算言語学協会。doi: 10.18653/v1/W18-5446。URL https://www.aclweb.org/ anthology/W18-5446。

[61] Alex Wang、Yada Pruksachatkun、Nikita Nangia、Amanpreet Singh、Julian Michael、Felix Hill、Omer Levy、Samuel Bowman。SuperGLUE：汎用言語理解システムのためのより粘着性のあるベンチマーク。H. Wallach、H. Larochelle、A. Beygelzimer、 F. d\textquotesingle Alché-Buc、E. Fox、R. Garnett編、『Advances in Neural Information Processing Systems』32、3261〜3275ページ。Curran Associates、Inc.、2019年。URL https://arxiv.org/abs/1905.00537 。

[62] Shuohang Wang、Mo Yu、Xiaoxiao Guo、Zhiguo Wang、Tim Klinger、Wei Zhang、Shiyu Chang、 Gerry Tesauro、Bowen Zhou、Jing Jiang。R3 ：オープンドメインの質問応答のための強化ランクリーダー。Sheila A. McIlraithとKilian Q. Weinberger編、第32回AAAI人工知能会議（AAAI-18）、第30回革新的人工知能アプリケーション（IAAI-18）、および第8回AAAI人工知能教育進歩シンポジウム（EAAI-18)の議事録、米国ルイジアナ州ニューオーリンズ、2018年2月2日〜7日、5981〜5988ページ。 AAAI Press、2018 年。URL https://www.aaai.org/ocs/index。 php/AAAI/AAAI18/paper/view/16712。

[63] Shuohang Wang、Mo Yu、Jing Jiang、Wei Zhang、Xiaoxiao Guo、Shiyu Chang、Zhiguo Wang、 Tim Klinger、Gerald Tesauro、およびMurray Campbell。オープンドメインの質問応答における回答の再ランキングのための証拠の集約。ICLR、2018年。URL https://openreview.net/forum?id=rJl3yM-Ab 。

[64] Jason Weston、Sumit Chopra、Antoine Bordes。記憶ネットワーク。Yoshua BengioとYann LeCun編、第3回国際学習表現会議、ICLR 2015、サンディエゴ、カリフォルニア州、米国、2015年5月7〜9日、会議トラック議事録、2015年。URL http://arxiv.org/abs/1410.3916。

[65]ジェイソン•ウェストン、エミリー•ディナン、アレクサンダー•ミラー。「検索と改良：対話のためのシーケンス生成モデルの改良」。2018 EMNLPワークショップSCAIの議事録：検索指向会話AIに関する第2回国際ワークショップ、87〜92ページ、ブリュッセル、ベルギー、 2018年10月。計算言語学協会。doi：10.18653/v1/W18-5713。URL https://www.aclweb.org/anthology/W18-5713。

[66]トーマス•ウルフ、リサンドル•デビュー、ヴィクター•サン、ジュリアン•ショーモン、クレメント•ドゥラング、アンソニー•モワ、ピエリック•シスタック、ティム•ロー、レミ•ルーフ、モーガン•ファントウィッツ、ジョー•デイヴィソン、サム•シュライファー、パトリック•フォン•プラテン、クララ•マー、ヤシン•ジャーナイト、ジュリアン•プルー、カンウェン•シュー、テヴェン•ル•スカオ、シルヴァン•ガガー、マリアマ•ドラメ、クエンティン•ロエスト、アレクサンダー•M•ラッシュ。ハギングフェイスのトランスフォーマー:最先端の自然言語処理。 ArXiv、abs/1910.03771、2019。

[67] Shiyue ZhangとMohit Bansal。半教師あり質問応答のための質問生成における意味ドリフトへの対処。2019年自然言語処理の経験的手法に関する会議および第9回自然言語処理に関する国際合同会議（EMNLP-IJCNLP)の議事録、2495〜2509ページ、中国香港、2019年11月。計算言語学協会。doi：10.18653/v1/D19-1253。URL https://www.aclweb.org/anthology/D19-1253。

[68]鍾万君、徐京景、唐度宇、徐禅南、南端、周明、王嘉海、殷建。ファクトチェックのためのセマンティックレベルのグラフに対する推論。 ArXiv、abs/1909.03745、2019。
URL https://arxiv.org/abs/1909.03745.

検索強化生成のための付録 知識集約型NLPタスク

実装の詳細

オープンドメイン QA では、RAG トークン モデル用に取得した 15 個のドキュメントを使用してテスト数を報告します。RAG-Sequence モデルについては、取得した 50 のドキュメントを使用したテスト結果を報告し、回答は一般に短いため、徹底的なデコーディング手法を使用します。ビーム検索による結果の改善が見られなかったため、QA には貪欲なデコーディングを使用します。Open-MSMarco および Jeopardy の質問生成については、RAG-Token と RAG-Sequence の両方について取得した 10 のドキュメントを使用したテスト数を報告し、ベースラインとして BART ラージ モデルもトレーニングします。ビーム サイズは 4 を使用し、RAG-Sequence モデルには高速デコーディング手法を使用します。徹底的なデコーディングではパフォーマンスが向上しなかったためです。

B 人間による評価

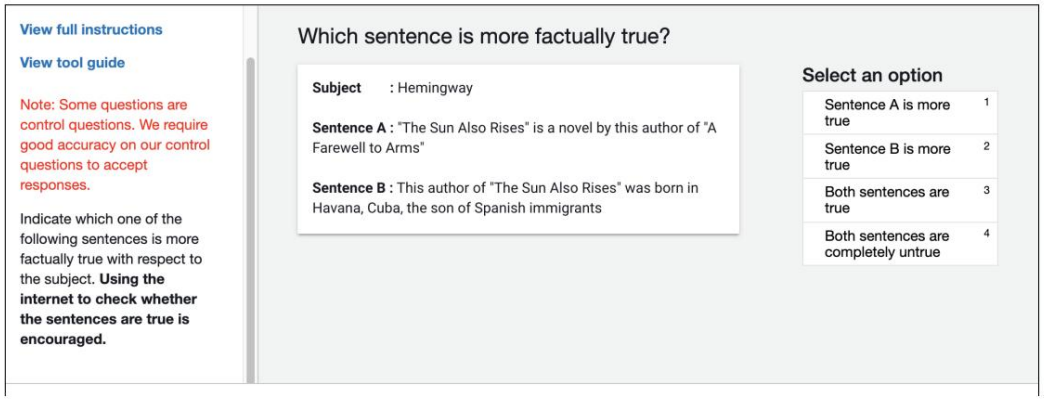

図 4: 事実性を人間が評価するための注釈インターフェイス。 [ツール ガイドの表示] をクリックすると、詳細な手順と実際の例がポップアウト表示されます。

図 4 は、人間による評価のためのユーザー インターフェイスを示しています。画面位置による偏りを避けるため、各例で文 A と文 B に対応するモデルがランダムに選択されました。
注釈者はインターネットを使用してトピックを調査するよう奨励され、完全な説明タブで詳細な説明と実例が提供されました。注釈者の正確性を評価するために、いくつかのゴールド センテンスを含めました。2 人の注釈者はこれらの例で良い成績を残せなかったため、結果から注釈が削除されました。

C トレーニング設定の詳細

すべての RAG モデルと BART ベースラインを Fairseq [45] を使用してトレーニングします。[2]トレーニングは混合精度浮動小数点演算[40] を使用して、 8 つの 32GB NVIDIA V100 GPU に分散して行いますが、トレーニングと推論は 1 つの GPU で実行できます。FAISS で最大内積探索を実行するとCPU で十分に高速であることがわかったので、ドキュメント インデックス ベクトルを CPU に保存します。これにより、Wikipedia 全体で約 100 GB の CPU メモリが必要になります。提出後、コードを HuggingFace Transformers [66]に移植しました。これは、以前のバージョンと同等のパフォーマンスを実現しますが、よりクリーンで使いやすい実装です。このバージョン[3]もオープンソースです。また、 FAISS の圧縮ツールを使用してドキュメント インデックスを圧縮し、CPU メモリ要件を 36GB に削減しました。RAG で実験を実行するためのスクリプトは、 https://github.com/huggingface/transformers/blob/master/examples/rag/README.mdにあります。 RAGモデルのインタラクティブデモはhttps://huggingface.co/rag/でご覧いただけます。

[2] https://github.com/pytorch/fairseq
[3] https://github.com/huggingface/transformers

D オープンドメインQAの詳細

オープンドメイン QA の場合、特定の質問に対して複数の回答注釈が利用できることがよくあります。これらの回答注釈は、トレーニング中に抽出モデルによって活用されます。通常、トレーニング データを準備するときに、すべての回答注釈がドキュメント内の一致を見つけるために使用されるためです。RAG の場合、各(q, a)ペアを使用してモデルを個別にトレーニングすることにより、Natural Questions と WebQuestions の複数の注釈例も活用し、精度がわずかに向上します。TriviaQA の場合、特定の質問に対して有効な回答が多数存在することがよくありますが、絵文字やスペルのバリエーションなど、トレーニング対象として適さないものもあります。TriviaQA の場合、クエリの上位 1000 ドキュメントに含まれない回答候補は除外されます。

CuratedTrecの前処理CuratedTrecの回答は正規表現の形式で与えられるため、回答生成モデルには適さないと示唆されている[20]。
これを克服するために、まず各クエリの上位 1000 件のドキュメントを取得し、正規表現パターンに最も頻繁に一致する回答を監視ターゲットとして使用する前処理手順を使用します。一致するものが見つからない場合は、単純なヒューリスティックに頼ります。つまり、各正規表現のすべての可能な順列を生成し、正規表現のネストされたツリー構造内の非決定的なシンボルを空白に置き換えます。

TriviaQA 評価のセットアップオープンドメイン QA コミュニティでは、 QA データセットのテスト データは制限されていることが多く、読解目的専用であるため、テスト データセットとして公開されている開発データセットを使用するのが通例です。私たちは、オープンドメイン QA の一般的な慣行と一致しているDPR [26]で使用されているデータセット分割を使用して結果を報告します。TriviaQA の場合、このテスト データセットは公開されているTriviaQA Web 開発分割です。Roberts ら[52]は、代わりにTriviaQA 公式 Wikipedia テスト セットを使用しました。Févry ら[14]は、Roberts ら[52]と比較するためにこの慣例に従っています([14]の付録を参照)。両方のアプローチを公平に比較できるように、両方のテスト セットの結果を報告します。
従来のオープンドメイン テスト セットではなく、公式 Wiki テスト セットを使用するとパフォーマンスがはるかに高くなることがわかりました。これは、公式 Wiki テスト セットの質問がWikipedia から簡単に回答できるためだと考えられます。

E FEVERのさらなる詳細

FEVER分類では、 [32]の手法に従い、まず主張を再生成し、次に最終的な隠れ状態の表現を使用して分類し、最後に文書間で周辺化してクラス確率を取得します。FEVERタスクには伝統的に2つのサブタスクがあります。1つ目は、主張を「支持」、「反駁」、「情報不足」のいずれかに分類することです。これは、メインの論文で検討するタスクです。FEVERのもう1つのサブタスクは、分類予測を裏付ける証拠としてWikipediaから文章を抽出することです。FEVERは私たちとは異なるWikipediaダンプを使用するため、このタスクに直接取り組むのは簡単ではありません。今後の作業でこれに対処したいと考えています。

F ヌル文書確率

REALM [20]に似た「ヌル文書」メカニズムを RAG に追加して、特定の入力に対して有用な情報を取得できない場合をモデル化する実験を行いました。ここでは、 k 個の文書が取得された場合、さらに空の文書を「取得」し、ヌル文書のロジットを予測してから、 k + 1 個の予測をマージナライズします。このヌル文書ロジットのモデル化を、 (i) ヌル文書の文書埋め込み、(ii) 静的に学習されたバイアス項、または (iii) ロジットを予測するニューラル ネットワークを学習することによって検討しました。これらによってパフォーマンスが向上することは確認されなかったため、簡潔にするために省略します。Open MS-MARCO では、有用な取得文書が常に取得されるとは限らないため、モデルは、取得のメリットが少ない質問に対して常に特定の文書セットを取得するように学習することがわかり、ヌル文書メカニズムは RAG には必要ない可能性があることが示唆されます。

Gパラメータ

RAGモデルには、BERTベースのクエリとドキュメントエンコーダーのトレーニング可能なパラメータが含まれています。
DPR、それぞれ1億1000万のパラメータ（ただし、ドキュメントエンコーダーは自分でトレーニングしていません）
BART-largeから406Mの訓練可能なパラメータ、406Mのパラメータ、合計626Mの訓練可能パラメータ

表7: 使用されたデータセットのインスタンス数。*このデータの非表示のサブセットが評価に使用されます

タスク	列車	開発	テスト
自然な質問	79169	8758	3611
トリビアQA	78786	8838	11314
ウェブ質問	3418	362	2033
キュレートレック	635	134	635
ジェパディ問題生成	97392	13714	26849
MS-マルコ	153726	12468	101093*
フィーバー3ウェイ	145450	10000	10000
フィーバーツーウェイ	96966	6666	6666

パラメータ。最もパフォーマンスの高い「クローズドブック」（パラメトリックのみ）オープンドメインQAモデルはT5-11Bです。
110億の訓練可能なパラメータを持つ。我々のモデルに最も近いパラメータ数を持つT5モデルは
モデルはT5-large（770Mパラメータ）であり、Natural Questions [52]で28.9 EMのスコアを達成している。
RAG-Sequence が達成する 44.5 を大幅に下回っており、強力なオープンドメイン QA パフォーマンスを得るためには、ハイブリッド パラメトリック/ノンパラメトリック モデルでトレーニング可能なパラメーターがはるかに少なくて済むことを示しています。
非パラメトリックメモリインデックスは、トレーニング可能なパラメータではなく、21Mで構成されています。
728次元ベクトルは153億の値から成り、8ビット浮動小数点数で簡単に保存できます。
メモリとディスクのフットプリントを管理するためのポイント精度。

H 取得の崩壊

予備実験では、ストーリー生成[11]などのいくつかのタスクでは、
検索コンポーネントは「崩壊」し、
入力。これらのケースでは、検索が失敗すると、ジェネレータは文書を無視することを学習し、
RAGモデルはBARTと同等の性能を発揮するだろう。崩壊は、より明示的ではない
いくつかのタスクでは事実の知識が求められる、またはターゲットシーケンスが長くなるなど、
ペレスら[46]は、検索者にとってあまり有益でない勾配を発見した。
下流のタスクのパフォーマンスを向上させるために検索コンポーネントを最適化する場合。

I データセットあたりのインスタンス数

各データセットのトレーニング、開発、テストのデータポイントの数を表 7 に示します。