E-022

RAG による生活保護質問応答システムにおけるテキスト分割手法の評価
Evaluation of Text Segmentation Methods in a RAG-based Public Assistance Question Answering System

大畠 瑞輝† 石井 雅樹† 伊東 嗣功† 堂坂 浩二†
Mizuki Ohata Masaki Ishii Hidekatsu Ito Kohji Dohsaka

1. はじめに

生活保護申請者に対応して業務を行うものをケースワーカーという．平成 28 年の福祉事務所人員体制調査[1]によると，ケースワーカーの 4 分の 1 が経験 1 年未満，3 分の 2 が経験 3 年未満である．また，ケースワーカーに必要な社会福祉主事資格の取得率は約 82%に留まっており，経験や専門知識の面で課題を抱える人材が少なくない状況がうかがえる[1]．そのため，ICT 技術を活用した業務支援は喫緊の課題となっている．特に，生活保護業務に関する膨大な手引書である「生活保護手帳」に記載された情報の中から，個別の業務に関連する情報を効率的に検索し，活用できるシステムの開発が強く求められている．

これまでの我々の取り組みとして，生活保護ケースワーカー業務支援のための質問応答システムが開発された[2]．このシステムでは質問文と応答文の対の集合である質問応答データベースを構築している．質問文は，(A)生活保護に関する問答の「問」の部分，(B)法律の専門家が作成した生活保護に関する問答・法令が回答根拠となる質問文，(C)法律の専門家が(A)と(B)の質問文を言い換えたものから構成され，応答文は質問の回答の根拠となる生活保護手帳の問答又は法令である．入力された質問文に対し，データベースから類似度の高い質問文を検索し，対応する応答文を出力する仕組みとなっており，類似文検索性能の向上のため，事前学習済みの BERT モデルを特定のタスクに特化させるファインチューニングを適用している．また，データベースの構築には，秋田県健康福祉部において独自に厚労省の開示資料等から集めたデータを利用しており，本研究においてもこのデータを活用している．しかしながら，この従来システムにはいくつかの課題が残っている．第一に，データの更新が高コストであるという点である．法令の追加や改訂が頻繁に行われる生活保護のドメインにおいて，その都度ファインチューニングを行うことは時間的にも経済的にも大きなコストを伴う．第二に，質問応答データベース中の文書の一部をそのまま出力する仕組みであるため，質問の意図に沿った自然で柔軟な回答が難しいという点が挙げられる．

本研究では，これらの課題を解決するアプローチとして，Retrieval-Augmented Generation(RAG)という技術を利用する[3]．RAG は，クエリに対して外部文書から回答の根拠となる関連情報を検索し，その情報をもとに大規模言語モデル(LLM)がテキスト生成を行うという手法である．この手法を用いることで，外部文書の変更のみでデータ更新を低コストで行うことができるようになり，生活保護手帳の全文を外部文書とすることでシステムの対応範囲を拡大し，LLM の高度な言語処理能力によって文脈に沿った自然で柔軟な回答を期待できる．RAG を生活保護ドメインに適用する際の課題としては，膨大で複雑な構造からなる文書の適切な箇所を検索する困難さや，法令に関わる内容の生成を行う上での誤情報の生成(ハルシネーション)への注意が挙げられる．本研究では，特に「該当箇所の検索の困難さ」に焦点を当てた．この検索困難性を解決するため，RAG 構築に不可欠な外部文書のテキスト分割(Chunking)において，ドメインの文書構造を考慮した手法を適用することにより，検索性能が向上し，結果として回答生成の精度向上も期待できると考えた．本研究の目的は，汎用的なテキスト分割手法である固定チャンク分割と，生活保護手帳の文書構造に合わせた提案方式について，検索・生成性能を比較することにある．これにより，膨大かつ複雑な文書において，テキスト分割をドメイン特性に合わせて行うことの有効性を検証する．

以下において，2 節では本研究と関連性の高い研究の紹介とそれに対する本研究の立ち位置の説明を行う．3 節では本研究を進めるうえでの前提技術及び実験の方法について説明を行う．4 節では，実験で得られた評価結果とその考察を行う．5 節では，本研究のまとめと今後の課題について論ずる．

2. 関連研究

財務報告書のドメインにおいて文書構造によるチャンキングの影響についてまとめた従来研究がある[4]．この研究では財務文書特有の要素分類機能を追加した Chipper モデルというエンコーダデコーダモデル(文書構造解析モデル)を採用し，2048 字未満の要素で後続と自動マージを行う．その後，「セクション」，「サブセクション」，「表」といった要素タイプと親子関係の階層情報をメタデータとして付与することでチューニング不要で構造要素の自然な長さを活用した適応的なチャンク生成を行っていた．固定サイズの分割方法と比較して,検索性能，回答整合性の向上とエラー率の減少を達成した．特に，財務に関する複雑な質問において顕著な改善がみられた．本研究とは，特定の複雑な情報を扱うドメインの検索・生成精度の向上を期待して文書構造に沿ってチャンキングを行うという面で類似している．本研究との差異として，以下の点が挙げられる．第一に，対象ドメインが日本語かつ複雑な内容である生活保護分野であるため，LLM の処理能力に影響を与えると考えられる．第二に，本研究では「生活保護手帳」の文書構造に特化したチャンキングを行っているため，類似文書に直接適用できず汎用性は制限されるが，その一方で pdf 解析の性能に依存せず，低いコストで安定して意味的にまとまったデータの分割を行うことができる．

†秋田県立大学 Akita Prefectural University

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

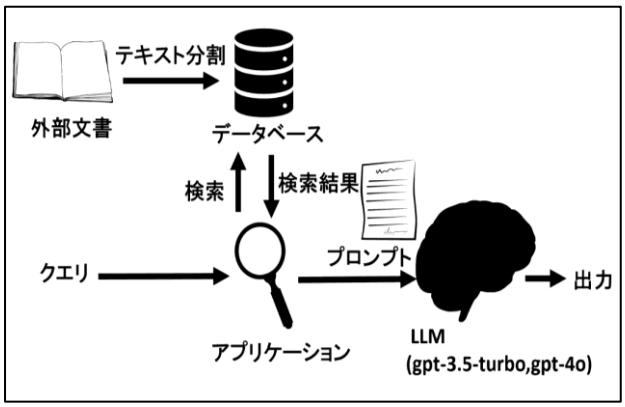

図 1　提案システムの構成

3. 方法

RAG による生活保護質問応答システムを構築するうえでの課題として，生活保護手帳が章や節に分かれた深い階層構造を持つため，質問に対して適切な箇所を検索するためには，従来の均一な分割手法だけでは不十分であるという課題がある．この課題に対して，外部文書のテキスト分割の際に文書構造に合わせて意味的なまとまりを作ることによって解決を図る．

3.1. 使用したモデル・ライブラリ

本研究は以下の動作環境で行われる．

- Python 3.10.13
- OpenAI API
  - ◆text-embedding-ada-002：埋め込みモデル
  - ◆gpt-3.5-turbo-0125，gpt-4o：出力用 LLM
- LangChain 0.3.0:LLM の機能拡張フレームワーク
  Text Splitters:LangChain 内の固定チャンク分割
  - ◆RecursiveCharacterSplitter
  - ◆RecursiveJSONSplitter
- FAISS-gpu 1.7.2:ベクトルデータベース
- RAGAS 0.1.15:RAG の評価フレームワーク

3.2. RAG

本研究では，質問応答システム開発のアプローチとして RAG という手法を利用する．近年では，ChatGPT をはじめとした LLM が発展しており需要が高まっているが，LLM は膨大な一般情報からトレーニングした情報をもとに回答を行うため，最新の情報や専門性の高いドメイン知識に対しては適切な回答を返せない場合がある．ドメイン特化の情報を追加学習させる方法としてファインチューニングが存在するが，学習コストが高く，追加学習を繰り返し行う必要があるため迅速な対応が難しいというデメリットがある．その解決策の一つとして RAG という手法があり，RAG とは LLM がテキスト生成をする際に，回答に必要な情報を外部知識源から検索を行いその情報をもとに出力を行うという手法である．RAG はファインチューニングを行って知識を追加するよりも低コストで実装でき，誤った情報を真実のように出力してしまう現象であるハルシネーションを低減する効果が報告されている．

本研究での RAG システムの構成は図 1 に示す．このシステムでは，まず JSON ファイルで構成された生活保護手帳のデジタルデータを後述するテキスト分割手法を用いてチャンクに分割する．次に，分割された各チャンクに対して，埋め込みモデルを用いてベクトル埋め込みを行い，FAISS を基盤としてベクトルデータベースを構築する．ユーザーからのクエリ(質問)が入力されると，そのクエリもベクトル化されてベクトルデータベースから類似度の高い情報を上位 k 件検索する．その後，検索された関連情報(コンテキスト)，クエリ，及び制御用の指示文をまとめたプロンプトを出力用の LLM に入力し，回答を生成させるといった流れになっている．

3.3. テキスト分割

RAG 適用における課題である「文書の階層構造に起因する適切な箇所の検索困難性」に対し，外部文書のテキスト分割を文書構造に合わせて適切に行うことで改善できると考えた．RAG では，外部文書のテキストを指定サイズにまとめて分割し，ベクトルデータベースを構築する必要がある．テキスト分割の差異は、検索時の類似度評価や検索結果のコンテキストの文脈保持に影響を与えるため，検索・生成ともに精度の向上が期待される．

本研究で扱う外部文書は，章や節，項目が深い階層構造を持つ JSON 形式の約 780 万文字のデータである．この JSON ファイルに対して，固定チャンクサイズ分割[5]による 2 種類の手法と提案方式を含む，以下の計 3 種類のテキスト分割を適用し，それぞれの性能への影響を実験によって検証する．

● Recursive split by Character (Character):
テキストを指定した文字列で分割し，指定のチャンクサイズに収まるようにまとめる最もシンプルな方法である．本研究では LangChain 内の RecursiveCharacterTextSplitter を使用した．チャンクサイズは 512 トークン，チャンクオーバーラップは 25 パーセントに当たる 128 トークンとした．これは，一般的に推奨される LLM のコンテキストウィンドウサイズと，チャンク間の連続性を考慮した値である．

● Recursive split by JSON (JSON):
JSON データの階層ごとに調べながら分割することで，ネストされたオブジェクトを保持しつつまとめる方法である．こちらも同様に LangChain 内の RecursiveJSONSplitter を使用した．マックスチャンクサイズは Character と同様の 512 トークンとした．この手法は，JSON 構造が持つ意味的なまとまりを考慮し，関連性の高い情報を同一チャンク内に維持することを目的としている．

● 提案するテキスト分割手法（提案方式）：
本研究では，生活保護手帳の文書構造に特化したテキスト分割手法を提案する．具体的には，JSON ファイル内の各 ID に含まれる以下の項目を 1 つのチャンクとして統合する．
-分類 1，分類 2：文書の階層情報
-タイトル，質問，回答：核となる質問応答情報
-本文，参考資料：補足・根拠情報
ただし，極端に長いトークンを持つ項目は除外する．この手法は，生活保護手帳において各項目が相互に強い関連性を持ち，適切な回答には質問・回答・根拠資料が一体となった情報が必要であるというドメイン特性に基づいている．固定サイズ分割では，関連する情報が複数のチャンクに分散する問題があったが，提案方式により，質問に対する回答とその根拠となる本文・参考資料が同一コンテキスト内で検索されるため，情報の散逸による検索失敗や回答に必要な情報不足を回避できると期待される．

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

あなたは与えられた文書の内容に基づいて質問に答える専門家です.
[質問]に対して，文書の情報のみから要約して詳細かつ正確に回答してください。
回答内容には質問自体やタグは含めないでください.
(文書は分類 1，分類２，タイトル，質問，回答，参考資料，本文の順でひとまとまりになっています．)(提案手法のみ)
#制約：質問に関連のある文書のみに基づいて回答してください.
[質問]：{question}
[文書]：以下の k(検索件数)項目
{context}

図 2　LLM の制御プロンプト

3.4. 評価方法

本研究の評価方法として，3.3 節で述べた 3 種類のテキスト分割を行い，それぞれ FAISS によってベクトルデータベースを構築した．各データベースに対して，検索件数を 1 件，3 件，5 件の 3 パターンと，出力用 LLM の性能による影響を確認するため gpt-3.5-turbo と gpt-4o の 2 パターンで評価を行った．評価には RAG パイプラインの評価フレームワークである RAGAS[6]を用いた．RAGAS には複数の評価指標があり，その中から Faithfulness,Answer Semantic Similarity,Context Recall,Context Precision の 4 つについて評価を行った．これらとは別に，RAGAS の代表的な評価指標として Answer Relevancy が存在するが，ほぼすべてのデータにおいて 0 を出力し，有効な評価が行えなかったため，本稿では記載を省略する．このような結果になった理由については 4 節において論ずる．各評価指標の意味は以下のとおりである.

生成に関する評価指標

- Faithfulness(F):生成された回答が与えられたコンテキストに基づいているかを測る指標
- Answer Semantic Similarity(ASS):生成された回答と正解の意味的な類似度を測る指標
- Answer Relevancy:生成された回答がプロンプトにどれだけ適切であるかを測る指標

検索に関する評価指標

- Context Recall(CR):取得されたコンテキストが正解とどの程度一致しているかを測る指標
- Context Precision(CP):コンテキスト内の正解の関連アイテムが高順位にランク付けされているか測る指標

また，LLM に与える制御プロンプトは図 2 のように設定しており，LLM が質問とコンテキストを適切に解釈して回答生成を行うように指示している．テストデータは以下の 4 種類のタスク，合計 80 件の質問と正解のペアを用意し，それぞれ評価を行った.

● 文脈理解不要なタスク

1. Simple:生活保護手帳別冊問答集から(問)と(答)のペアを question と ground_truth とした 50 件．簡単な事実確認能力を確認する目的のタスク.

● 文脈に沿った回答が必要なタスク

2. 文章抽出:生活保護手帳別冊問答集から(問)の内容を略したものをクエリとし，(答)から必要な部分を抽出したものを ground_truth とした 10 件．質問文から具体的な情報を抽出する能力を評価する目的のタスク.

表 1　全タスクの評価平均

F=Faithfulness,ASS=AnswerSemanticSimilarity,CR=ContextRecall,CP=ContextPrecision

model=gpt-3.5-turbo-0125					model=gpt-4o				
検索件数=1									
全タスク平均	F	ASS	CR	CP	全タスク平均	F	ASS	CR	CP
Character	0.511	0.901	0.300	0.679	Character	0.503	0.910	0.272	0.667
JSON	0.490	0.905	0.457	0.720	JSON	0.511	0.914	0.447	0.732
提案方式	**0.652**	**0.943**	**0.731**	**0.914**	提案方式	**0.590**	**0.948**	**0.776**	**0.941**
検索件数＝3									
全タスク平均	F	ASS	CR	CP	全タスク平均	F	ASS	CR	CP
Character	0.682	0.931	0.454	0.784	Character	0.623	0.913	0.418	0.735
JSON	0.643	0.937	0.606	0.848	JSON	0.581	0.925	0.519	0.800
提案方式	**0.695**	**0.966**	**0.856**	**0.985**	提案方式	**0.718**	**0.950**	**0.804**	**0.935**
検索件数=5									
全タスク平均	F	ASS	CR	CP	全タスク平均	F	ASS	CR	CP
Character	0.616	0.907	0.374	0.739	Character	0.651	0.916	0.382	0.744
JSON	0.581	0.918	0.559	0.806	JSON	0.573	0.928	0.598	0.777
提案方式	**0.673**	**0.941**	**0.824**	**0.928**	提案方式	**0.676**	**0.952**	**0.788**	**0.936**

表 2　Simple タスクの評価(検索件数 3 件)

Simple	F	ASS	CR	CP	Simple	F	ASS	CR	CP
Character	0.578	0.909	0.310	0.707	Character	0.612	0.918	0.326	0.730
JSON	0.580	0.923	0.574	0.822	JSON	0.549	0.935	0.581	0.837
提案方式	**0.592**	**0.955**	**0.875**	**0.989**	提案方式	**0.749**	**0.960**	**0.889**	**0.967**

表 3　文章抽出タスクの評価(検索件数 3 件)

文章抽出	F	ASS	CR	CP	文章抽出	F	ASS	CR	CP
Character	0.732	0.907	0.455	0.741	Character	**0.820**	0.907	0.425	0.717
JSON	0.599	0.899	0.450	0.700	JSON	0.731	0.924	0.475	0.700
提案方式	**0.748**	**0.931**	**0.775**	**0.900**	提案方式	0.627	**0.940**	**0.750**	**0.858**

表 4　本文タスクの評価(検索件数 3 件)

本文	F	ASS	CR	CP	本文	F	ASS	CR	CP
Character	0.799	0.904	0.620	0.933	Character	0.521	0.908	0.605	0.933
JSON	0.611	0.897	0.503	0.926	JSON	0.698	0.904	0.591	0.933
提案方式	**0.801**	**0.933**	**0.670**	**0.950**	提案方式	**0.791**	**0.933**	**0.713**	**0.950**

表 5　代名詞タスクの評価(検索件数 3 件)

代名詞	F	ASS	CR	CP	代名詞	F	ASS	CR	CP
Character	0.386	0.896	0.100	0.633	Character	**0.580**	0.900	**0.684**	0.581
JSON	0.367	0.886	0.233	0.483	JSON	0.474	0.900	0.180	0.583
提案方式	**0.508**	**0.908**	**0.267**	**0.883**	提案方式	0.577	**0.929**	0.525	**0.833**

3. 本文:生活保護手帳内の質問応答形式でないドキュメントの内容について問うクエリを作成し，その奥襟に対応した答えを ground_truth とした 10 件．自由形式の文書から情報を取得する能力の評価を目的とするタスク.
4. 代名詞:生活保護手帳別冊問答集から法令の番号や固有名詞などの LLM にとって曖昧な情報を含む ID に対するクエリと，それらを正しく言い換えたものを ground_truth とした 10 件．文脈依存性の高い表現の理解と適切な回答生成の評価を目的とするタスク.

● 全タスク平均　上記の 80 件のテストデータ平均.

4. 結果と考察

4.1　評価結果

各評価タスクの評価結果を表 1～表 5 に示す．各表の左側にある表は gpt-3.5-turbo で出力した結果，右側にある表は gpt-4o で出力した結果である．また，各表ではテキスト分割方式に対する評価指標の結果が記載されており，F は

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

Faithfulness,ASS は Answer Semantic Similarity,CR は Context Recall,CP は Context Precision を表している．太字で書かれたスコアは各指標で最も高かったものを表している．

表 1 の全タスク平均では，すべての評価指標において提案方式が最も高いスコアを記録した．特に検索に関する評価指標である Context Recall と Context Precision においては他の手法を大きく上回っている．これは，提案方式が生活保護手帳のような複雑なドメインの文書構造に適合していることが示唆されている．言語モデルによる差異は，タスクごとに特異不得意が現れたものの，総じて一貫した結果は得られなかった．本稿では紙幅の都合上，表 2~5 においては比較的スコアが良好であった検索件数 3 件の結果のみを記載することとした．表 2，表 4 のタスクにおいては全ての項目で提案方式が最も高いスコアが得られた．本稿では記載していないが，シンプルなタスクにおいては検索件数による精度の向上はあまり見られなかった．表 3 では gpt-4o の出力で Faithfulness 型の手法より低いスコアを記録するケースがみられた．特筆すべき点として，代名詞タスク(表 5)では他のタスクと比較して検索性能・回答性能ともに著しく低いスコアとなった．

4.2 考察

RAGAS の評価結果について考察する前に論ずるべきこととして，Answer Relevancy を採用しなかった理由は，評価時に 90%以上の割合で 0 と出力されてしまったことが原因である．これは，日本語かつ生活保護手帳のような複雑なドメインの評価を行う上で擬似質問生成等の内部ロジックが適切に機能しなかったことが原因と考えられる．また，Faithfulness が軒並み低い結果となっているのも類似する問題が原因であると考えられる．代替として利用した Answer Semantic Similarity は，入力文を繰り返すような回答に対しても高いスコアを出力する傾向がある．したがって，本研究における生成性能の評価は，あくまで傾向を把握するための参考値として位置づけるのが適切である．

評価結果から，検索に関する評価指標である CR と CP において，提案方式は高いスコアを獲得している．このことから，生活保護手帳のような膨大で複雑なドメインを外部文書として RAG を実装する際に，意味的なまとまりを考慮したテキスト分割を行うことで検索性能を大幅に向上させることができるといえる．タスク別の分析では，文章抽出タスクや本文タスクが Simple タスクと同程度の性能を示したのに対し，代名詞を含むタスクでは検索・生成ともに著しく低いスコアを記録した．これは，クエリとコンテキストの一致度低下による検索失敗や，代名詞の指示対象特定に必要な情報の欠落が要因と考えられる．代名詞タスクの問題解決には LLM の高度な推論能力と適切なコンテキストの関連性認識が必要であり，特に困難なタスクといえる．出力用 LLM による性能差については，タスクごとに得手不得手があったものの，全体として有意な差は見られなかった．これは，RAGAS 評価が冗長な文章を生成する傾向のある gpt-4o との相性問題が影響していると考えられる．総括すると，提案方式のテキスト分割により検索性能は大幅に向上したが，生成性能の向上は限定的であった．

5. まとめ

本研究では，生活保護ケースワーカーの負担軽減を目的として，生活保護手帳を外部文書とした RAG による質問応答システムを構築した．特に，生活保護手帳のような膨大で複雑なドメインの文書における適切な情報検索の困難性に対し，文書構造に沿ったテキスト分割を提案し，その効果を検証した．

実験結果から，提案手法は RAG の利点である LLM の高度な言語理解・生成能力を活かし，文章抽出タスクや自由形式文書からの情報検索において良好な性能を示した．一方で，代名詞を含む文脈依存性の高い質問では，Context Recall の最高値が 0.267 と低く，依然として課題が残ることが明らかになった．汎用的な固定チャンクサイズ分割との比較では，全タスク平均において Context Recall 0.856,Context Precision 0.985 を達成し，検索性能の大幅な向上を確認した．しかし，生成性能の向上は限定的であった．これらの結果から，膨大で複雑なドメインを対象とした RAG 構築において，文書構造に適応したテキスト分割は検索性能向上に効果的である一方，実用システムとしては回答生成性能のさらなる改善が必要であることが示された．

今後の課題として，以下の点が挙げられる．第一に，テストデータの拡張と評価手法の改善である．本研究のテストデータ数は限定的であり，特に文脈理解タスクの質問・正解ペアは手動作成に依存しているため，より体系的なデータ構築手法が求められる．第二に，評価手法の多角化である．RAGAS による機械的評価では，日本語の複雑な内容生成は容易ではなく，特に生成の評価指標では人的評価と乖離する結果が得られたため，複数の評価手法の併用や人的評価の導入が必要である．第三に，システムの信頼性向上である．法令関連内容を扱うシステムとして，ハルシネーション抑制機構の導入や，代名詞などの文脈依存表現に対する処理能力の向上が重要な課題となる．

謝辞

本研究を進めるにあたり，秋田県健康福祉部において独自に厚労省の開示資料等から集めたデータを提供してくださった北日本コンピューターサービス株式会社五十嵐直樹氏に感謝いたします．

参考文献

[1] 厚生労働省, “平成 28 年 福祉事務所人員体制調査について”, https://www.mhlw.go.jp/toukei/list/dl/125-1-01.pdf (最終閲覧日:2025 年 5 月 29 日).

[2] 堂坂浩二, 金子和樹, 木村幸司, 伊東嗣功, 石井雅樹: “生活保護業務支援のための質問応答システムの開発と評価”, 第 21 回情報科学技術フォーラム (2022).

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Vladimir Karpukin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela:”Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”, In:Proceedings of the 34th International Conference on Neural Information Processing System (NIPS’20), 9459-9474 (2020).

[4] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, Leah Li, “Financial Report Chunking for Effective Retrieval Augmented Generation”, arxiv:2402.05131 (2024).

[5] LangChain, Text splitters, https://python.langchain.com/docs/how_to/#text-splitters (最終閲覧日:2025 年 5 月 29 日).

[6] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert: “RAGAS:Automated Evaluation of Retrieval Augmented Generation”, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, 150-158 (2024).

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.