

Linked Open Data 環境における
メタデータ記述語彙の類似度算出手法

筑波大学

図書館情報メディア研究科

2018 年 3 月

二十歩 亮介

目次

1.	はじめに	1
2.	メタデータの共有方法と相互運用性	2
2.1.	Linked Open Data	2
2.2.	メタデータスキーマ	5
2.2.1.	メタデータ語彙定義	5
2.2.2.	メタデータ記述規則定義	8
2.3.	メタデータの相互運用性	10
3.	メタデータ記述語彙の探索と問題	12
3.1.	メタデータ記述語彙の探索	12
3.2.	メタデータ記述語彙の探索における問題	16
4.	メタデータ記述語彙の類似度算出手法	17
4.1.	メタデータ記述語彙の類似度算出手法の提案	17
4.1.1.	タームのベクトル化手法の提案	17
4.1.2.	タームの類似度算出手法の提案	19
4.2.	メタデータ記述語彙の類似度算出手法の実現	23
5.	評価実験	28
5.1.	評価手法と実験	28
5.2.	実験結果	34
6.	考察	37
7.	関連研究	41
8.	おわりに	42
	謝辞	43
	参考文献	44

図目次

図 1	2009 年版 LOD cloud diagram ([4]より引用)	3
図 2	2011 年版 LOD cloud diagram ([4]より引用)	3
図 3	2017 年版 LOD cloud diagram ([4]より引用)	4
図 4	RDF グラフの例	4
図 5	RDFS による foaf:Agent と foaf:interest の定義	8
図 6	簡易 DSP の記述例	10
図 7	DCAP におけるメタデータ記述規則定義作成の概要	11
図 8	青空文庫メタデータ語彙定義のメタデータ	13
図 9	aozora:reviser の語彙定義	13
図 10	vocab.cc における foaf:interest の統計情報	14
図 11	LOV のターム探索における検索結果	15
図 12	プロパティの類似度算出手法の概要	21
図 13	クラスの類似度算出手法の概要	22
図 14	語彙定義から取得した foaf:interest の情報	24
図 15	LOD のデータセットから取得した foaf:interest の情報	25
図 16	抽出した代表語と周辺情報	26
図 17	foaf:interest との類似度上位 10 件のタームと類似度	27
図 18	プロパティ評価セット 1 を用いた実験の結果	34
図 19	プロパティ評価セット 2 を用いた実験の結果	34
図 20	プロパティ評価セット 3 を用いた実験の結果	35
図 21	クラス評価セット 1 を用いた実験の結果	35
図 22	クラス評価セット 2 を用いた実験の結果	36
図 23	クラス評価セット 3 を用いた実験の結果	36
図 24	ラベルのみを利用した類似度算出の実験結果	39

表目次

表 1	DCMES のプロパティの一覧.....	6
表 2	FOAF のタームの一部.....	7
表 3	Basic Geo Vocabulary のタームの一覧.....	7
表 4	各情報の記述に使用されるプロパティ	24
表 5	gensim による学習時の設定	27
表 6	各評価セットの X , A , B , C の例	31
表 7	プロパティの類似度算出における各種重みと ICF の閾値の組み合わせ	32
表 8	クラスの類似度算出における各種重みと ICF の閾値の組み合わせ	32
表 9	各パターンの説明	33
表 10	各プロパティ評価セットにおける類似度の平均の比較	38
表 11	ベクトル化できなかった単語の分類.....	40

1. はじめに

誰でも利用、再利用、再配布できるデータ[1]が様々な組織、コミュニティによって多様な形式で作成され Web 上に公開されている^{1,2}。なかでも、標準化されたデータ表現とアクセス方式、そして他のデータへのリンクにより相互運用性や発見性に優れた Linked Open Data (LOD) 環境での公開が望まれている。そのための標準としてしばしば Resource Description Framework (RDF) が利用される。RDF においてデータの記述にはメタデータ語彙を使用する。メタデータ語彙とは特定のドメインにおけるデータ記述に使用する用語をまとめたものである。用語にはデータ項目を記述するためのプロパティと記述対象自身の分類を記述するためのクラスがあり、プロパティとクラスを総称してタームという。メタデータ語彙は誰でも作成することができ、現在、様々な目的を持った多数のメタデータ語彙が Web 上に公開されている。RDF によるデータ記述においてはそれらの公開されている既存のメタデータ語彙からタームを選択し再利用することで作成されたデータの相互運用性を高めることができる。タームの探索は Linked Open Vocabularies (LOV) に代表されるメタデータ語彙探索支援システムを用いて行われることが多い。それらのシステムが提供する検索環境により利用目的に応じたタームの候補を発見することが可能だが、検索キーワードによっては適切なタームが候補に含まれるとは限らない。ここで、利用目的に完全に合致せずとも類似するタームから関連するタームを参照することができれば、適切なタームの発見を支援することが可能であると考えられる。しかしタームはメタデータ語彙ごとに独立して定義されており、関連するタームを参照することが難しい。

本研究は関連するタームを参照可能にすることを目的として、タームの類似度算出手法を提案する。本手法はタームの URI および語彙定義における名前と説明文からタームを特徴付ける単語である代表語を抽出し、それらの分散表現を比較することでタームの類似度を算出する。本論文ではまず 2 章で LOD 環境におけるデータ共有に関する基礎知識とデータの相互運用性について述べる。3 章でメタデータ語彙の探索と探索における問題について述べ、4 章で問題を解決するためのタームの類似度算出手法を提案、実現する。5 章で提案手法の評価実験、6 章で考察を述べ、7 章で関連研究を紹介する。最後に 8 章で本研究のまとめを述べる。

¹ <http://datahub.io/>

² <http://www.data.go.jp/>

2. メタデータの共有方法と相互運用性

2.1. Linked Open Data

標準化されたデータ表現およびアクセス方式と他のデータへのリンクによってデータの相互運用性と発見性を高める技術や取り組みを Linked Data[2]という。以下は Linked Data の基本原則である（[3]より引用）。

1. *Use URIs as names for things.*
2. *Use HTTP URIs so that people can look up those names.*
3. *When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).*
4. *Include links to other URIs, so that they can discover more things.*

Linked Data の基本原則に従い、かつオープンライセンスであるデータを Linked Open Data (LOD) という。LOD のデータセットは様々な分野で作成されており、Web 上に公開されるデータセットの数は年々増加の一途を辿っている。図 1、図 2、図 3 はそれぞれ 2009 年、2011 年、2017 年における LOD のデータセットのリンク関係を示す LOD cloud diagram[4]である。各ノードが 1 つのデータセットを表し、エッジがリンク関係を表している。また、ノードの色によって出版、地理、政治、生命科学などデータセットが作成された分野を表している。2009 年には 89 件しかなかったデータセットが、2011 年には 295 件、最新版の 2017 年には 1163 件と急激に増加していることがわかる。

LOD の実現にはしばしば Resource Description Framework (RDF) [5]が利用される。RDF は Web 上で識別可能なすべての事物をリソースとし、そのリソースに関するデータを記述するための枠組みである。RDF ではデータを主語、述語、目的語の三つ組みによって表現する。この三つ組みはトリプルと呼ばれ、RDF における記述の最小単位である。複数のトリプルを結合して RDF グラフを形成することにより、複雑なデータを作成環境や分野を問わず論理的に表現することができる。例えば図 4 は「<http://mdlab.slis.tsukuba.ac.jp>」を主語、「タイトル」を述語、「メタデータ研究室」を目的語とするトリプルと、「<http://mdlab.slis.tsukuba.ac.jp>」を主語、「分類」を述語、「文書」を目的語とするトリプルを結合した RDF グラフであり、「<http://mdlab.slis.tsukuba.ac.jp> のタイトルはメタデータ研究室であり、文書というグループに属する」ということを表している。

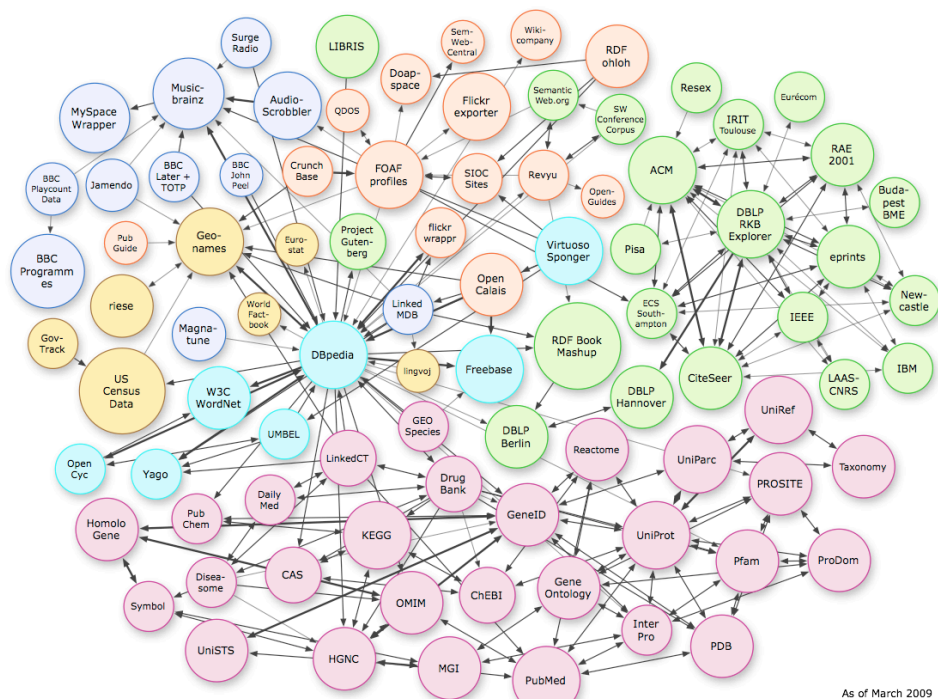


図 1 2009 年版 LOD cloud diagram ([4]より引用)

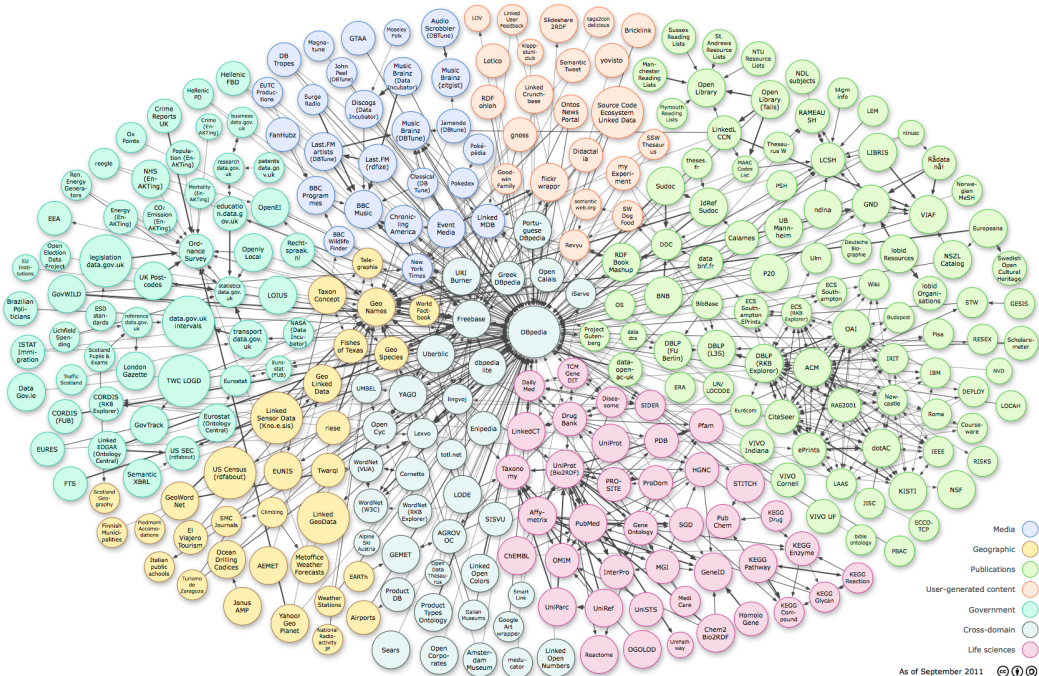


図 2 2011 年版 LOD cloud diagram ([4]より引用)

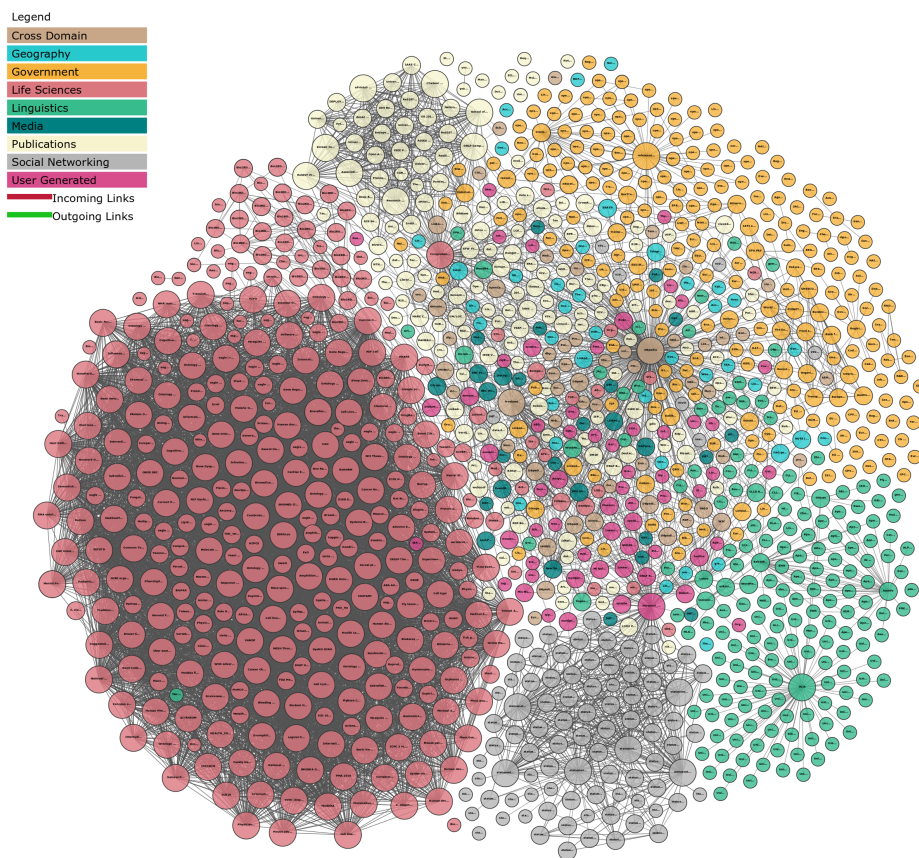


図 3 2017 年版 LOD cloud diagram ([4]より引用)

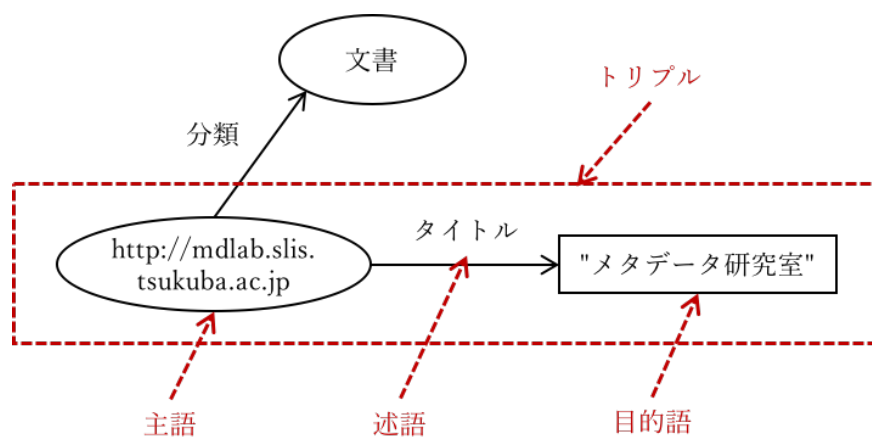


図 4 RDF グラフの例

2.2. メタデータスキーマ

メタデータとは「データについてのデータ」を表す言葉であり、記述対象となるリソースに対してそのリソースが持つ一つ以上の属性（記述項目）と属性値を記述したものである。メタデータの身近な例として書籍の奥付などに記される書誌情報がある。書誌情報とは書籍のタイトルや著者、発行者、ISBNなどを記載したものであり、書誌情報が提供されることによって書籍の管理や発見が容易になる。メタデータはあらゆるところに存在するが、本研究では「RDFによって記述された機械判読可能なデータ」をメタデータとして扱う。

メタデータスキーマはメタデータを記述する際の規則を定義したものである。メタデータとして記述する項目、記述に使用する用語、記述する値に対する制約などを事前に定義することで、メタデータの作成、利用、バージョン管理などが容易になる。メタデータスキーマはメタデータの記述に用いる語彙の仕様を定義するメタデータ語彙定義、メタデータを記述する際の規則を定義したメタデータ記述規則定義から構成される。

2.2.1. メタデータ語彙定義

RDFはデータの表現形式であるため、実際のメタデータ記述のための用語を別に用意する必要がある。メタデータ記述のための用語集をメタデータ記述語彙、あるいはメタデータ語彙という。メタデータ語彙には記述対象の属性、つまりRDFのトリプルにおける述語を記述するためのプロパティターム（以下、プロパティ）と、記述対象自身の分類を表すクラスターム（以下、クラス）が含まれる。また、プロパティとクラスを総称してタームという。例えば、図4における「タイトル」「分類」はプロパティ、「文書」はクラスとしての役割を持つ。ここで「タイトル」は自然言語で表現されており、人間であれば国語辞典の語義や、主語や目的語などの文脈から語の意味を理解できる。しかし計算機がRDFグラフを処理する場合、「タイトル」を作品の題名を意味するプロパティとして扱うのか、あるいは称号を意味するプロパティとして扱うのかを判断できない。そこでメタデータ語彙はタームに対してURIを与えることでタームの役割を一意に識別している。このURIで表現されたタームの仕様を定義したものがメタデータ語彙定義である。

現在、様々な目的を持ったメタデータ語彙が各組織、コミュニティによって定義され、Web上に公開されている。表1は分野を問わずメタデータの記述に用いる語彙であるDublin Core Metadata Element Set (DCMES) [6]のプロパティの一覧、表2は人物や組織に関するメタデータの記述に用いる語彙であるFriend of a Friend Vocabulary (FOAF) [7]

のタームの一部、表 3 は地理的実体に関するメタデータの記述に用いる語彙である Basic Geo Vocabulary[8]のタームの一覧である。ただし、タームの名称には名前空間接頭辞を用いている。例えば、FOAF のタームの URI に共通する名前空間である「<http://xmlns.com/foaf/0.1/>」を「foaf:」に置き換えている。この場合「<http://xmlns.com/foaf/0.1/Person>」は「foaf:Person」と表記される。本論文では以降もタームの表記に名前空間接頭辞を用いる。

表 1 DCMES のプロパティの一覧

名称*	説明
dc:contributor	リソースの内容に寄与している個人、団体等
dc:coverage	リソースが関わる空間的、あるいは時間的な範囲や領域
dc:creator	リソースの内容の作成に主たる責任を持つ個人、団体等
dc:date	リソースのライフサイクルにおける出来事に関する時間や期間
dc:description	リソースの内容の説明
dc:format	リソースの物理的形態、あるいはデジタル表現形式
dc:identifier	リソースへの曖昧さのない参照
dc:language	リソースの言語
dc:publisher	リソースの公開に責任を持つ個人、団体等
dc:relation	関連するリソース
dc:rights	リソースの権利に関する情報
dc:source	派生元のリソース
dc:subject	リソースの主題
dc:title	リソースに与えられた名前
dc:type	リソースの性質、あるいはジャンル

*dc:は <http://purl.org/dc/elements/1.1> の名前空間接頭辞

表 2 FOAF のタームの一部

名称*	タームの種類	説明
foaf:Agent	クラス	人物, グループ, ソフトウェア, 人工物など
foaf:Person	クラス	人物. 存命中, 故人, 実在, 架空を問わない
foaf:Organization	クラス	組織
foaf:Document	クラス	文書
foaf:name	プロパティ	事物の名前
foaf:title	プロパティ	称号 (Mr, Mrs, Ms, Dr. など)
foaf:knows	プロパティ	(相互に) 知っている人物
foaf:interest	プロパティ	関心を持っていることに関する文書

*foaf:は <http://xmlns.com/foaf/0.1/> の名前空間接頭辞

表 3 Basic Geo Vocabulary のタームの一覧

名称*	タームの種類	説明
geo:SpatialThing	クラス	空間的な広がりを持つ実体
geo:Point	クラス	座標系を用いて表される地点
geo:lat	プロパティ	緯度
geo:long	プロパティ	経度
geo:lat_long	プロパティ	カンマで区切られた緯度と経度
geo:alt	プロパティ	標高
geo:location	プロパティ	地点, または他の地理的実体との関係

*geo:は http://www.w3.org/2003/01/geo/wgs84_pos# の名前空間接頭辞

メタデータ語彙定義は計算機が処理できるよう RDF によって記述される。記述には W3C が RDF Vocabulary Description Language として推奨している RDF Schema (RDFS) [9] というメタデータ語彙を用いる。メタデータ語彙定義では各タームに対して人間が解釈可能な名前と説明であるラベルとコメントの記述が推奨されている。またプロパティの場合、主語が取り得るクラスであるドメイン、目的語が取り得るクラスであるレンジを定義することが多い。RDFS ではラベル、コメント、ドメイン、レンジの記述に用いるプロパティである `rdfs:label`, `rdfs:comment`, `rdfs:domain`, `rdfs:range` を提供している。図 5 は RDFS による `foaf:Agent` と `foaf:interest` の定義を RDF グラフで表現したものである。定義から

foaf:Agent の名前が「Agent」であり、人物やグループ、ソフトウェア、人工物などを表すクラスであることがわかる。また foaf:interest の名前が「interest」であり、対象の人物が関心を持っていることに関するページを記述するプロパティであることがわかる。さらにドメインとレンジから foaf:Agent によって分類されるリソースの属性の記述に用いられ、その値が foaf:Document によって分類されるリソースであることがわかる。

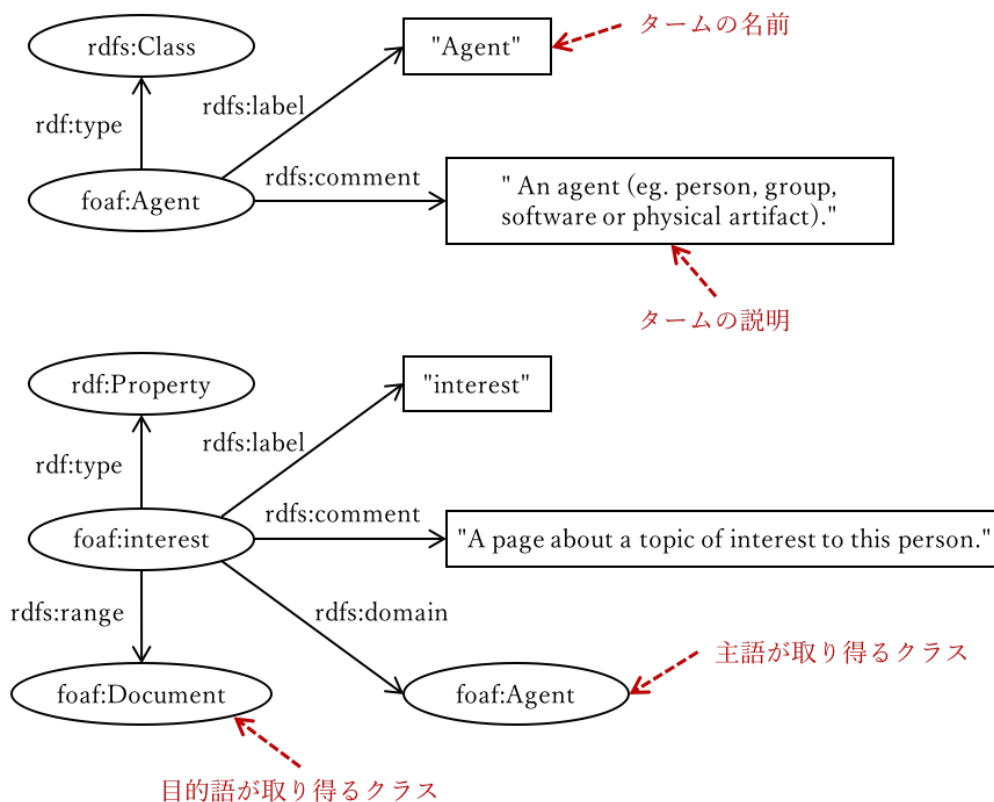


図 5 RDFS による foaf:Agent と foaf:interest の定義

2.2.2. メタデータ記述規則定義

メタデータの記述対象とその記述対象が持つメタデータ記述項目、各項目を記述する際の制約を定めたものをメタデータ記述規則定義という。メタデータを作成する場合、メタデータ記述規則定義を参照することで作成者によるメタデータの不揃いを緩和することができる。またメタデータを利用する場合、メタデータ記述規則定義を参照することでメタデータの内容の正しい理解の支援となる。

メタデータ記述規則定義における各記述項目に対する代表的な制約を以下に挙げる。

- 記述項目名
 - 記述項目を識別するための名称。例えば「title」「author」など。
- ターム
 - 記述に用いるターム。例えば書籍の題名に dc:title を用いるなど。
- 省略可能性
 - 項目が必須であるか任意であるか。例えば書籍の題名は必須、発行者は任意など。
- 繰り返し条件
 - 項目が繰り返し記述可能であるか。例えば書籍の題名は繰り返し記述不可、著者は繰り返し記述可能など。
- 値制約
 - 文字列や数値などのデータ型、従うべきシンタックス、統制語彙などの項目値に対する制約。例えば書籍の発行日として記述する値のデータ型は xsd:date であるなど。デフォルト値や選択肢を用意する場合もある。

メタデータ記述規則定義の標準的な記述形式として Dublin Core Metadata Initiative (DCMI) が提唱する Description Set Profile (DSP) [10]がある。DSP はメタデータ記述規則定義を機械判読可能な XML や RDF で記述することができる。また、メタデータ情報基盤構築事業がメタデータや XML, RDF に関する知識が乏しい人でも DSP を作成できるよう、簡易 DSP[11]を提唱している。簡易 DSP は DSP の標準化された表形式の表現方法であり、テンプレートとして用意された表を埋めるだけで DSP を作成できる。図 6 は簡易 DSP の記述例である。メタデータの記述対象である書籍と著者の各記述項目に対して記述の際の制約を与えている。定義から書籍と著者の分類の記述に schema:Book, foaf:Person を用いること、書籍の題名は必須で繰り返し記述不可であり記述に dc:title を用いること、またその値は文字列であること、書籍の著者は任意で繰り返し記述可能であり記述に schema:author を用いること、またその値は foaf:Person により分類されるリソースであること、著者の名前は必須で繰り返し記述不可であり記述に foaf:name を用いること、またその値は文字列であることがわかる。

[MAIN]

ラベル	ターム	最小	最大	値タイプ	値制約	説明
ID	schema:Book	1	1	ID		書籍の ID
title	dc:title	1	1	文字列		書籍の題名
author	schema:author	0	-	構造化	構造化著者	書籍の著者

[構造化著者]

ラベル	ターム	最小	最大	値タイプ	値制約	説明
ID	foaf:Person	1	1	ID		著者の ID
name	foaf:name	1	-	構造化		著者の名前

図 6 簡易 DSP の記述例

2.3. メタデータの相互運用性

メタデータに対する要求は利用目的によって異なり、メタデータスキーマは特定のコミュニティやアプリケーションごとに作成されることになる。そのため、異なるメタデータスキーマに従い作成されたメタデータを組み合わせて利用することが困難な場合がある。この問題に対し DCMI は Dublin Core Application Profile (DCAP) [12]を提唱している。DCAP はあるアプリケーションのためのメタデータ作成上の必要事項を定めたドキュメント集である。DCAP ではメタデータ記述規則定義を作成する際に、既存のメタデータ語彙から適切なタームを選択し、再利用することでメタデータの相互運用性の向上を図ると同時に、メタデータ語彙の不要な増加を防いでいる。

図 7 は DCAP におけるメタデータ記述規則定義作成の概要を表している。既存のメタデータ語彙である DCMES と FOAF から各項目の記述に用いるタームとして適切なタームを選択し、再利用することでメタデータ記述規則定義を作成している。メタデータ記述規則定義 X, Y に従い作成されたメタデータは、共通のタームである foaf:name を介して相互運用が可能である。

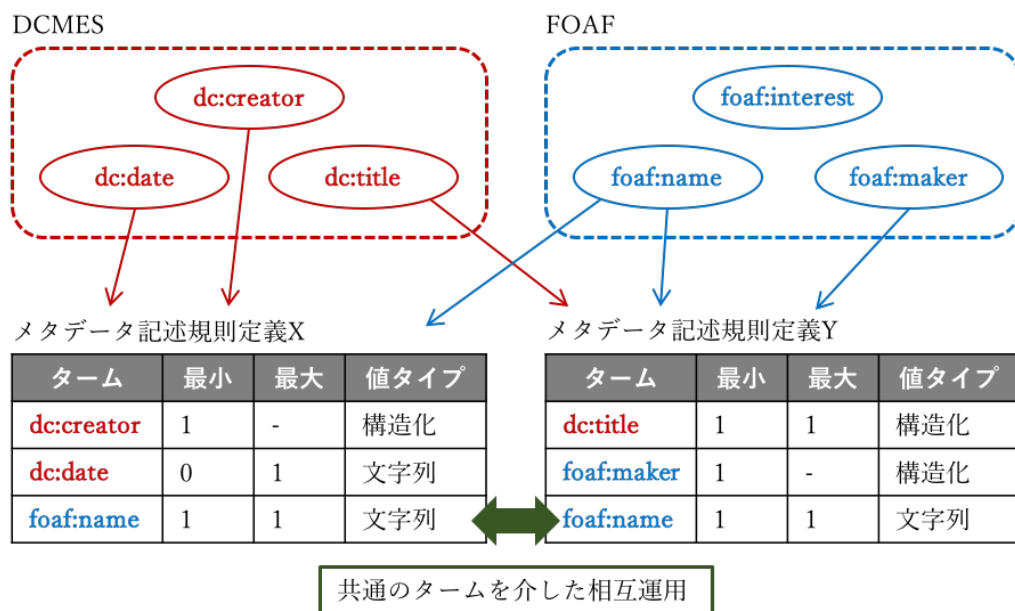


図 7 DCAP におけるメタデータ記述規則定義作成の概要

3. メタデータ記述語彙の探索と問題

3.1. メタデータ記述語彙の探索

2.3 節で述べた通り，メタデータ記述規則定義を作成する際に既存のメタデータ語彙から適切なタームを選択し再利用することでメタデータの相互運用性を高めることができる．タームを選択する際に確認すべきことを以下に示す．

(a) タームの用法

そのタームを用いることでメタデータの記述対象や記述項目を過不足なく表現できるか確認する．もし不適切なタームを選択してしまった場合，利用目的にそぐわないメタデータが作成されたり，第 3 者によるメタデータの利用が困難になったりする．用法を確認するにはタームの語彙定義を参照する必要がある．

(b) メタデータ語彙の管理状況・来歴情報

そのメタデータ語彙は適切に維持されているか，また作成者や作成日などのメタデータが記述されているか確認する．タームの URI が参照解決できない，作成者が明確でないといった場合，そのメタデータ語彙のタームを用いること自体が適切でないことがある．

(c) メタデータ語彙の利用状況

そのメタデータ語彙は広く利用されているか確認する．利用されていないメタデータ語彙のタームを選択してしまった場合，共通のタームによる相互運用ができなくなる．

(a) と (b) に関してはメタデータスキーマレジストリを利用することで確認できる．メタデータスキーマレジストリとは人間や計算機が様々なコミュニティによって作成されたメタデータスキーマを利用できるよう，メタデータスキーマを蓄積，公開する仕組みである．代表的なメタデータスキーマレジストリとして DCMI によって標準化されたメタデータ語彙定義を管理する The Dublin Core Metadata Registry³，主にメタデータ語彙定義提供者を対象に語彙定義の公開とバージョン管理を支援する Open Metadata Registry⁴，メタデ

³ <http://dcmi.kc.tsukuba.ac.jp/dcregistry/>

⁴ <http://metadataregistry.org/>

ータ語彙定義とメタデータ記述規則定義の管理に加えて作成支援機能を備えた MetaBridge⁵がある。図 8 は MetaBridge に登録されている青空文庫メタデータ語彙定義のメタデータ、図 9 は青空文庫メタデータ語彙のタームである aozora:reviser の語彙定義である。語彙定義のバージョン、作成者、作成日や、タームのラベル、コメント、ドメイン、レンジなどが確認できる。図は Web ブラウザによる閲覧のため表形式の表示だが、API を用いることにより RDF で記述されたメタデータスキーマを取得することができる。

名前空間(接頭辞)	aozora※標準の接頭辞
名前空間(URI)	http://purl.org/net/aozora/
バージョン	http://purl.org/net/aozora/?20130109 過去のバージョン
バージョン情報 (reg:version)	1
タイトル (reg:title)	青空文庫メタデータ語彙定義
コメント (reg:comment)	青空文庫の書誌データのために作成した語彙定義 (LODチャレンジ提出用)
作成者 (reg:creator)	筑波大学
登録者 (reg:registrant)	
作成日 (reg:created)	2013-01-09
分類 (reg:tag)	
キーワード (reg:keyword)	
権利情報 (reg:rights)	
アイコン画像 (reg:icon)	
その他	

図 8 青空文庫メタデータ語彙定義のメタデータ

語彙定義URI	http://purl.org/net/aozora/
語彙定義タイトル	青空文庫メタデータ語彙定義
名前空間(URI)	http://purl.org/net/aozora/reviser
バージョン	http://purl.org/net/aozora/?20130109
ラベル (rdfs:label)	校訂者
コメント (rdfs:comment)	校訂者を示す
定義域 (rdfs:domain)	http://purl.org/net/aozora/BibResource 詳細表示
値域 (rdfs:range)	http://purl.org/net/aozora/Person 詳細表示
タイプ (rdf:type)	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property 詳細表示

図 9 aozora:reviser の語彙定義

⁵ <https://www.metabridge.jp>

(c) に関してはメタデータ語彙についての統計情報を利用することで確認できる。代表的な統計情報として LOD cloud diagram に登録されているデータセットについての統計情報を提供する State of the LOD cloud[13], 旧 DataHub⁶, Data.gov⁷, PublicData.eu⁸など各種データカタログサイトに登録されているデータセットについての統計情報を提供する LODStats[14], Billion Triples Challenge Dataset⁹で使用されているメタデータ語彙についての統計情報を提供する vocab.cc[15]がある。図 10 は vocab.cc における foaf:interest の統計情報である。データセット全体で 2,810,540 回使用されていること、その使用回数は統計に含まれるプロパティにおいて 75 番目であること、そして使用回数でなく使用されているデータセット数の場合の情報が確認できる。

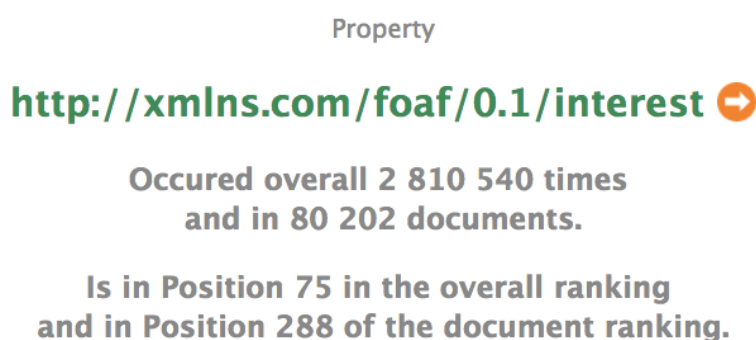


図 10 vocab.cc における foaf:interest の統計情報

このように既存のメタデータ語彙から適切なタームを選択するためには様々な事項を確認する必要がある。しかし、存在するすべてのタームに対して上記の確認を行うことは現実的ではない。そこで、利用目的に応じて確認を行うタームを候補として絞り込む必要がある。その際にはメタデータ語彙探索支援システムを利用する。メタデータ語彙探索支援システムは、複数のメタデータ語彙定義に対しての検索環境を提供するものである。多くの場合、メタデータスキーマレジストリはスキーマに対する検索機能を有するが、よりメタデータ語彙の探索に特化したシステムとして Linked Open Vocabularies (LOV) [16]がある。LOV は Web 上に公開されたメタデータ語彙を独自に収集し、それらに対するキーワード検索とファセット検索を提供している。LOV には 2018 年 1 月現在、627 件のメタデータ

⁶ <https://old.datahub.io/>

⁷ <https://www.data.gov/>

⁸ <http://publicdata.eu/>

⁹ <http://km.aifb.kit.edu/projects/btc-2012/>

語彙が登録されており、その語彙定義では合計 59,622 件のタームが定義されている。キーワード検索ではタームの URI や語彙定義におけるラベル、コメント、そのタームを定義しているメタデータ語彙のラベル、コメントに対する全文検索が行われる。ファセット検索ではタームのタイプ（クラス/プロパティ）、メタデータ語彙、メタデータ語彙に付与されたタグで絞り込みを行う。検索結果はキーワードが一致した部分と LOD のデータセットにおける使用回数によってスコア付けされ、ランキング表示される。図 11 は LOV のターム探索においてキーワードを「interest」、タームのタイプを「property」とした場合の検索結果の一部である。

The screenshot displays the LOV (Linked Open Vocabulary) search interface. At the top, a search bar contains the keyword 'interest', with a red dashed box and arrow indicating it as the search keyword. Below the search bar, a sidebar on the right shows the 'Type' filter set to 'property/class' and 'property (341)' selected. The main area displays a list of search results, with a red dashed box highlighting the first four entries. A red dashed arrow points to the bottom of this list, indicating that terms are ranked by score.

Results	Term	Score
341 results	foaf:interest (foaf) 11 occurrences in 1 LOD datasets http://xmlns.com/foaf/0.1/interest rdfs:comment A page about a topic of interest to this person. rdfs:label interest localName interest	0.654
	dbpedia-owl:interest (dbpedia-owl) n/a (use in LOD) http://dbpedia.org/ontology/interest rdfs:label interest @en localName interest	0.551
	akt:addresses-generic-area-of-interest (akt) 3,172,911 occurrences in 8 LOD datasets http://www.aktors.org/ontology/portal#addresses-generic-area-of-interest localName addresses-generic-area-of-interest	0.515
	dbpedia-owl:mainInterest (dbpedia-owl) n/a (use in LOD) http://dbpedia.org/ontology/mainInterest rdfs:label main interest @en localName mainInterest	0.469

Tags: Time (107), Press (74), eBusiness (35), People (25), Geography (17), Health (15)

図 11 LOV のターム探索における検索結果

3.2. メタデータ記述語彙の探索における問題

メタデータ語彙探索支援システムが提供する検索環境を利用することで利用目的に応じたタームの候補を発見できるが、検索キーワードによっては適切なタームが候補に含まれているとは限らない[17][18]。現在、タームに付与された URI や語彙定義におけるラベルなど、タームの名付けに関する明確な指針は確認できない。また語彙定義におけるタームの説明文であるコメントに関しても必要最小限しか記述されていなかったり、例などを交えて詳細に記述されていたりとタームによってばらつきがある。そのため、検索キーワードと利用目的に合ったタームの URI や語彙定義に使用されている単語が異なることがあり、その場合目的のタームを発見できない。ここで、利用目的に完全に合致せずとも類似するタームから関連するタームを参照することができれば、適切なタームの発見を支援することができると思われる。関連するとは例えばプロパティの場合、そのプロパティによって記述されるメタデータ項目が「著者」と「筆者」のように類似していたり、「日付」に対する「作成日」「更新日」のように特殊化であったり、「作者」「発行者」に対する「貢献者」のように一般化であったり、「作者」と「作成物」のように反対の役割を持っていたり、「組織の名前」と「プロジェクトの名前」のように記述対象は異なるが同じ役割を持っていたり、「地理的実体の緯度」と「地理的実体の経度」のように記述対象が同じで似た役割を持っていたりすることである。

しかしタームはメタデータ語彙ごとに独立して定義されており、関連するタームを参照することが難しい。そのため、ターム間の関連の有無や強弱を判断する基準が必要である。本研究は関連するタームを参照可能にするためのタームの類似度算出手法を提案する。

4. メタデータ記述語彙の類似度算出手法

4.1. メタデータ記述語彙の類似度算出手法の提案

メタデータ記述語彙として公開されているタームの類似度を算出する方法として、タームに付与された URI や語彙定義におけるラベルに使用されている単語の文字列の類似度を利用するものがある。文字列の類似度算出にはレーベンシュタイン距離やジャロ・ウィンクラ距離が用いられる。しかしこれらの手法はあくまで文字列の比較であり、単語の意味を考慮していない。文字列の比較以外の方法としてはタームの語彙定義を文書として扱い、情報検索の分野において文書の類似度算出によく用いられる TF-IDF とコサイン類似度を利用するものがある。しかしこの手法は語彙定義に使用されている単語が一致していないと類似度が高くなる。そこで本研究はタームの名前や語彙定義に使用されている文字列や単語そのものではなく、その意味に基づき類似度を算出するため分散表現を利用する。

分散表現とは単語を数百次元のベクトルで表現する手法である。近い意味を持つ単語を近いベクトルで表現することにより、ベクトル間の演算を単語の意味の演算に対応させることができる。また、ベクトルの加減算が単語の意味の加減算となる性質を加法構成性という。加法構成性を持つ分散表現を生成するツールとして Word2Vec[19]がある。本研究はこの Word2Vec を用いてタームをベクトル化し、類似度を算出する。提案手法はタームのベクトル化手法と類似度算出手法から構成される。

4.1.1. タームのベクトル化手法の提案

まず、分散表現を利用するためにタームを単語の集合で表現する。本研究ではこの集合に含まれる単語をタームの代表語（以下、代表語）と呼ぶ。代表語はタームの URI と語彙定義から抽出する。この際、タームの名前を記述するために使用されている代表語とタームの説明を記述するために使用されている代表語ではタームの類似度における重要度が異なると考えられる。そのため抽出源ごとに代表語の集合を作成する。抽出源はローカル名、ラベル、コメントの 3 種類である。

ローカル名はタームに付与された URI における名前空間を除いた文字列を指す。例えば、`http://xmlns.com/foaf/0.1/interest` のローカル名は `interest`、`http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing` のローカル名は `SpatialThing` である。ローカル名は `interest` のように 1 つの単語で構成される場合と、`SpatialThing` のように複数の単語で構成される

場合がある。複数の単語で構成される場合、それらを一纏めにする方法はキャメルケース、スネークケース、ケバブケースの3つがある。キャメルケースは SpatialThing のように各単語の一文文字目を大文字にする方法、スネークケースは spatial_thing のように単語間をアンダースコアで繋ぐ方法、ケバブケースは spatial-thing のように単語間をハイフンで繋ぐ方法である。ローカル名が1つの単語で構成される場合はその単語をそのまま代表語とし、複数の単語で構成される場合は分割してそれぞれの単語を代表語とする。ただしケバブケースの場合、well-known のようにハイフンを含めて意味を成している場合がある。そこでケバブケースの分割は、ハイフンを含めて意味を成している場合は分割せずに代表語とし、そうでない場合はハイフンを削除し連結した単語が意味を成すか調べ、意味を成す場合はその単語を代表語とし、意味を成さない場合は分割してそれぞれの単語を代表語とする。意味を成すかの判断は外部の概念辞書を用いて行う。本手法ではこのキャメルケース、スネークケース、ケバブケースを分割する処理を複合語分割と呼ぶ。

ラベル、コメントは語彙定義におけるタームの名前と説明の記述を指す。それぞれ自然言語で表現されるが、複数の単語が一纏めにされた表現が含まれれば複合語分割を行う。名前を表すラベルは類似度算出において重要度が高く、また多くの場合1～数単語で構成されるためすべての単語を代表語とする。一方で説明を表すコメントは名前と比べて重要度が低いと考えられ、また文章であるため類似度算出に不必要な単語が多く含まれる。そこで、コメントとして記述された文章に対して形態素解析を行い、名詞、動詞、形容詞の単語のみを代表語とし、さらに重み付けを行う。重み付けには Inverse Comment Frequency (ICF) を定義する。ICF はあるタームのコメントの代表語として抽出された単語が、同じメタデータ語彙のタームのコメントの代表語として抽出されているほど小さくなる。つまり同じメタデータ語彙において多くのタームのコメントの記述に使用されている単語は、そのタームを特徴付けるものでないとし、重みを小さくしている。ICF は2を底とするコメントの総数を対象の単語を含むコメントの数で割った値の対数である。

代表語はすべてステミングを行い、抽出源ごとに重複を取り除く。ステミングとは語形の変化を除去して原形にする処理である。

次に、抽出した代表語からタームのベクトル化に使用する代表語を選出する。ローカル名とラベルはともにタームの名前に関する代表語を持つ。ローカル名とラベルの代表語抽出は、ローカル名とラベルの代表語が全く同じである、ローカル名よりラベルの代表語の数が多いまたは詳細な単語である、ローカル名が通し番号のような意味を持たない代表語でラベルの代表語は自然言語である、語彙定義にタームの名前の記述がなくローカル名の代表語のみが抽出されているといった4パターンに大別される。そこでラベルの代表語が抽出

されている場合はラベルのみを使用し、ローカル名の代表語を使用するのはラベルの代表語が全く抽出されていない場合のみとする。コメントに関しては、ICF の値が閾値を越えた代表語のみを使用する。

最後に Word2Vec を用いて選出した代表語をベクトル化する。それらのベクトルの演算によりタームのベクトルを求める。あるターム t のラベルから抽出した代表語のベクトルを a_i 、ベクトルの数を M 個、コメントから抽出した代表語のベクトルを b_j 、ベクトルの数を N 個としたとき、 t のベクトル $V(t)$ を以下の式で定義する。ただし α_1 、 α_2 は 0.0 から 1.0 の値を取り、 α_1 、 α_2 の合計は 1.0 である。

$$V(t) = \alpha_1 * \sum_{i=1}^M a_i + \alpha_2 * \sum_{j=1}^N b_j \quad (4.1)$$

4.1.2. タームの類似度算出手法の提案

タームの類似度算出手法をプロパティとクラスそれぞれの場合に分けて提案する。

本研究ではタームの名前と説明の記述をまとめて基礎記述と呼ぶ。プロパティの類似度を求める場合、そのプロパティの基礎記述以外にもドメインとレンジを考慮する必要がある。基礎記述が示すプロパティの役割が同じでも、ドメインやレンジが異なれば同等のプロパティではないためである。しかし基礎記述、ドメイン、レンジのいずれかでも似ているならば、プロパティ自体にも何らかの関連があると考えられる。そこでプロパティの類似度は基礎記述の類似度、ドメインの類似度、レンジの類似度を個別に計算し、重み付けした合計とする。

ドメインとレンジは語彙定義から取得するが、プロパティによっては定義されていない、あるいは owl:Thing のようなすべての事物と定義されているが実際のデータセットでは特定のクラスのリソースを記述対象や値として持つことが多い場合がある。そこで Web 上に公開されている LOD のデータセットを収集し、データセット内で当該プロパティが使用されている例を確認できれば、そのトリプルの主語のクラスをドメイン、目的語のクラスをレンジとして扱う。しかしその実例が少なければそのプロパティの用法は間違っている、あるいは一般的でない可能性がある。そのため、データセットからドメイン、レンジとして取得したクラスには重み付けを行う。重み付けには Dataset Frequency (DF) を定義する。DF はあるタームの特定の用法が確認できるデータセットの数である。ここでの DF はあるク

ラスが当該プロパティのドメイン、レンジとして記述されているデータセットの数である。DF が閾値を越えたクラスのみを当該プロパティのドメイン、レンジとして採用する。

また、語彙定義とデータセットからドメインを取得できない場合、当該プロパティを含むメタデータ語彙が定義しているすべてのクラスをドメインとして扱う。これはメタデータ語彙が扱う分野はそのメタデータ語彙のクラスによって表され、そのメタデータ語彙のプロパティの記述対象はそのクラスのいずれかであると考えたためである。

図 12 はプロパティの類似度算出手法の概要である。プロパティの基礎記述、ドメイン、レンジの類似度を個別に計算し、重みを付けて足すことで最終的な類似度を算出する。ドメインの類似度はドメインとして取得したクラスの基礎記述の類似度である。ドメインとして取得したクラスが複数ある場合、すべてのクラスの組み合わせで類似度を計算し、最も高い類似度をドメインの類似度として扱う。レンジについても同様である。類似度の計算にはコサイン類似度を利用する。コサイン類似度はベクトルのなす角のコサインであり、ベクトルの近さを表す尺度である。提案するタームのベクトル化手法は分散表現を用いており、タームのベクトルの近さがタームの意味の近さに対応する。したがってコサイン類似度がタームの類似度として機能する。コサイン類似度は-1.0 から 1.0 の値をとり、1.0 に近いほど類似度が高い。ある D 次元のベクトル p, q のコサイン類似度 $\cos(p, q)$ は以下の式で表される。

$$\cos(p, q) = \frac{\sum_{i=1}^D p_i q_i}{\sqrt{\sum_{i=1}^D p_i^2 \sum_{i=1}^D q_i^2}} \quad (4.2)$$

プロパティ A, B の基礎記述のベクトルを A_{des}, B_{des} 、類似度が最大となるドメインのクラスのベクトルを A_{domain}, B_{domain} 、類似度が最大となるレンジのクラスのベクトルを A_{range}, B_{range} としたとき、プロパティ A, B の類似度 $PropSim(A, B)$ を以下の式で定義する。ただし、 $\beta_1, \beta_2, \beta_3$ は 0.0 から 1.0 の値をとり、 $\beta_1, \beta_2, \beta_3$ の合計は 1.0、ドメインとしてクラスが取得できなかった場合 β_2 を、レンジとしてクラスが取得できなかった場合 β_3 を β_1 に足す。またそれぞれのコサイン類似度が負の値になった場合は 0.0 として扱う。

$$\begin{aligned} PropSim(A, B) = & \beta_1 * \cos(A_{des}, B_{des}) \\ & + \beta_2 * \cos(A_{domain}, B_{domain}) \\ & + \beta_3 * \cos(A_{range}, B_{range}) \end{aligned} \quad (4.3)$$

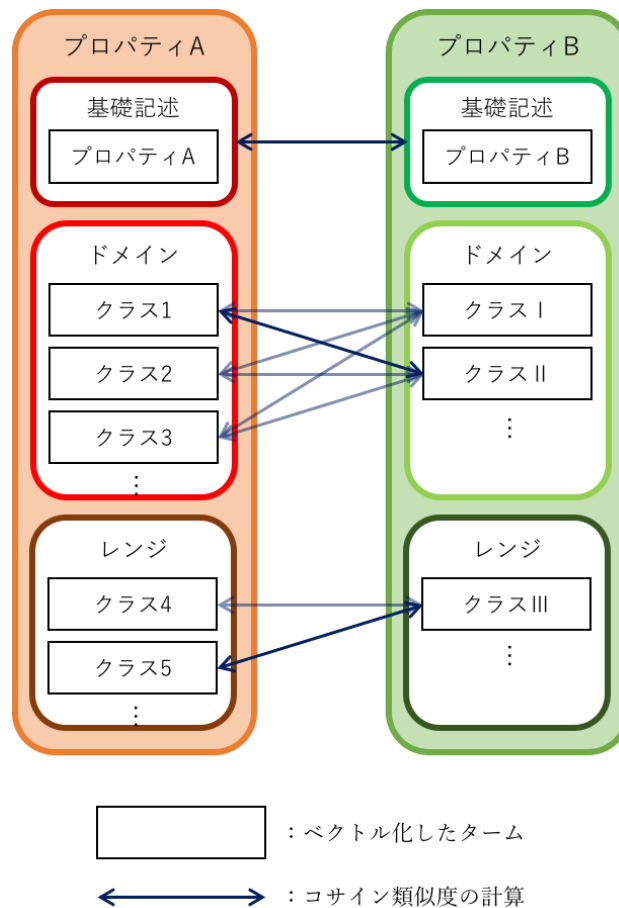


図 12 プロパティの類似度算出手法の概要

クラスの類似度算出では基礎記述に加え、当該クラスをドメインとして持つプロパティ（以下、クラス共起プロパティ）を利用する。クラス共起プロパティは語彙定義から取得するが、プロパティのドメイン、レンジと同様の理由で LOD のデータセットからも取得する。データセット内で当該クラスが使用されている例を確認できれば、そのクラスによって分類されるリソースを主語とするトリプルの述語であるプロパティをクラス共起プロパティとして扱う。データセットからクラス共起プロパティとして取得したプロパティには DF によって重み付けを行い、閾値を越えたプロパティのみをクラス共起プロパティとして採用する。ここでの DF はあるプロパティが当該クラスのクラス共起プロパティとして記述されているデータセットの数である。

図 13 はクラスの類似度算出手法の概要である。クラスの基礎記述、クラス共起プロパティの類似度を個別に計算し、重みを付けて足すことで最終的な類似度を算出する。クラス共

起プロパティの類似度はクラス共起プロパティとして取得したプロパティの基礎記述の類似度である。クラス共起プロパティとして取得したプロパティが複数ある場合は、プロパティの類似度算出手法におけるドメイン、レンジの類似度の計算とは異なり、すべてのプロパティの基礎記述のベクトルの総和をクラス共起プロパティのベクトルとして扱い類似度の計算を行う。クラス A , B の基礎記述のベクトルを A_{des} , B_{des} , クラス共起プロパティのベクトルの総和を A_{prop} , B_{prop} としたとき、クラス A , B の類似度 $ClassSim(A, B)$ を以下の式で定義する。ただし、 γ_1 , γ_2 は 0.0 から 1.0 の値をとり、 γ_1 , γ_2 の合計は 1.0, クラス共起プロパティとしてプロパティが取得できなかった場合 γ_2 を γ_1 に足す。またそれぞれのコサイン類似度が負の値になった場合は 0.0 として扱う。

$$ClassSim(A, B) = \gamma_1 * \cos(A_{des}, B_{des}) + \gamma_2 * \cos(A_{prop}, B_{prop}) \quad (4.4)$$

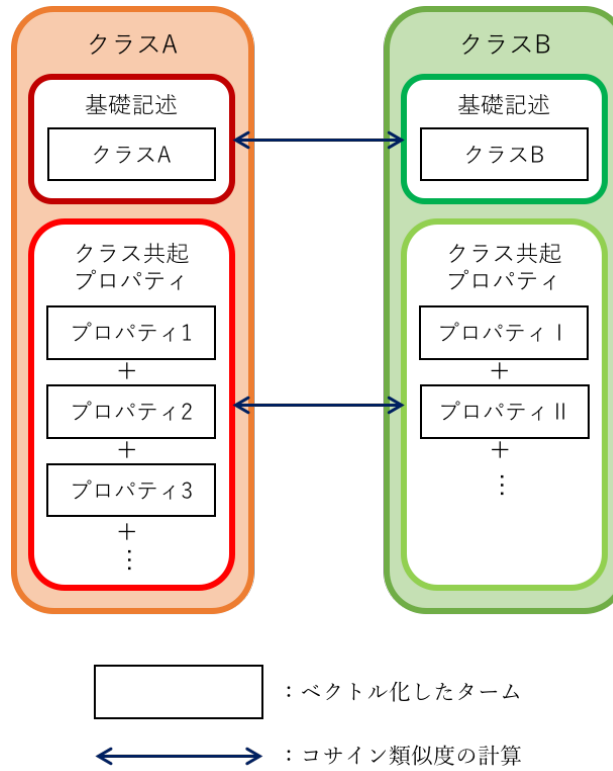


図 13 クラスの類似度算出手法の概要

4.2. メタデータ記述語彙の類似度算出手法の実現

プログラミング言語 Python を用いて提案した類似度算出手法を実現する。類似度算出の対象は LOV に登録されているメタデータ語彙のタームとする。

まず、メタデータ語彙定義と語彙定義の情報の拡充に使用する LOD のデータセットを取得する。語彙定義は LOV が公開しているもの、データセットは 2014 年版の Billion Triples Challenge Dataset¹⁰の一部を利用した。Billion Triples Challenge Dataset は研究者などが実験に使用するデータセットを収集するコストを軽減するため、独自に収集したデータセットを公開しているものである。取得した語彙定義とデータセットは RDF ストアへ格納し、RDF の問い合わせ言語である SPARQL[20]による検索を可能にする。

続いて対象とする各タームの語彙定義とデータセットから必要な情報を取得する。必要な情報とはローカル名、ラベル、コメントに加え、プロパティの場合はドメインとレンジ、クラスの場合はクラス共起プロパティである。語彙定義においてそれぞれの情報の記述に使用されるプロパティを表 4 のように定めた。ローカル名はタームの URI における最後の"/"か"#以降の文字列、クラス共起プロパティはドメインの記述に使用されるプロパティを述語として持つトリプルの目的語が当該クラスであった場合の主語のプロパティである。取得する情報の値が文字列であり、複数の言語で記述されている場合は英語の情報を取得した。またデータセットにおいて当該プロパティが使用されていた場合、そのプロパティを述語とするトリプルの主語、目的語のクラスをそれぞれドメイン、レンジとし、当該クラスが使用されていた場合、そのクラスにより分類されるリソースを主語とするトリプルの述語のプロパティをクラス共起プロパティとする。

図 14 は語彙定義から取得した情報、図 15 は LOD のデータセットから取得した情報を JSON 形式で出力した例である。図 15 におけるドメイン、レンジの URI をキーとするハッシュの値はその URI で識別されるクラスが当該プロパティのドメイン、レンジとして記述されているデータセットの数である。

¹⁰ <http://km.aifb.kit.edu/projects/btc-2014/>

表 4 各情報の記述に使用されるプロパティ

情報の種類	プロパティ
ラベル	http://www.w3.org/2000/01/rdf-schema#label
コメント	http://www.w3.org/2000/01/rdf-schema#comment http://www.w3.org/2004/02/skos/core#definition http://purl.org/dc/terms/description
ドメイン	http://www.w3.org/2000/01/rdf-schema#domain http://schema.org/domainIncludes
レンジ	http://www.w3.org/2000/01/rdf-schema#range http://schema.org/rangeIncludes

```
"http://xmlns.com/foaf/0.1/interest": {
  "local_name": "interest",
  "label": "interest",
  "comment": "A page about a topic of interest to this person.",
  "domain": [
    "http://xmlns.com/foaf/0.1/Agent"
  ],
  "range": [
    "http://xmlns.com/foaf/0.1/Document"
  ],
  "type": "property"
}
```

図 14 語彙定義から取得した foaf:interest の情報


```

"http://xmlns.com/foaf/0.1/interest": {
  "domain": {
    "http://xmlns.com/foaf/0.1/Agent": 4,
    "http://xmlns.com/foaf/0.1/Project": 248,
    "http://www.w3.org/2006/vcard/ns#VCard": 4,
    "http://xmlns.com/wot/0.1/SigEvent": 1,
    "http://xmlns.com/wot/0.1/User": 1,
    "http://xmlns.com/foaf/0.1/Person": 363,
    "http://rdfs.org/sioc/ns#UserAccount": 721,
    "http://purl.org/dc/terms/Agent": 6,
    "http://www.w3.org/2006/vcard/ns#Individual": 3,
    "http://xmlns.com/foaf/0.1/Group": 1,
    "http://xmlns.com/foaf/0.1/Document": 2724,
    "http://schema.org/Person": 6,
    "http://xmlns.com/foaf/0.1/PersonalProfileDocument": 2
  },
  "range": {
    "http://xmlns.com/foaf/0.1/PersonalProfileDocument": 1,
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag": 1,
    "http://xmlns.com/foaf/0.1/Organization": 1,
    "http://xmlns.com/foaf/0.1/Document": 7,
    "http://xmlns.com/foaf/0.1/OnlineAccount": 1,
    "http://rdfs.org/sioc/ns#User": 1
  }
}

```

図 15 LOD のデータセットから取得した foaf:interest の情報

次に取得した情報から代表語を抽出する。4.1.1 項で述べたように複合語分割、形態素解析、ステミングを行う。複合語分割におけるケバブケースの分割ではハイフンで繋がれた複合語がハイフンを含めて意味を成すか、またハイフンを削除し連結した単語が意味を成すか調べる必要がある。今回は英語の概念辞書である WordNet¹¹に登録されている場合に意味を成すとした。形態素解析とステミングには形態素解析ツールである TreeTagger¹²を用いた。TreeTagger は分かち書きされた文章を入力することで、各単語の品詞と原形を出力する。品詞は独自の品詞コードで表現される。コメントに関しては名詞を表す NN, NNS, 動詞を表す VV, VVD, VVG, VVN, VVP, VVZ, 形容詞を表す JJ, JJR, JJS を品詞コードとして持つ単語を代表語として採用する。ローカル名、ラベルに関しては品詞に関係なく

¹¹ <http://wordnet.princeton.edu/>

¹² <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

全ての単語を代表語として採用するが、記号や数字などを表す LS, SENT, SYM, CD, :, \$,), (を品詞コードとして持つものは除外する。図 16 は代表語として抽出した単語を JSON 形式で出力した例である。ドメイン、レンジなど URI で表現されるものはそのまま記述しており、データセットから得た情報を統合している。コメントの代表語をキーとするハッシュの値はその単語の ICF、データセットから取得したドメインの URI をキーとするハッシュの値はその URI で識別されるクラスの DF である。ICF は対象となっているメタデータ語彙での最大値、DF は対象となっているタームでの最大値で正規化しているため、ともに最大値は 1.0 となる。

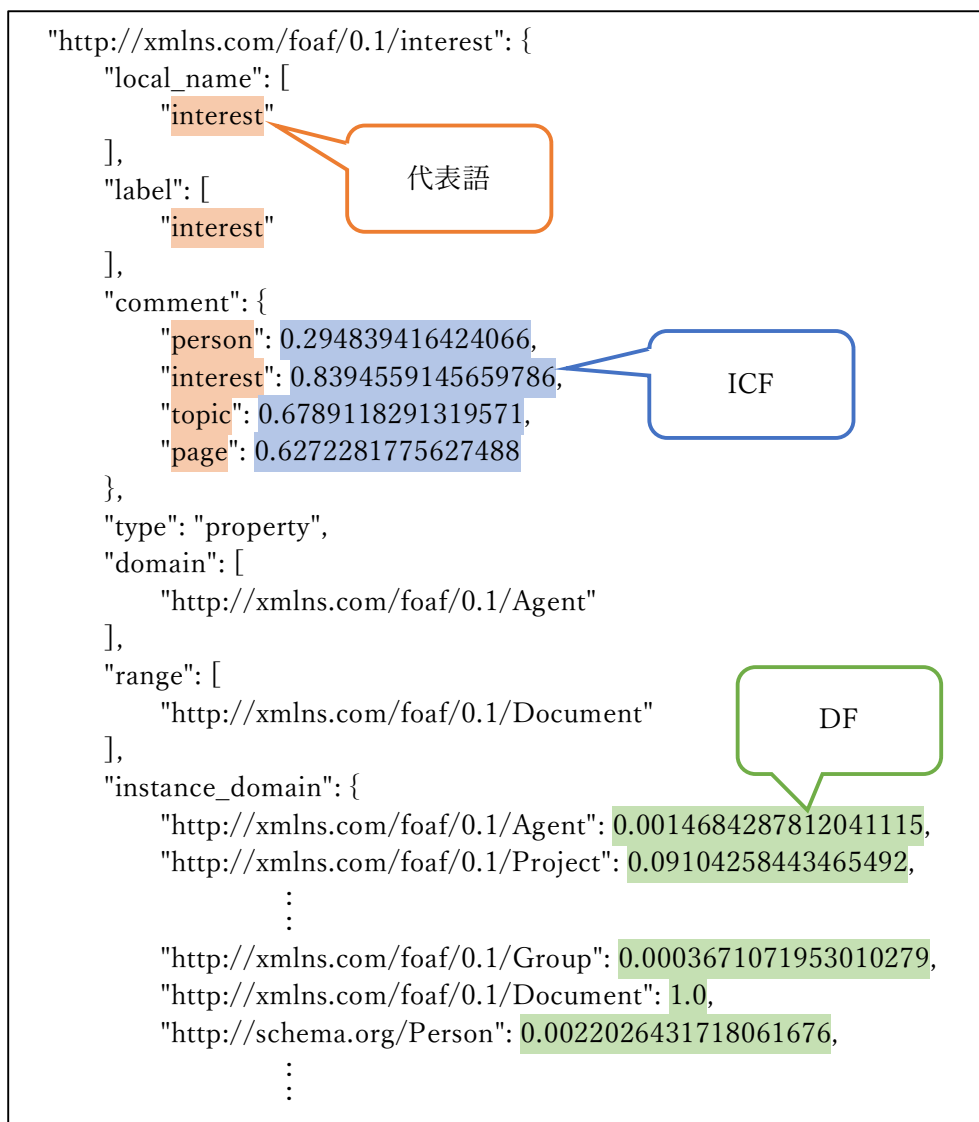


図 16 抽出した代表語と周辺情報

代表語を抽出した後は、Python のライブラリである gensim¹³が実装している Word2Vec を用いてその代表語をベクトル化する。Word2Vec に与える学習データは 2017 年 9 月 3 日時点における最新版の英 Wikipedia¹⁴の本文とし、学習時の設定は表 5 に示すようにした。ベクトル化した代表語と式 (4.1) を用いてタームをベクトル化する。図 17 は式 (4.1) における重み α_1 を 0.7, α_2 を 0.3, 式 (4.3) における重み β_1 を 0.5, β_2 を 0.3, β_3 を 0.2, ICF の閾値を 0.5, DF の閾値を 0.1 とした場合の、LOV に登録されているメタデータ語彙における foaf:interest との類似度上位 10 件のタームとその類似度である。

表 5 gensim による学習時の設定

項目	設定値	説明
size	300	ベクトル空間の次元数
window	15	文脈を判断するための学習対象の単語前後の単語数
min_count	20	学習対象となる単語の最小出現回数
sg	0	学習アルゴリズムの選択 (0 なら C-BOW, 1 なら Skip-gram)

1: http://rdf.myexperiment.org/ontologies/base/interests , 0.8209717902 2: http://sw-portal.deri.org/ontologies/swportal#hasInterests , 0.796833124029 3: http://rdfs.org/sioc/ns#topic , 0.781408630197 4: http://purl.org/dc/terms/contributor , 0.767543949821 5: http://voag.linkedmodel.org/voag#isInterestOf , 0.731232153364 6: http://voag.linkedmodel.org/voag#interestIn , 0.726134888734 7: http://dbpedia.org/ontology/mainInterest , 0.715315905702 8: http://dbpedia.org/ontology/interest , 0.714476278288 9: http://purl.org/ontology/cco/core#interest , 0.70441571807 10: http://www.aktors.org/ontology/portal#has-research-interest , 0.697832336816
--

図 17 foaf:interest との類似度上位 10 件のタームと類似度

¹³ <https://radimrehurek.com/gensim/>

¹⁴ <https://en.wikipedia.org/>

5. 評価実験

5.1. 評価手法と実験

任意のターム間に対する人間が判断する関連の強弱と類似度算出結果の比較により、提案した類似度算出手法の精度を評価する。より関連の強いターム間ほど高い類似度が得られると考える。

評価には *X-ABC* 評価セット [21] を用いる。本研究における *X-ABC* 評価セットとはあるターム *X*、人間が判断して *X* と強い関連を持つと考えられるターム *A*、*A* ほどではないが *X* と何らかの関連を持つと考えられるターム *B*、*X* と関連を持たないと考えられるターム *C* を 1 組とし、その組を複数まとめたものである。*X* と *A* の類似度を $Sim(X, A)$ 、*X* と *B* の類似度を $Sim(X, B)$ 、*X* と *C* の類似度を $Sim(X, C)$ 、評価セットに含まれるすべての組における $Sim(X, C)$ の平均を $AveSim(X, C)$ とすると、評価セットのある 1 組に対して以下の式が成り立つとき正しく類似度を算出できているとする。ただし set_{num} は評価セットに含まれる組の総数である。

$$Sim(X, A) - Sim(X, B) > AveSim(X, C) \quad (5.1)$$

$$Sim(X, B) - Sim(X, C) > AveSim(X, C) \quad (5.2)$$

$$AveSim(X, C) = \frac{\sum_{i=1}^{set_{num}} Sim(X_i, C_i)}{set_{num}} \quad (5.3)$$

式(5.1)によってより強い関連を持つターム間の類似度が高くなること、式(5.2)によって何らかの関連を持つターム間の類似度が関連を持たないターム間の類似度より高くなることを表している。評価セットに含まれるすべての組に対して評価を行い、正しく類似度を算出できている組の比率を類似度算出手法の精度とする。

評価セットはプロパティとクラスを対象にそれぞれ 3 種類ずつ人手で作成する。プロパティ、クラスともに *X* を schema.org [22] のタームとし、*A* を *X* と同等のターム、*B* を *X* の上位にあたるターム、下位にあたるターム、階層的でない関連を持つタームの 3 種類、*C* を *X* と関連を持たないタームとする。schema.org はウェブページをマークアップするためのメタデータ語彙であり、分野を問わず様々なタームを提供しているため基準となる *X* に相応しいと判断した。以下に評価セット作成の手順をプロパティ (P1~P8) とクラス (C1~C8) に分けて示す。

プロパティ評価セット 1 の作成

- P1. LOV に登録されているメタデータ語彙（以下、LOV メタデータ語彙）の定義から owl:equivalentProperty を述語とするトリプルを取得する。
- P2. P1 で取得したトリプルにおいて主語か目的語のどちらか一方のみが schema.org のプロパティの場合、そのプロパティを X とし、他方のプロパティを A とする。
- P3. P2 で取得した X と A のすべての組に対して、その X の上位にあたるプロパティを LOV メタデータ語彙から探索し B とする。また LOV メタデータ語彙からランダムにプロパティを 1 件抽出し、語彙定義を参照し X と関連を持たないことを確認して C とする。この X , A , B , C の組の集合をプロパティ評価セット 1 とする。

プロパティ評価セット 2 の作成

- P4. LOV メタデータ語彙の定義から rdfs:subPropertyOf を述語とするトリプルを取得する。
- P5. P4 で取得したトリプルにおいて主語が schema.org のプロパティの場合、そのプロパティを X とし、目的語のプロパティを B とする。ただし B も schema.org のプロパティであった場合、LOV メタデータ語彙から B と同等のプロパティが発見できればそのプロパティを B と置き換える。
- P6. P5 で取得した X と B のすべての組に対して、その X と同等であるプロパティを LOV メタデータ語彙から探索し A とする。また P3 と同様に C を加える。この X , A , B , C の組の集合をプロパティ評価セット 2 とする。

プロパティ評価セット 3 の作成

- P7. 作成したプロパティ評価セット 1 と 2 から重複を除き X と A の組を取得する。
- P8. P7 で取得した X と A のすべての組に対して、その X と記述対象が同等であり、共に使用されるような階層的でない関連を持つプロパティを LOV メタデータ語彙から探索し B とする。また P3 と同様に C を加える。この X , A , B , C の組の集合をプロパティ評価セット 3 とする。

クラス評価セット 1 の作成

- C1. LOV メタデータ語彙の定義から owl:equivalentClass を述語とするトリプルを取得する。
- C2. C1 で取得したトリプルにおいて主語か目的語のどちらか一方のみが schema.org のクラスの場合、そのクラスを X とし、他方のクラスを A とする。
- C3. C2 で取得した X と A のすべての組に対して、その X の上位にあたるクラスを LOV メタデータ語彙から探索し B とする。また LOV メタデータ語彙からランダムにクラスを 1 件抽出し、語彙定義を参照し X と関連を持たないことを確認して C とする。この X , A , B , C の組の集合をクラス評価セット 1 とする。

クラス評価セット 2 の作成

- C4. LOV メタデータ語彙の定義から rdfs:subClassOf を述語とするトリプルを取得する。
- C5. C4 で取得したトリプルにおいて主語が schema.org のクラスであり目的語が schema.org のクラスでない場合、主語のクラスを X 、目的語のクラスを B とする。
- C6. C5 で取得した X と B のすべての組に対して、その X と同等であるクラスを LOV メタデータ語彙から探索し A とする。また C3 と同様に C を加える。この X , A , B , C の組の集合をクラス評価セット 2 とする。

クラス評価セット 3 の作成

- C7. 作成したクラス評価セット 1 と 2 から重複を除き X と A の組を取得する。
- C8. C7 で取得した X と A のすべての組に対して、その X と共通の上位クラスを持つような階層的でない関連を持つクラスを LOV メタデータ語彙から探索し B とする。また C3 と同様に C を加える。この X , A , B , C の組の集合をクラス評価セット 3 とする。

P3, C3 における上位にあたるタームの探索は可能な限り階層的に近いタームを選定した。表 6 は各評価セットの X , A , B , C の例である。作成した評価セットは作成者の他 1 名により確認を行い、不適切なタームの組を修正した。

表 6 各評価セットの X , A , B , C の例

プロパティ評価セット 1 ($set_{num} = 47$)	
X :	http://schema.org/actor
A :	http://dbpedia.org/ontology/starring
B :	http://www.w3.org/ns/ma-ont#hasContributor
C :	http://open.vocab.org/terms/recordedAddress
プロパティ評価セット 2 ($set_{num} = 79$)	
X :	http://schema.org/productID
A :	http://www.w3.org/2006/vcard/ns#prodid
B :	http://purl.org/goodrelations/v1#hasEAN_UCC-13
C :	http://purl.org/ontology/po/actor
プロパティ評価セット 3 ($set_{num} = 60$)	
X :	http://schema.org/author
A :	http://dbpedia.org/ontology/author
B :	http://d-nb.info/standards/elementset/gnd#editor
C :	http://rdf.myexperiment.org/ontologies/base/organisation
クラス評価セット 1 ($set_{num} = 69$)	
X :	http://schema.org/AdministrativeArea
A :	http://rdfs.co/juso/PoliticalDivision
B :	http://dbpedia.org/ontology/Place
C :	http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#Image
クラス評価セット 2 ($set_{num} = 70$)	
X :	http://schema.org/CreativeWork
A :	http://dbpedia.org/ontology/Work
B :	http://purl.org/library/Game
C :	http://dbpedia.org/ontology/AutoRacingLeague
クラス評価セット 3 ($set_{num} = 61$)	
X :	http://schema.org/ImageObject
A :	http://xmlns.com/foaf/0.1/Image
B :	http://dati.beniculturali.it/cis/VideoObject
C :	http://purl.org/linguistics/gold/Mid

作成した評価セットを用いて提案した手法の評価実験を行う。その際、タームのベクトル化、類似度算出における各種重みと ICF の閾値を変えて精度を比較する。プロパティの類似度算出における各種重みと ICF の閾値の組み合わせを表 7 に、クラスの類似度算出における各種重みと ICF の閾値の組み合わせを表 8 に示す。またプロパティ、クラスの各パターンの説明を表 9 に示す。ただし、DF の閾値は 0.1 に固定している。

表 7 プロパティの類似度算出における各種重みと ICF の閾値の組み合わせ

パターン	α_1 (ラベル)	α_2 (コメント)	ICF	β_1 (基礎記述)	β_2 (ドメイン)	β_3 (レンジ)
1	0.7	0.3	0.0	1.0	0.0	0.0
2	0.7	0.3	0.5	1.0	0.0	0.0
3	0.7	0.3	1.0	1.0	0.0	0.0
4	0.3	0.7	0.5	1.0	0.0	0.0
5	0.7	0.3	0.0	0.5	0.3	0.2
6	0.7	0.3	0.5	0.5	0.3	0.2
7	0.7	0.3	1.0	0.5	0.3	0.2
8	0.3	0.7	0.5	0.5	0.3	0.2

表 8 クラスの類似度算出における各種重みと ICF の閾値の組み合わせ

パターン	α_1 (ラベル)	α_2 (コメント)	ICF	γ_1 (基礎記述)	γ_2 (クラス共起プロパティ)
1	0.7	0.3	0.0	1.0	0.0
2	0.7	0.3	0.5	1.0	0.0
3	0.7	0.3	1.0	1.0	0.0
4	0.3	0.7	0.5	1.0	0.0
5	0.7	0.3	0.0	0.7	0.3
6	0.7	0.3	0.5	0.7	0.3
7	0.7	0.3	1.0	0.7	0.3
8	0.3	0.7	0.5	0.7	0.3

表 9 各パターンの説明

パターン	対象	説明
1	プロパティ	コメントよりラベルを重視，コメントのすべての代表語を利用，基礎記述のみを用いて類似度算出
2	プロパティ	コメントよりラベルを重視，コメントの一部の代表語を利用，基礎記述のみを用いて類似度算出
3	プロパティ	コメントよりラベルを重視，コメントの最小限の代表語を利用，基礎記述のみを用いて類似度算出
4	プロパティ	ラベルよりコメントを重視，コメントの一部の代表語を利用，基礎記述のみを用いて類似度算出
5	プロパティ	コメントよりラベルを重視，コメントのすべての代表語を利用，基礎記述・ドメイン・レンジを用いて類似度算出
6	プロパティ	コメントよりラベルを重視，コメントの一部の代表語を利用，基礎記述・ドメイン・レンジを用いて類似度算出
7	プロパティ	コメントよりラベルを重視，コメントの最小限の代表語を利用，基礎記述・ドメイン・レンジを用いて類似度算出
8	プロパティ	ラベルよりコメントを重視，コメントの一部の代表語を利用，基礎記述・ドメイン・レンジを用いて類似度算出
1	クラス	コメントよりラベルを重視，コメントのすべての代表語を利用，基礎記述のみを用いて類似度算出
2	クラス	コメントよりラベルを重視，コメントの一部の代表語を利用，基礎記述のみを用いて類似度算出
3	クラス	コメントよりラベルを重視，コメントの最小限の代表語を利用，基礎記述のみを用いて類似度算出
4	クラス	ラベルよりコメントを重視，コメントの一部の代表語を利用，基礎記述のみを用いて類似度算出
5	クラス	コメントよりラベルを重視，コメントのすべての代表語を利用，基礎記述・クラス共起プロパティを用いて類似度算出
6	クラス	コメントよりラベルを重視，コメントの一部の代表語を利用，基礎記述・クラス共起プロパティを用いて類似度算出
7	クラス	コメントよりラベルを重視，コメントの最小限の代表語を利用，基礎記述・クラス共起プロパティを用いて類似度算出
8	クラス	ラベルよりコメントを重視，コメントの一部の代表語を利用，基礎記述・クラス共起プロパティを用いて類似度算出

5.2. 実験結果

プロパティ評価セット 1～3, クラス評価セット 1～3 を用いた実験結果をそれぞれ図 18～23 に示す. 横軸はパターンの番号, 縦軸は類似度算出手法の精度である.

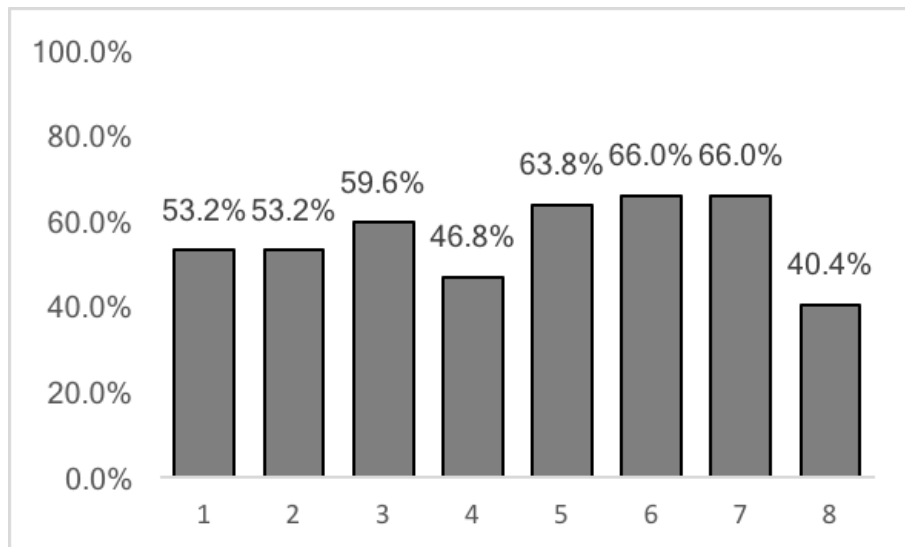


図 18 プロパティ評価セット 1 を用いた実験の結果

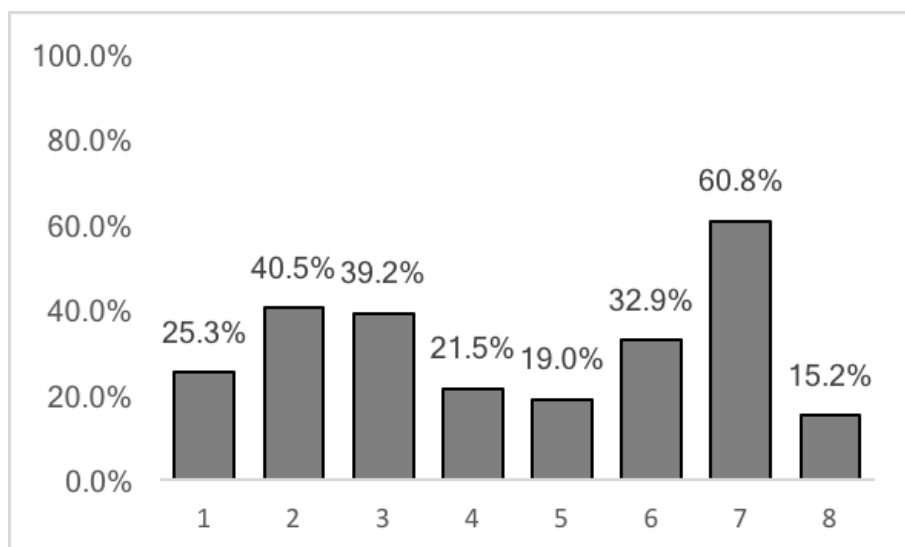


図 19 プロパティ評価セット 2 を用いた実験の結果

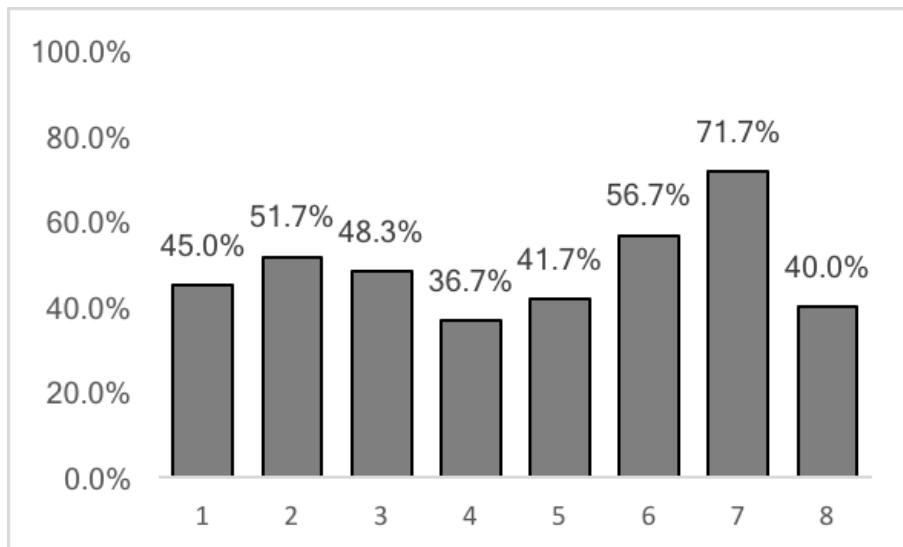


図 20 プロパティ評価セット 3 を用いた実験の結果

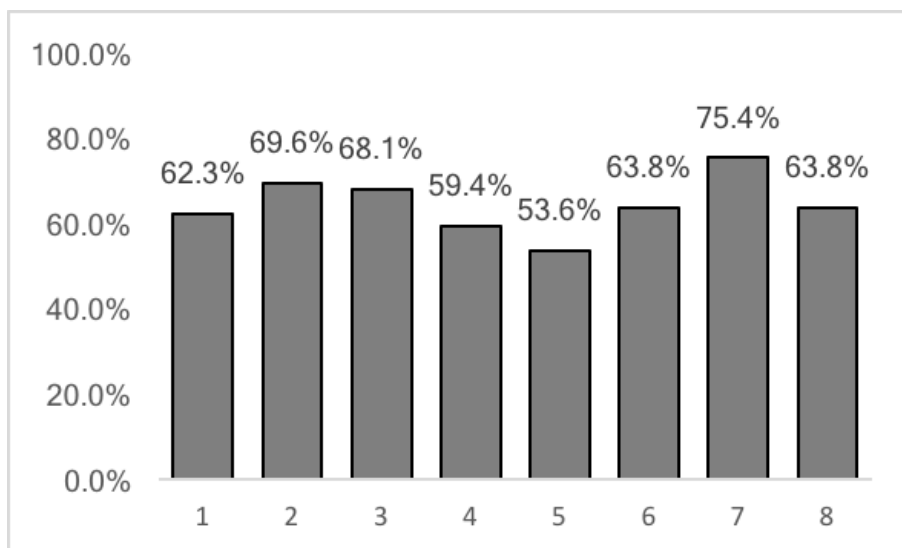


図 21 クラス評価セット 1 を用いた実験の結果

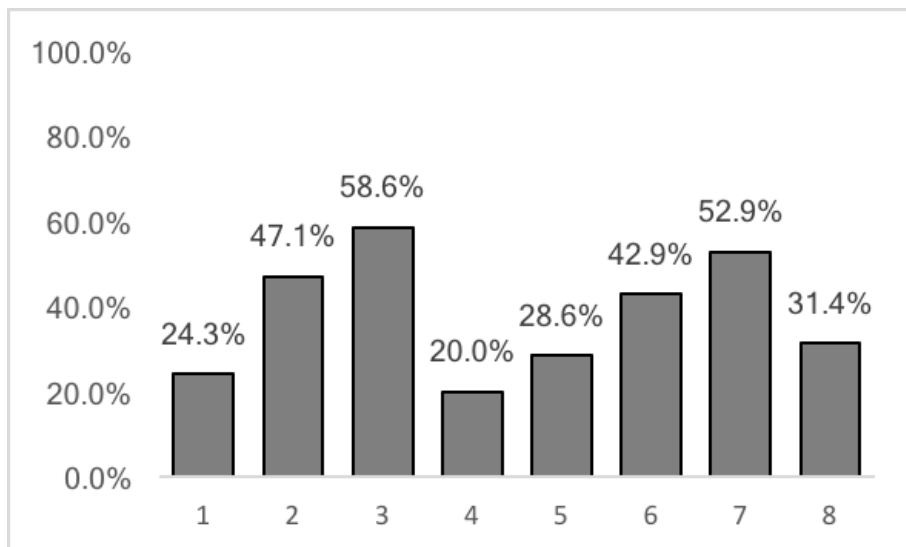


図 22 クラス評価セット 2 を用いた実験の結果

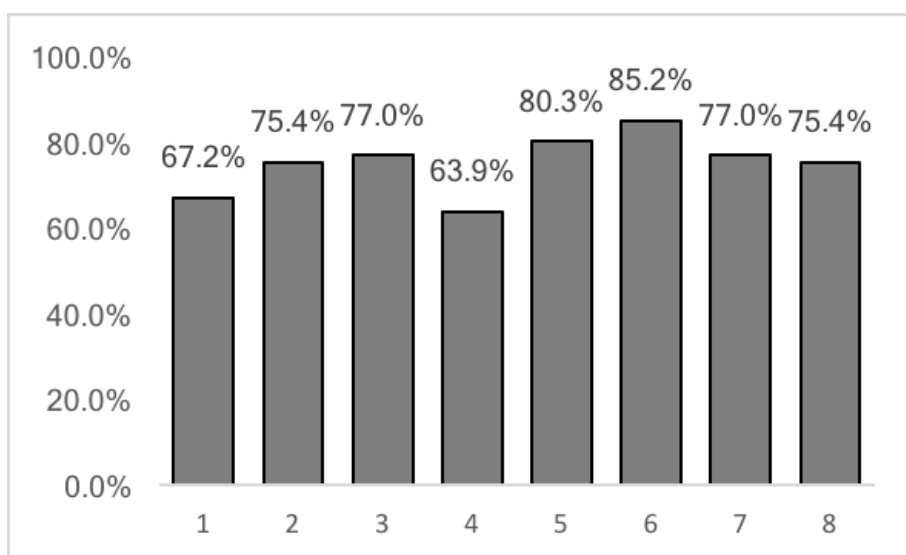


図 23 クラス評価セット 3 を用いた実験の結果

6. 考察

5 章で行った評価実験の結果について考察する。結果から、ICF の閾値のみを変化させているパターン 1~3 とパターン 5~7 において、類似度算出の対象、評価セットの種類を問わずパターン 1, 5 の精度が他のパターンと比べ低い傾向にあることがわかる。本手法では抽出した各代表語のベクトルの総和をタームのベクトルとしているため、利用する代表語の数が多い場合に類似度が高くなる傾向がある。パターン 1, 5 は ICF の閾値が 0.0, つまりコメントの代表語をすべて利用してタームをベクトル化している。したがって弱い関連を持つターム間や関連を持たないターム間の類似度が高くなってしまい、類似度算出の精度が落ちている。一方で ICF の閾値を 0.5, 1.0 として利用するコメントの代表語を減らしているパターン 2, 3, 6, 7 の精度は高くなっている。このことから、提案手法では多数の代表語を利用するよりもそのタームを特徴付ける少数の代表語を利用することが重要であると考えられる。

パターン 4, 8 はラベルから抽出した代表語よりもコメントから抽出した代表語を重視してタームをベクトル化しているが、コメントよりもラベルの代表語を重視して他の条件が同じであるパターン 2, 6 と比べて精度が低い傾向にある。これはラベルがタームを識別するための名前であり、コメントと比べて抽出される代表語がタームを特徴付けるものであることが多いためである。このことから、提案手法ではコメントよりもラベルから抽出した代表語を重視すべきだと考えられる。

パターン 1~4 はタームの類似度算出に基礎記述のみを利用し、パターン 5~8 は基礎記述に加えてプロパティの場合はドメインとレンジ、クラスの場合はクラス共起プロパティを利用している。

プロパティの類似度算出ではドメインとレンジを利用することで、とくに評価セット 2, 3 において精度の向上が見られる。これは基礎記述からはわからない関連をドメイン、レンジから得ることができるためである。表 10 は各プロパティ評価セットに含まれるすべての組における $Sim(X, A)$, $Sim(X, B)$, $Sim(X, C)$ の平均 (それぞれ $AveSim(X, A)$, $AveSim(X, B)$, $AveSim(X, C)$) のパターン 3 と 7 での比較である。すべての評価セットにおいて類似度算出にドメイン、レンジを利用するパターン 7 の $AveSim(X, B)$ が基礎記述のみを利用するパターン 3 と比べて大きく上昇している。このことから、提案手法ではドメインとレンジの利用が、同等ではないが何らかの関連を持つプロパティ間の類似度算出に有用であることがわかる。

表 10 各プロパティ評価セットにおける類似度の平均の比較

プロパティ評価セット 1

パターン	$AveSim(X, A)$	$AveSim(X, B)$	$AveSim(X, C)$
3	0.75420726857	0.257578439496	0.0664950485872
7	0.737921219161	0.323556858888	0.112849509389

プロパティ評価セット 2

パターン	$AveSim(X, A)$	$AveSim(X, B)$	$AveSim(X, C)$
3	0.794727577164	0.325290088902	0.0769794853269
7	0.682684204818	0.422324978553	0.122495788721

プロパティ評価セット 3

パターン	$AveSim(X, A)$	$AveSim(X, B)$	$AveSim(X, C)$
3	0.772903468442	0.270708223731	0.0716629872851
7	0.726133102381	0.383553959179	0.11057128146

クラスの類似度算出ではクラス共起プロパティを利用することで精度が向上する場合と低下する場合があるが、いずれも大きな変化ではない。これはクラス共起プロパティとして採用したすべてのプロパティから抽出した代表語を利用して類似度を計算していることから、類似度が平均的に高くなり差がつかなかったためだと考えられる。

プロパティ、クラスともに評価セット 2 は B を X の下位にあたるタームとしている。結果から評価セット 2 を用いた実験における類似度算出の精度が他の評価セットと比べ低いことがわかる。これは評価セット作成において評価セット 1, 3 は B を作成者が一定の基準で選定しているのに対して、評価セット 2 は下位にあたるタームの探索が難しいという理由から既存の関係記述を流用しており、組によって X と B の階層的な距離に差があったためだと考えられる。

今後の課題として代表語の抽出と選出方法ならびに Word2Vec に与える学習データの改良が挙げられる。

実験結果の考察においてよりタームを特徴付ける少数の代表語を利用することが重要だと述べた。提案手法ではコメントから抽出した代表語は独自に定義した ICF によって重み付けを行い利用する代表語の選出を行っている。コメントから抽出した代表語のその他の選出方法として、コメント内での抽出源の利用がある。タームによってはコメントが複数の文で記述されているものがあり、その場合は最初の文から抽出した代表語がよりタームを

特徴付けると考えられる。またコメントから抽出した代表語をまったく用いず、ラベルから抽出した代表語のみを利用する類似度算出が考えられる。図 24 は各評価セットを用いた実験結果においてラベルとコメントから抽出した代表語を利用して最も高い精度となったパターンと、ラベルから抽出した代表語のみを利用し、基礎記述に加えプロパティの場合はドメインとレンジ、クラスの場合はクラス共起プロパティを用いて類似度算出を行った場合の精度の比較である。プロパティの類似度算出においては精度が向上し、クラスの類似度算出においては精度の低下が見られるもののわずかな差となっていることがわかる。さらに今回類似度算出の対象としたプロパティ 33,692 件、クラス 24,872 件のうちコメントが取得できたのはそれぞれ 17,290 件、13,001 件であり、タームの定義として記述が推奨されているコメントを持たないタームも多い。このことと図 24 が示す結果から、ラベルのみから代表語を抽出する方法が有用である可能性があるとわかる。

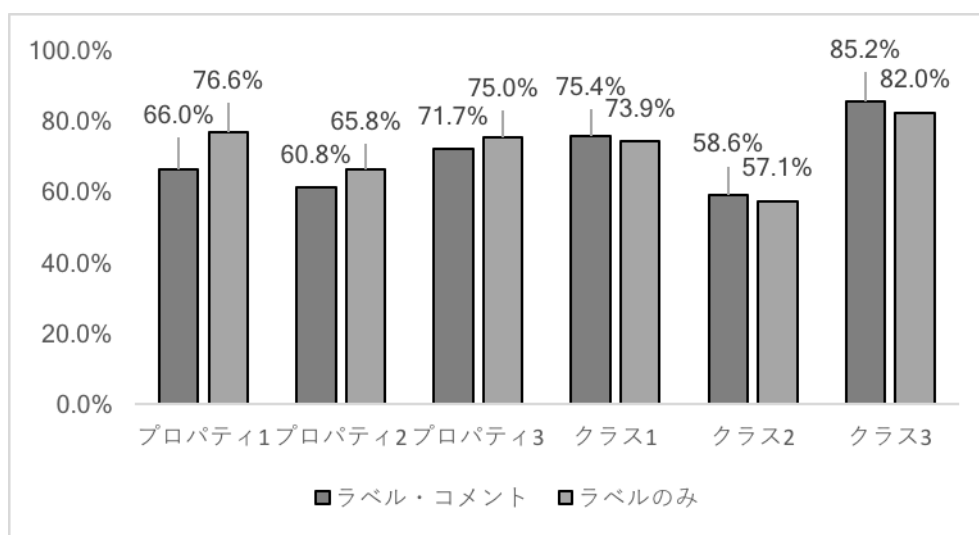


図 24 ラベルのみを利用した類似度算出の実験結果

提案手法の実現において、LOV メタデータ語彙に含まれるタームから代表語として抽出した 20,490 語の単語のうち 5,780 語は Word2Vec によってベクトル化できなかった。ベクトル化できなかった単語を分類し、例とともに表 11 に示す。通し番号や分類記号などに関しては、その代表語の抽出元のタームを含むメタデータ語彙の名称や作成者からその通し番号や分類記号を持つ分類体系などを推定することで、自然言語に置き換えられる可能性がある。記号と複合した単語、複合語分割が不十分な単語、スペルミスを含んだ単語に関しては、その単語を作成したモデルに問い合わせ、学習済みの単語として登録されていなかった

表 11 ベクトル化できなかった単語の分類

分類	例
通し番号や分類記号など	「aat2882」「#.b.7.1」
記号と複合した単語	「.width」「#ask」
複合語分割が不十分な単語	「byday」「prodname」
スペルミスを含んだ単語	「affiliation」「requeriment」
英語以外の言語	「国名コード」「ч а с т и」
学習データに含まれない， あるいは出現回数が極端に少ない単語	「followee」「well-known」

た場合には記号の削除，n-gram で分割して意味のある単語の抽出，レーベンシュタイン距離やジャロ・ウィンクラー距離などの指標をもとにした修正などを試みる．対象としている言語以外の単語に関しては対象言語に翻訳するといった対策が考えられる．学習データに含まれない，あるいは出現回数が極端に少ない単語に関しては，代表語をもとに学習データを用意することで解決できる．今回は英 Wikipedia の本文を学習データとしているが，代表語として抽出した単語を含む文書を収集し学習データとすることで，ベクトル化されない単語を減らすことができる．

7. 関連研究

タームの関連を扱う研究分野としてオントロジーアライメント[23]がある。オントロジーとは特定のドメインの知識を概念と概念間の関係の記述により表現したものであり、概念をタームと捉えるとメタデータ語彙もオントロジーの一種である[24]。オントロジーアライメントとは異なるオントロジー間の概念の関係を発見することであり、アライメントを実現する手法には概念間の類似度算出を伴うものが多い。メタデータ語彙を対象としたアライメント手法に PropString[25]がある。PropString はプロパティのアライメント手法であり、プロパティの語彙定義から単語を抽出し、Soft TF-IDF を用いて類似度を算出している。Soft TF-IDF とは単語が完全に一致していないと類似度が高くないという TF-IDF の問題を、単語の一致判定にジャロ・ウィンクラー距離を利用することで改善する手法である。しかしジャロ・ウィンクラー距離はスペルミスや語形の変化には対応できるが、文字列が大きく異なる意味的に近い単語には対応できない。本研究は分散表現を用いることで単語の意味に基づきタームの類似度を算出している。

久永らはスプレッドシートや csv 形式で公開されている日本のオープンデータの RDF 化を目的として、Word2Vec を用いたプロパティの推薦手法を提案している[26]。オープンデータの記述に使用されている項目名を RDF のトリプルにおける述語と捉え、項目名とメタデータ語彙のプロパティの類似度を算出し、類似度の高いプロパティを推薦する。項目名およびプロパティのラベルから抽出した単語を Word2Vec によりベクトル化し、ベクトル化した単語の総和のコサイン類似度を計算することで類似度を算出している。本研究はプロパティの類似度算出にラベルに加えてコメントやドメイン、レンジを用いて精度を向上させている。

8. おわりに

本論文では、既存のメタデータ語彙から利用目的に応じたタームを探索する際に関連するタームの参照が有用だと考え、あるタームから関連タームを参照できるようタームの類似度算出手法を提案、実現した。

本手法ではタームの意味に基づき類似度算出が行えるよう分散表現を利用した。タームの URI および語彙定義における名前と説明文からタームを特徴付ける単語である代表語を抽出し、分散表現を生成するツールである Word2Vec を用いてベクトル化した。ベクトル化した代表語の演算によりタームのベクトルを求め、ベクトルのコサイン類似度を計算することでタームの類似度を算出した。また対象となっているターム自身だけでなく、プロパティの場合はドメインとレンジ、クラスの場合はクラス共起プロパティの類似度も求め、最終的な類似度算出に利用した。

提案手法を実現し、独自に作成した *X-ABC* 評価セットを用いて類似度算出における重みや閾値を変えて評価実験を行い、結果を比較したところ、多数の代表語を利用するよりもより重要な少数の代表語を利用することで類似度算出の精度が向上すること、ドメインとレンジを利用することでプロパティの類似度算出の精度が向上することがわかった。さらに、タームの説明文であるコメントから抽出した代表語を用いず類似度を算出した場合、プロパティの類似度算出では精度が向上すること、クラスの類似度算出では精度が低下するがわずかな差であること、コメントの記述がないタームが多く存在することから、タームの名前であるラベルから抽出した代表語のみを利用した類似度算出が有用である可能性があるとわかった。また、代表語として抽出してもベクトル化できない単語が多く存在することがわかり、代表語の抽出と選出方法ならびに Word2Vec に与える学習データの改良の必要性に関する知見を得た。

謝辞

これまでの研究を進めるにあたり、テーマの模索からゼミでの議論の方法、論文のまとめ方まで様々な場面でご指導いただいた永森光晴先生と杉本重雄先生に深謝いたします。また杉本・永森研究室の皆様にも多くのご助言、励ましをいただきました。ここに心より感謝の意を表します。

参考文献

- [1] Open Knowledge International. “What is Open Data?”. OPEN DATA HANDBOOK.
<http://opendatahandbook.org/guide/en/what-is-open-data/>, (accessed 2018-01-10)
- [2] Linked Data. <http://linkeddata.org/>, (accessed 2018-01-10)
- [3] Tim Berners-Lee. Linked Data - Design Issues.
<https://www.w3.org/DesignIssues/LinkedData.html>, (accessed 2018-01-10)
- [4] The Linking Open Data cloud diagram. <http://lod-cloud.net/>, (accessed 2018-01-10)
- [5] “RDF”. Semantic Web Standards. <https://www.w3.org/RDF/>, (accessed 2018-01-10)
- [6] Dublin Core Metadata Initiative. “Dublin Core Metadata Element Set, Version 1.1: Reference Description”. Dublin Core Metadata Initiative.
<http://dublincore.org/documents/dces/>, (accessed 2018-01-10)
- [7] Dan Brickley, Libby Miller. FOAF Vocabulary Specification 0.99.
<http://xmlns.com/foaf/spec/>, (accessed 2018-01-10)
- [8] Dan Brickley. Basic Geo (WGS84 lat/long) Vocabulary.
<https://www.w3.org/2003/01/geo/>, (accessed 2018-01-10)
- [9] RDF Schema 1.1. <https://www.w3.org/TR/rdf-schema/>, (accessed 2018-01-10)
- [10] Mikael Nilsson. “Description Set Profiles: A constraint language for Dublin Core Application Profiles”. Dublin Core Metadata Initiative.
<http://dublincore.org/documents/dc-dsp/>, (accessed 2018-01-11)
- [11] メタデータ情報基盤構築事業. “メタデータ情報共有のためのガイドライン”.
<http://www.mi3.or.jp/item/A03.pdf>, (accessed 2018-01-11)
- [12] Karen Coyle, Thomas Baker. “Guidelines for Dublin Core Application Profiles”. Dublin Core Metadata Initiative. <http://dublincore.org/documents/profile-guidelines/>, (accessed 2018-01-11)
- [13] Max Schmachtenberg, Christian Bizer, Heiko Paulheim. “State of the LOD Cloud 2014”. The Linking Open Data cloud diagram. http://lod-cloud.net/state/state_2014/, (accessed 2018-01-11)
- [14] LODStats. <http://stats.lod2.eu/>, (accessed 2018-01-11)
- [15] vocab.cc. <http://vocab.cc/>, (accessed 2018-01-11)
- [16] Linked Open Vocabularies. <http://lov.okfn.org/dataset/lov>, (accessed 2018-01-11)
- [17] 落合香織. (2014). “アジャイル開発モデルに基づくメタデータスキーマ設計手法と支

- 援システム”. 筑波大学, 修士論文.
- [18] 西出頼継. (2015). “メタデータスキーマ作成のためのメタデータ語彙探索支援システムの構築”. 筑波大学, 修士論文.
- [19] Tomas Mikolov, kai Chen, Greg Corrado, Jeffrey Dean. (2013). “Efficient Estimation of Word Representations in Vector Space”. Proceedings of the International Conference on Learning Representations 2013.
- [20] SPARQL 1.1 Query Language. <https://www.w3.org/TR/sparql11-query/>, (accessed 2018-01-11)
- [21] 奥村紀之, 土屋誠司, 渡部広一, 河岡司. (2007). “概念間の関連度計算のための大規模概念ベースの構築”. 自然言語処理, Vol. 14, No. 5, pp. 41-64.
- [22] schema.org. <http://schema.org/>, (accessed 2018-01-11)
- [23] Jérôme Euzenat, Pavel Shvaiko. (2007). “Ontology Matching”. Springer, p. 42.
- [24] Tom Heath, Christian Bizer. (2011). “Linked Data: Evolving the Web into a Global Data Space”. Morgan & Claypool, p. 57.
- [25] Michelle Cheatham, Pascal Hitzler. (2014). “The Properties of Property Alignment”. Proceedings of the 9th International Workshop on Ontology Matching collocated with 13th International Semantic Web Conference, pp. 13-24.
- [26] 久永忠範, 湊田孝康, 能登大輔, 郭崇, 陳博. (2017). “オープンデータにおける RDF 変換の研究”. 情報知識学会誌, Vol. 27, No. 2, pp. 207-212.