

# 複数の知識ベースおよびテキストを横断した問合せ

中野茉莉香<sup>†</sup> 天笠 俊之<sup>††</sup>

<sup>†</sup> 筑波大学 情報理工学位プログラム 〒 305-8573 茨城県つくば市天王台 1-1-1

<sup>††</sup> 筑波大学 計算科学研究センター 〒 305-8573 茨城県つくば市天王台 1-1-1

E-mail: <sup>†</sup>marikan@kde.cs.tsukuba.ac.jp, <sup>††</sup>amagasa@cs.tsukuba.ac.jp

あらまし 近年、事物に関する一般的な知識を構造化データとして表現し、機械処理に利用可能な知識ベースが注目され、様々な分野で利用されている。知識ベースはターゲットやドメインの異なる 60 万以上のデータセットが存在しており、複数知識ベースを横断した問合せを行う手法が提案されてきたが、語彙やスキーマの異種性を考慮できていないという課題があった。また、知識ベースは不完全であり、未知のエンティティや新規の事象に対する問合せに対応できないという問題点も挙げられる。そこで本研究では、複数の異なる知識ベースとテキストからの情報を組み合わせた横断的な問合せを行う手法を提案する。既存研究で課題となっていた語彙やスキーマの異種性に対応させるために、mediator/wrapper アプローチを用いた複数知識ベースへの分散問合せを行う。実験によって、複数の知識ベースやテキストからの情報を用いることによって、問合せ結果のカバレッジを改善できることが示された。

キーワード 知識ベース, セマンティック Web, 情報統合

## 1 はじめに

近年、事物に関する一般的な知識を構造化データとして表現し、機械処理に利用可能な知識ベースが注目され、様々な分野で利用されている。知識ベースは、数十億以上のエンティティとその関連情報をグラフによって形式的に表現する。

知識ベースは、ターゲットやドメインの異なる 60 万以上のデータセットが存在する [6]。例えば、医学や生物学等の特定分野に特化した知識ベースが多く存在する他、汎用大規模知識ベースであっても、保有する情報の粒度や語彙体系に多くの違いがある。そのため、目的の情報に関するデータを広く取得するためには、単一知識ベースに限定せず、複数知識ベースを対象に問合せを行う必要がある。

そこで、複数知識ベースを横断した問合せを行う手法が提案されている [7]。これらの手法では、複数知識ベースを横断するクエリを、そのグラフパターンを分解することによって実現している。

しかしながら、問合せを行う知識ベースはクエリ内で指定されたものに限定され、同一概念を持つ異種知識ベースへの展開を実現できていないという課題があった。知識ベースごとに保有する情報の種類や粒度は異なるため、クエリ中で指定されたエンティティから、他の異種知識ベースへの問合せを行えるようにクエリを展開する必要がある。

また、知識ベースは不完全であり、未知のエンティティや新規の事象に対する問合せに対応できないという問題点も挙げられる [12]。知識ベースの更新は頻繁ではなく、古い情報を多く含むため、新規の事象についてはニュースや SNS などの Web テキストの方が有効性が高いとされている。

そこで本研究では、複数の異なる知識ベースとテキストからの情報を組み合わせた横断的な問合せを行う手法を提案する。

既存研究で課題となっていた語彙やスキーマの異種性に対応させるために、mediator/wrapper アプローチを用いた複数知識ベースへの分散問合せを行う。また、オンデマンドにテキストからの知識抽出を行うことによって、知識ベースが保有していない未知のエンティティや事象についての情報の補完を行う。

複数知識ベースとニュースデータセットを用いた実験によって、既存の知識ベース横断問合せ手法と比較して、提案手法がよりカバレッジの高い問合せを実現していることを示した。さらに、提案手法ではユーザが知識ベース間の語彙やスキーマの異種性を意識することなく、横断的な問合せを行うことが可能であることを示した。

本稿の構成は以下の通りである。まず、2 章で本研究における前提知識について説明する。3 章では関連研究について説明する。4 章で提案手法について説明し、5 章で実験について述べる。最後に 6 章で本研究の結論と今後の方針についてまとめる。

## 2 前提知識

### 2.1 知識ベース

知識ベースとは様々な分野の知識を組織化し、蓄積したデータベースである。大規模な知識ベースの代表例としては Wikipedia 等がある。

知識ベースの中でも特に、構造化され、機械処理に利用可能な知識ベースが増えている。このような知識ベースは RDF (Resource Description Framework) (後述) のようなデータ形式を用いて論理的に一貫した形式で記述がなされている。

構造化された知識ベースの例を図 1 に示す。図 1 の知識ベースは、映画とそのディレクターを務めた人物の名前と国籍、そして映画のジャンルの関係を示す。

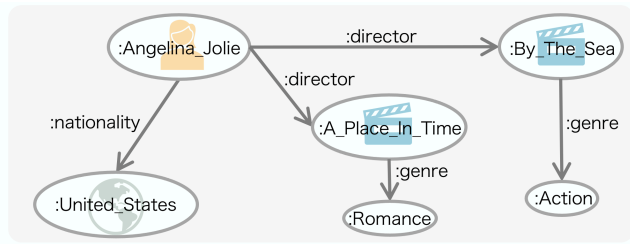


図 1 知識ベースの例

構造化された知識ベースの代表例としては、DBPedia<sup>1</sup> [2], YAGO<sup>2</sup> [3], Freebase [4], GeoNames [5] などが存在する。

## 2.2 RDF (Resource Description Framework)

知識ベース等の様々な資源を構造的に記述するためのデータ形式として、RDF(Resource Description Framework) [1] がある。RDF とは、Web 上のリソースのメタデータを記述するための枠組みである。

RDF データは最小単位として、主語 (Subject)-述語 (Predicate)-目的語 (Object) からなるトリプルを持つ。トリプルの各要素は、基本的には Web 上のデータの識別子である URI(Uniform Resource Identifier) を用いて表現される。

URI 以外の表現方法としては主語、述語、目的語の各要素ごとに制約が異なる。主語は URI または空白ノードで構成される。述語は URI を持つ。目的語は URI またはリテラルか空白ノードで構成される。リテラルとは、URI のように形式化された識別子ではなく文字列をそのまま記述したものを示す。

この RDF トリプルを組み合わせることによって RDF グラフが構成される。RDF グラフは、主語と目的語がノード、述語がエッジを示すラベル付き有向グラフとして表現される。

以後の表記として、URI を表すノードは楕円で、リテラルを表すノードは長方形で表現することとする。

## 2.3 SPARQL (SPARQL Protocol and RDF Query Language)

知識ベースから目的のデータを取得するために問合せを行う言語が、SPARQL(SPARQL Protocol and RDF Query Language) である。ユーザは SPARQL クエリで問い合わせたい特定のグラフパターンを指定することで、照合するグラフパターンを Web 上から検索することができる。

## 3 関連研究

### 3.1 テキストからの知識抽出

OpenIE [10], [11] は、テキストからの関係タブルの抽出を行う手法である。従来の情報抽出システムでは、あらかじめ抽出する語彙やドメインを指定して抽出を行うことが一般的であっ

た。しかし、OpenIE ではあらかじめ関係語彙やドメインを絞る必要がなく、あらゆるテキストから情報を抽出できるという特徴を持つ。例えば、"Obama was born in USA." というテキストが与えられた場合、OpenIE は、(Obama, was born in, USA) という関係タブルを抽出することができる。

OpenIE を用いることで、知識ベースから欠落している未知のエンティティを含めた新規の知識ベースを Web テキストから構築できる。

ここで、OpenIE によって抽出された情報は単なるテキストから構成されるタブルであり、知識ベースとは異なり、URI による正規化はなされていない。正規化がなされた知識ベースとの結合を行うためには、OpenIE によって抽出された関係トリプルをエンティティリンキング等によって正規化を行う必要がある [12], [13]。

### 3.2 複数知識ベースへの横断問合せ

知識ベースは 1 つではなく、ターゲットとするドメインや形式が異なる知識ベースが多数存在し、目的の情報を取得するためには、これら全てを対象に問合せを行う必要がある。これを実現しているのが、複数知識ベースへの横断問合せ [7] を行う手法である。

知識ベース横断問合せは SPARQL クエリに含まれるグラフパターンを分解し、各単一グラフパターンの異種の知識ベースへの問合せ結果を統合して出力する。各単一グラフパターンの問合せ先の知識ベースは、クエリに含まれるエンティティの URI から推定される。

しかし、既存の知識ベース横断問合せ手法 [8], [9] では、クエリ最適化を重視している一方で、横断問合せの結果として得られる関係トリプルが限られているという課題に対応できていない。

これは知識ベース間で語彙やスキーマに異種性があることに起因する。同一の概念を示すエンティティが、異種知識ベースにおいては異なる URI を持つエンティティとして定義されているが、これを考慮していないため得られる情報が限られていた。知識ベースごとに保有する情報の種類や粒度は異なるため、クエリ中で指定されたエンティティを他の異種知識ベースへの問合せを行えるようにクエリを展開する必要がある。

## 4 提案手法

### 4.1 提案手法の方針

本研究では、複数の異なる知識ベースとテキストからの情報を組み合わせた横断的な問合せを行うことを目的とする。

手法の方針として、既存研究で課題となっていた語彙やスキーマの異種性に対応させるために、mediator/wrapper アプローチを用いた、複数知識ベースへの分散問合せを行う。各知識ベースに対応する wrapper が SPARQL クエリの再構築や各知識ベースへの問合せを行い、その各 wrapper の問合せ結果を mediator が統合する。

また、本研究では、mediator が用いるスキーマとして普遍的

1 : <http://wikidata.dbpedia.org/>

2 : <https://yago-knowledge.org/>

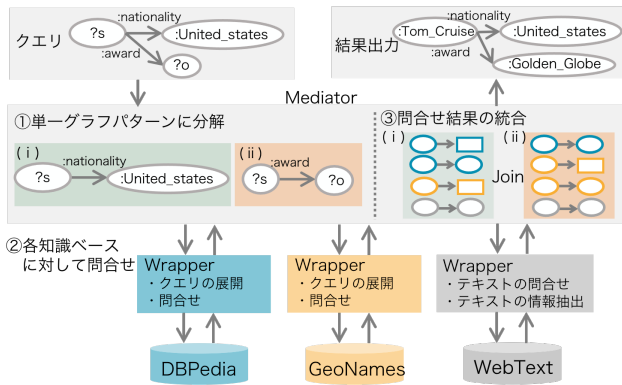


図 2 提案フレームワークの概要

な単一の mediated schema を想定する．今回は簡単のために mediated schema として DBpedia を利用するため，入力クエリは DBpedia へのクエリを想定したものとする．

## 4.2 提案フレームワーク

本節では，本研究で提案する，複数の異なる知識ベースとテキストからの情報を組み合わせた横断的な問合せを行うフレームワークについて説明する．提案フレームワークの概要を図 2 に示す．

フレームワークは 3 つのパートから成り立つ．まず，第一のステップとして，入力クエリの分解を行う．次に，第二のステップとして，分解されたそれぞれの単一グラフパターンについて，wrapper を介して各知識ベースに問合せを行う．ここで，問合せ先が YAGO などの異種の知識ベースである場合には，DBpedia へのクエリから YAGO へのクエリへの再構築と問合せを wrapper が行う．また，問合せ先が Web 上のテキストの場合には，テキストの問合せやテキストからの情報抽出を wrapper が行う．最後に，第三のステップとして，得られた各知識ベースへの単一グラフパターンの問合せ結果の統合を行い，結果を出力する．

次の項から，各ステップについて説明する．

### 4.3 クエリの分解

ユーザから与えられたクエリに含まれるマルチグラフパターンを単一グラフパターンへの分解を行う．

各単一グラフパターンについて問合せを行い，その結果を基に，次のグラフパターンの問合せを逐次的に行う．

### 4.4 各知識ベースへの問合せ

分解されたそれぞれの単一グラフパターンについて，wrapper を介して各知識ベースへの問合せを行う．各情報源に対応する wrapper は，情報源の形態や URI などを記述したメタデータを元にクエリの再構築と問合せを行う．

情報源が mediated schema である DBpedia の場合にはクエリの変換を行わずに問合せを行うことが可能となるが，異種知識ベースの場合には情報源ごとにクエリの再構築が必要となる．

Wrapper でのクエリの再構築は異種の知識ベースへの展開を行う場合と，テキストへの展開を行う場合の 2 種類に分類される．

#### 4.4.1 異種知識ベースへの展開

Mediated schema である DBpedia に対するクエリから，異なる知識ベースへのクエリの展開を行う．このために，知識ベース内に含まれる sameAs エッジを用いることでクエリの再構築を行う．sameAs エッジとは，異なる知識ベースで保有されている同一エンティティ同士を結ぶエッジであり，これを利用することで，異なるドメインの同一概念を特定することが可能となる．

#### 4.4.2 テキストへの展開

Mediated schema である DBpedia に対するクエリから，テキストへの展開を行う．テキストに含まれる関係トリプルを取得するために，テキストからクエリに関連する新規の知識ベースを構築する．Wrapper はクエリに含まれる URI のラベルを利用することでテキスト検索を行い，得られたテキストに対して OpenIE を行うことで新たなトリプルを取得する．

しかし，OpenIE を利用した新規知識ベースの構築には 2 つの問題点が挙げられる．

##### a) 表記揺れの問題

テキストに含まれる単語には多くの異なる表記が存在する．例えば，“United States”は“USA”や“America”など様々な表記が存在する．クエリに“United States”が含まれている場合にはこの全ての表記を含むテキストから情報を取得する必要がある．

そこで知識ベース内の wikipediaRedirects エッジを利用することで表記揺れ候補の取得を行う．wikipediaRedirects エッジは，Wikipedia 内でのリダイレクト関係を示す述語であり，自然言語文に含まれる多くの表記揺れ候補を捉えることが可能となる．

##### b) 述語の多様性の問題

述語はエンティティ以上に様々な表現を持つ．例えば，“:nationality”という述語の同義表現としては，“born in”，“born at”，“birthplace”などの種類が存在する．

そこで，問合せ対象の述語を持つトリプルを知識ベースから取得し，その主語と目的語を含むテキストをデータセットから抽出することで，同義表現の述語パターンを取得する処理を行う．これによって，クエリで指定された述語の同義表現を広くフォローする．

### 4.5 結果の統合

結果の統合のために，各グラフパターンの各知識ベースに対する問合せ結果の結合を行う．

結合を行う判定としては，エンティティ同士が完全に一致しているペアの他に，単語同士の類似度からも同一概念であるかの判定を行うこととする．これによって，知識ベース上に URI がまだ存在しない，テキストから新規に発見されたエンティティも結果の対象とすることが可能となる．

## 5 実 験

本実験では、提案手法による

1. 複数の知識ベースへの展開による SPARQL 横断問合せ結果のカバレッジの改善

2. テキストの情報への展開による SPARQL 横断問合せ結果のカバレッジの改善

の 2 点についての評価を行う。

表 1 評価クエリの特徴

Query ID	Type	# Triple Pattern
1-a	Chain	3
1-b	Chain	4
2-a	Star	3
2-b	Star	4
3-a	Chain-Star	4
3-b	Chain-Star	5
4-a	Chain-Star	4
4-b	Chain-Star	5
5-a	Chain-Star	4
5-b	Chain-Star	5
6-a	Chain	3
6-b	Chain	4
7-a	Star	2
7-b	Star	3
8-a	Chain	3
8-b	Chain	4
9-a	Chain	2
9-b	Chain	3
10-a	Chain-Star	4
10-b	Chain-Star	5

### 5.1 データセット

構造化知識ベースとしては、DBPedia [2] と GeoNames [5] の 2 つの知識ベースを用いる。DBPedia は Wikipedia を基に知識抽出を行うことで構築された知識ベースであり、GeoNames は全世界の地理データを収めた知識ベースである。どちらも対象とするデータのドメイン、また、語彙やスキーマの定義が異なるため、保有する情報の種類や粒度が異なる。

テキストデータセットとしては、ニューステキストから抽出された 36,000 文を持つ Reverb45K を用いる [14]。

### 5.2 実験環境

DBPedia と GeoNames への SPARQL 問合せには、それぞれ DBPedia エンドポイント<sup>3</sup>と GeoSPARQL エンドポイント<sup>4</sup>を利用する。

テキストからの知識抽出を行う OpenIE としては MinIE [15] を、得られたテキスト中のエンティティを解決するエンティティリンカーとして Stanford CoreNLP [16] を用いる。

```
SELECT ?city ?person ?unit WHERE {  
  ?city geo:ontology#name 'Kefar_Malal'.  
  ?person dbo:birthPlace ?city.  
  ?unit dbo:commander ?person.  
}
```

図 3 sameAs 関係を明示しないクエリ例 (1-a)

```
SELECT ?city ?person ?unit WHERE {  
  ?city geo:ontology#name 'Kefar_Malal'.  
  ?dbpediaCity dbo:sameAs ?city.  
  ?person dbo:birthPlace ?dbpediaCity.  
  ?unit dbo:commander ?person.  
}
```

図 4 sameAs 関係を明示したクエリ例 (1-b)

### 5.3 評価クエリ

SPARQL 横断問合せのベンチマーククエリである Fed-bench [17] を基に、2～5 個程度のトリプルパターンを持つ 10 個のクエリを作成し、これを用いて評価を行う。クエリは全て、DBPedia と GeoNames を横断する問合せであり、クエリのグラフパターンの形状として Chain, Star, Chain-Star の 3 種類が存在する。

また、既存の知識ベース横断問合せにおいては、異種知識ベース間での同一エンティティの sameAs 関係をクエリ中で指定することで知識ベース間の語彙やスキーマの異種性に対応していた。そこで本実験では、10 個のクエリそれぞれについて、(a) クエリ中に sameAs 関係を明示しない場合、(b) クエリ中に sameAs 関係を明示する場合の 2 種類のクエリを作成し、評価を行う。

評価に用いた各クエリの特徴を表 1 に示す。

また、(a)sameAs 関係を明示しないクエリの例を図 3 に、(b)sameAs 関係を明示したクエリの例を図 4 に示す。これらのクエリはどちらも、“Kefar Malal” という都市を出身とする人物とその人物が“commander”を務めた部隊について問合せを行う。図 3 のクエリでは、知識ベース間の異種性を考慮せずに記述がなされている。一方で、図 4 のクエリでは、各知識ベースから得られるトリプルパターンについてユーザが熟知し、知識ベース間での同一となるエンティティ指定を予め行なっている。

### 5.4 問合せ結果の取得数と適合率

複数の知識ベースへの展開とテキストからの情報への展開を行うことによる問合せ結果のカバレッジの変化を観察するため、既存の知識ベース横断問合せの手法と提案手法での、問合せ

3 : <https://dbpedia.org/sparql>

4 : <http://www.geosparql.org/>

表 2 (a) クエリ中に sameAs 関係を明示しないクエリ, (b) クエリ中に sameAs 関係を明示したクエリ, (a)+(b) のクエリにおける問合せ結果の合計取得数と適合率

	(a)		(b)		(a)+(b)	
	Precision	#Extractions	Precision	#Extractions	Precision	#Extractions
既存の知識ベース横断問合せ手法	-	0	100.0%	185	100.0%	185
提案手法	96.6%	240	96.6%	240	96.6%	480

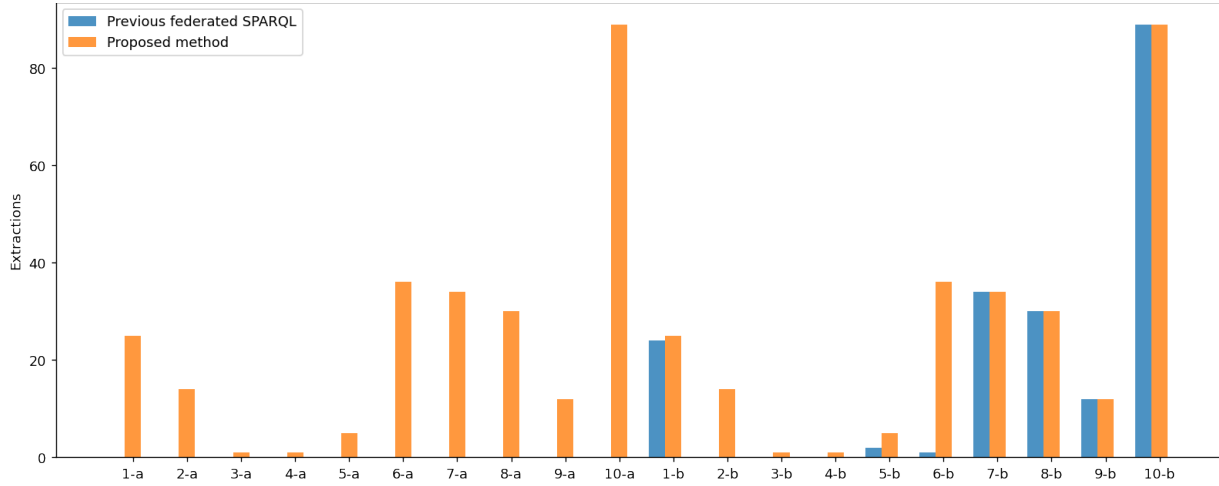


図 5 クエリごとの問合せ結果取得数

せ結果の取得数とその精度を評価する。上記に示した 20 クエリに対して問合せを行なった結果の取得数と適合率を表 2 に示す。表 2 は, (a) クエリ中に sameAs 関係を明示しないクエリ, (b) クエリ中に sameAs 関係を明示したクエリ, (a)+(b) のクエリにおける問合せ結果の合計取得数と適合率を示す。

Precision は, 問合せ結果がクエリに適する, 事実関係が正しいトリプルであった割合を示しているため, テキストからの情報を含まず, 知識ベースに対してのみ問合せを行なっている既存の知識ベース横断手法では, 100%となる。

まず, 既存の知識ベース横断問合せの手法では (a) の sameAs 関係を明示しないクエリによる問合せの場合に, 異種知識ベース間の同一エンティティの特定を行うことができないため, 結果が取得できていない。一方で提案手法では, クエリ中のエンティティの他の知識ベースへの展開を行うことによって, (a) の sameAs 関係を明示しないクエリと (b) sameAs 関係を明示したクエリのどちらでも, 同等の問合せ結果を得られている。このことから, 提案手法ではユーザが知識ベース間のスキーマの異種性を意識することなく, 横断的な問い合わせを行うことが可能だと言える。

また, (b) のクエリ中に sameAs 関係を明示する場合においても, 提案手法は既存の知識ベース横断問合せの手法よりも 55 個以上の関係を, 高い Precision を保ったまま取得できている。提案手法では, テキストからの情報に対してもオンデマンドに問合せを行うことで, 知識ベースにまだ存在しない情報の補填

を行うことが可能となっている。

### 5.5 クエリごとの問合せ結果取得数

20 クエリのそれぞれに対する, 問合せ結果の取得数を図 5 に示す。クエリごとの結果からも, (a) の sameAs 関係を明示しないクエリの場合に, 既存の知識ベース横断問合せの手法では問合せ結果が得られていない一方で, 提案手法では (b) の sameAs 関係が明示されたクエリと同等の結果が得られていることが分かる。また, (b) クエリ中に sameAs 関係を明示する場合においては, テキストからの情報に対してもオンデマンドに問合せを行うことによって, (b) の sameAs 関係が明示されたクエリの多くが, 既存の知識ベース横断問合せの手法よりも多くの結果を得られていることが読み取れる。例として, "Kefar Malal" という都市を出身とする人物とその人物が"commander"を務めた部隊について問合せを行う 1-b のクエリでは, DBPedia 上にまだ存在していない (Ariel Sharon, was commander of, Unit 101) という関係をニューステキスト中から新たに抽出したことにより, 既存の知識ベース横断問合せ手法以上の結果を取得できている。

## 6 まとめと今後の方針

本研究では, 複数の異なる知識ベースとテキストからの情報を組み合わせた横断的な問合せを行う手法を提案した。

提案手法では, mediator/wrapper アプローチを用いた複数

知識ベースの分散問合せを行うことで、既存研究で課題となっていた、語彙やスキーマの異種性への対応を実現した。また、入力クエリに応じて新規の関連トリプルを Web テキストから抽出することによって、知識ベースの不完全性の補完を行なった。

実験により、複数の知識ベースやテキストからの情報を用いることによって、問合せ結果のカバレッジを改善できることが示された。さらに、提案手法ではユーザが知識ベース間の語彙やスキーマの異種性を意識することなく、横断的な問合せを行うことが可能であることを示した。

今後の方針としては、テキストを含む複数の知識ベースへの問合せの展開を行なった場合の問合せの効率化を行うことを考えている。現在は、各クエリについて、問合せを行うことができる環境にある全ての知識ベースに対してクエリの展開を行う。

しかし、実際に問合せに関連するトリプルを保有する知識ベースは全体の一部であり、全知識ベースにクエリ展開を行う現在の手法は冗長であると言える。そこで、過去の問合せの統計量等を用いることによって、クエリ展開を行うか否かを定める閾値を定めることによって、問合せの最適化を行うことを予定している。

## 謝 辞

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務 (JPNP20006) の結果得られたものです。

## 文 献

- [1] Jeremy J. Carroll and Graham Klyne. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
- [2] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722-735). Springer, Berlin, Heidelberg.
- [3] Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics* 6.3 (2008): 203-217.
- [4] Bollacker, Kurt, et al. Freebase: a collaboratively created graph database for structuring human knowledge. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 2008.
- [5] GeoNames. GeoNames. <http://geonames.org/>. Retr. June 17, 2009.
- [6] Valdestilhas, André, Tommaso Soru, and Muhammad Saleem. More Complete Resultset Retrieval from Large Heterogeneous RDF Sources. *Proceedings of the 10th International Conference on Knowledge Capture*. 2019.
- [7] Schwarte, Andreas, et al. Fedx: Optimization techniques for federated query processing on linked data. *International semantic web conference*. Springer, Berlin, Heidelberg, 2011.
- [8] Charalambidis, Angelos, Antonis Troumpoukis, and Stasinou Konstantopoulos. SemaGrow: Optimizing federated SPARQL queries. *Proceedings of the 11th International Conference on Semantic Systems*. 2015.
- [9] Saleem, Muhammad, et al. CostFed: Cost-based query optimization for SPARQL endpoint federation. *Procedia Computer Science* 137 (2018): 163-174.

- [10] Etzioni, Oren, et al. Open information extraction from the web. *Communications of the ACM* 51.12 (2008): 68-74.
- [11] Niklaus, Christina, et al. A survey on open information extraction. *arXiv preprint arXiv:1806.05599* (2018).
- [12] Nguyen, Dat Ba, et al. Query-driven on-the-fly knowledge base construction. *Proceedings of the VLDB Endowment* 11.1 (2017): 66-79.
- [13] Lin, Xueling, et al. KBPearl: a knowledge base population system supported by joint entity and relation linking. *Proceedings of the VLDB Endowment* 13.7 (2020): 1035-1049.
- [14] Vashishth, Shikhar, Prince Jain, and Partha Talukdar. Cesi: Canonicalizing open knowledge bases using embeddings and side information. *Proceedings of the 2018 World Wide Web Conference*. 2018.
- [15] Gashtevski, Kiril and Gemulla, Rainer and del Corro, Luciano. MinIE: Minimizing Facts in Open Information Extraction. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017.
- [16] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.
- [17] Schmidt, Michael, et al. Fedbench: A benchmark suite for federated semantic data query processing. *International Semantic Web Conference*. Springer, Berlin, Heidelberg, 2011.