

メタデータのグラフ構造を利用した
メタデータタームの類似度算出手法

筑波大学

図書館情報メディア研究科

2019年3月

金城良大

目次

1	はじめに	4
2	セマンティックウェブとメタデータ	5
2.1	データのウェブの実現	5
2.2	Web上のメタデータとRDF	6
2.3	メタデータスキーマとインスタンス	7
2.3.1	語彙定義	8
2.3.2	記述規則	9
2.4	Linked Open Data	12
3	メタデータタームの探索と用法理解	14
3.1	LOD作成時のメタデータターム探索における問題	14
3.2	LOD利用時のメタデータタームの用法理解における問題	15
3.3	メタデータターム間類似度による問題の解決支援	16
4	グラフ構造を利用した類似度算出手法の提案	18
4.1	提案手法概要	18
4.1.1	エンベディング手法について	19
4.2	重み付きグラフの作成	21
4.3	事前のマッピングとグラフの結合	22
4.4	類似度の算出方法	23
5	関連研究	24
5.1	マッチング手法	24
5.1.1	文字列を基にしたマッチング手法	24
5.1.2	グラフ構造を利用したマッチング手法	25
5.1.3	ドメイン知識を活かしたマッチング手法	25
5.2	先行研究	25
6	評価実験	26

6.1	評価方法.....	26
6.2	実験環境.....	27
6.3	結果.....	27
7	考察.....	30
8	おわりに.....	36
	謝辞.....	37
	参考文献.....	38
	付録.....	41
	1. <i>DarkScape</i>	41
	2. <i>Old School RuneScape</i>	44
	3. <i>RuneScape</i>	46
	4. <i>Memory Alpha</i>	48
	5. <i>Memory Beta</i>	50
	6. <i>Star Trek</i>	52

図目次

図 1：文書のウェブ [5].....	5
図 2：データのウェブ [5].....	5
図 3：RDF トリプルの例.....	6
図 4：文字列を含むトリプルの例.....	6
図 5：クラスを含む RDF グラフの例.....	7
図 6：スキーマとインスタンスの関係図.....	8
図 7：XML 形式の DSP の例 [15].....	10
図 8：簡易 DSP の記述例.....	11
図 9：LOD cloud グラフ [1].....	13
図 10：人の名前を問い合わせる SPRAQL クエリ例.....	16
図 11：トリプル例 1.....	17
図 12：トリプル例 2.....	17
図 13：提案手法の概要図.....	19
図 14：ネットワークグラフの例([23]より).....	20
図 15：重み付きグラフ作成のアルゴリズム.....	22
図 16：URI の最後の単語文字列の完全一致によるマッチング.....	23
図 17：LOD データセットのヒストグラム.....	34
図 18：LOD データセットの累積ヒストグラム.....	35

表目次

表 1：FOAF の語彙定義の一部.....	9
表 2：DCMES の語彙定義の一部.....	9
表 3：DPLA の記述規則の一部.....	12
表 4：knowledge graph トラックのデータセット.....	27
表 5：提案手法による上位 10 以内の再現率.....	28
表 6：重み付けを行わない提案手法の上位 10 以内の再現率.....	28
表 7：正規化を行わない提案手法の上位 10 以内の再現率.....	29
表 8：前処理を行わなかった場合の上位 10 以内の再現率.....	29
表 9：LOD 組と合わせた共有率.....	33
表 10：提案手法による上位 n 以内の再現率.....	33
表 11：正規化を行わない提案手法の次元数ごとの再現率.....	34

1 はじめに

新しいデータ公開の仕組みとして Linked Open Data(LOD) が注目を集めている。LOD とはオープンライセンスで公開されており、標準化されたデータ表現とアクセス方式を持つ、相互運用性の向上を目指したデータである¹。2006 年頃から LOD としてデータを公開する取り組みは始まっており、年々その数は増加している [1]。また 2013 年の G8 サミットでは LOD として政府データを公開することを推奨する”オープンデータ憲章”が合意されるなど、LOD はデータ公開の標準として期待をされている [2]。

しかし LOD は研究者の間で普及が進む一方、専門知識を持たない人には扱いが難しい。本研究では LOD の扱いの難しさのうち、データ記述に使用するメタデータタームの探索と用法理解の難しさに注目した。メタデータタームとは記述する事物のデータ項目であるプロパティと記述する事物の属性を表すクラスの総称である。メタデータタームは利用方法や意味解釈のためのラベルなどの情報が形式的に定義され、その定義に従って利用されることが望まれている。しかし LOD ではメタデータタームが適切に定義されていない、あるいは定義を参照解決できないケースが多く見られる。そのため LOD を利用する際のメタデータタームの用法理解が難しくなる。また LOD を作成する際はメタデータタームを独自に定義するのではなく、既存のメタデータタームを再利用することが推奨されており、目的に即したメタデータタームを探索する必要がある。その際にも適切な定義がないことが、探索を難しくする要因の 1 つとなっている。

こうした問題に対処するために弊研究室では LOD のインスタンスから語彙定義²を推定する研究 [3] やメタデータターム間の類似度算出手法に関する研究 [4] を行ってきた。文献 [3] は LOD インスタンスすなわち実データから語彙定義を推定することで、メタデータタームの理解を支援しようという研究である。また文献 [4] は語彙定義で使われる単語列の類似度からメタデータターム間の類似度を算出することでターム間の類似関係を明らかにし、メタデータタームの探索を支援しようという研究である。

本研究では文献 [4] と同様にメタデータターム間の類似度算出手法を提案し、メタデータタームの用法理解と探索に役立てることを目的とする。語彙定義がない状況でも適用できるように LOD インスタンスのグラフ構造からその類似度の算出を行う。2 章ではメタデータと LOD の基本技術について説明する。3 章ではメタデータタームの問題点について述べ、4 章で提案手法を紹介する。5 章で関連研究を紹介し、6 章で実験を行う。7 章で実験結果の考察と手法全体の考察を行い、最後に 8 章で本研究のまとめを述べる。

¹文脈によってはデータ公開に関するセマンティックウェブの取り組みや周辺技術を総称する場合もある

²本研究ではメタデータタームの定義を語彙定義と呼ぶ

2 セマンティックウェブとメタデータ

本章では研究分野であるセマンティックウェブと Linked Open Data について説明する。

2.1 データのウェブの実現

バーナーズ=リーは人間が読むためのページがリンクする情報空間を「文書のウェブ」、そしてコンピュータ処理可能なデータが有機的に連動する空間を「データのウェブ」と呼んだ [5] [6]。図 1 に文書のウェブのイメージ図を、図 2 にデータのウェブのイメージ図をそれぞれ示す。2000 年代前半までのウェブはハイパーリンクが結ばれているだけの文書のウェブであり、セマンティックウェブとは「文書のウェブからデータのウェブへ」という考えである [7]。データのウェブを実現するための技術や活動の総称をセマンティックウェブと呼ぶ。セマンティックウェブを実現するためにはウェブ上の情報資源に対してメタデータを付与することが必要だとされている。メタデータとは「データについてのデータ」のことである。特にウェブにおいては機械可読なものを指す。潤沢なメタデータを付与することでコンピュータによるデータの解釈を容易にすることができる。例えば HTML 文書では head 要素で title タグを与えることで検索エンジンがその主題を明確に把握することができる。また meta タグ要素を用いることで、検索エンジンがそのページを検索結果に表示させないように制御することができる。このようにメタデータを付与することによって人間による解釈を挟まずに、コンピュータがデータを扱えるようになる。そして現在行われているデータのウェブ実現のための取り組みの 1 つに LOD プロジェクトがある(2.4 後述)。本研究ではこの LOD データを対象とした研究を行う。



図 1：文書のウェブ [5]

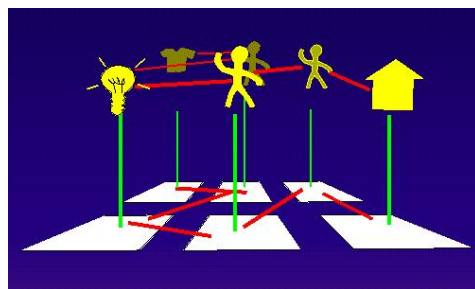


図 2：データのウェブ [5]

2.2 Web 上のメタデータと RDF

Web 上にはたくさんの情報資源が存在する。その情報資源の中でも URI を持つものをリソースと呼ぶ。セマンティックウェブの領域では、それらリソースのメタデータは RDF (Resource Description Framework) に従い記述される。RDF では主語 - 述語 - 目的語の三要素の組み合わせでメタデータを記述する。この三要素で構成されるリソース記述をトリプルと呼ぶ。また「述語」を「プロパティ」と表現することもあり、本研究では以降「述語」ではなく「プロパティ」を使う。図 3 に RDF トリプルの例を示す。このトリプルは主語リソースが「太宰治」、プロパティが「write」、目的語が「人間失格」である。このトリプルの意味としては「太宰治が人間失格を書く(write)こと」を表している。RDF ではリソースとリテラル(URL ではない単なる文字列)を明確に分ける。図 4 に文字列を含むトリプルの例を示す。この例では「太宰治の誕生日(birth date)が 1909/06/19」であることを表している。図 4 のようにトリプルを描画する際にはリソースは楕円で、リテラルは四角で囲み表現する。リテラルは目的語としてのみ扱われトリプルの主語になることはない。また RDF ではリソースに対して型属性であるクラスを与えることができる。図 5 にクラス表現を含む RDF グラフの例を示す。図 5 では「rdf:type³」プロパティを用いて「太宰治が人間(Person)クラスに属している」ことを示している。RDF で使用されるクラスとプロパティを総称してメタデータタームと呼ぶ。また図 5 のように RDF トリプルは連結することが可能であり、RDF で記述されたメタデータはグラフ構造を持つ。RDF を用いることで複雑な記述対象であっても柔軟にメタデータの記述が可能になる。



図 3 : RDF トリプルの例

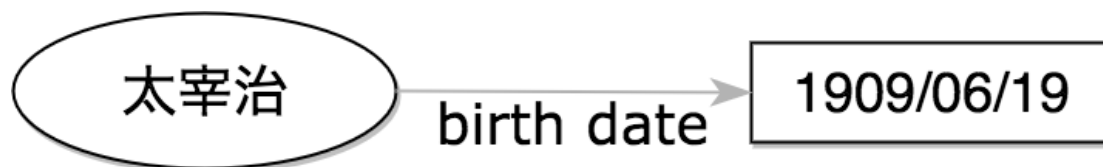


図 4 : 文字列を含むトリプルの例

³ rdf:は”http:www.w3.org/1999/02/22-rdf-syntax-ns#type”の接頭辞

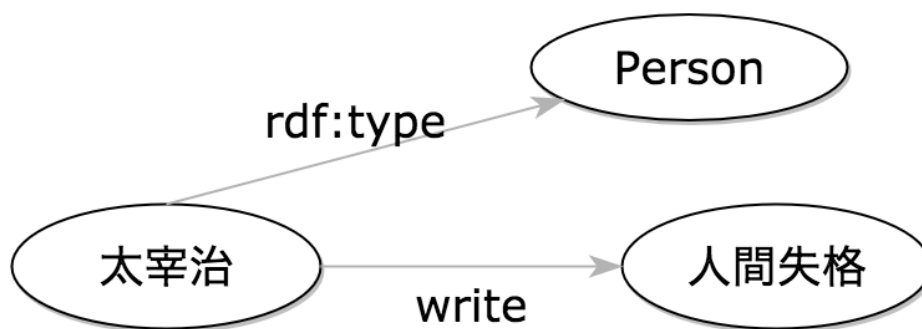


図 5：クラスを含む RDF グラフの例

2.3 メタデータスキーマとインスタンス

メタデータはその目的にあわせて多種多様なものが作成されている。それらメタデータは目的に応じて異なる構造を持っている。その構造はメタデータスキーマによって定義されている。メタデータスキーマとはメタデータの構造の定義のことであり、語彙定義と記述規則を合わせたものである [8]。言い換えるとメタデータを記述する際にどのようなクラスをあたえるのか、どのようなプロパティを用いるのか、そのときにどのような制約をもたせるのかといった仕様のことである。語彙定義ではメタデータタームの基本的な使い方や説明が定義される。一方で記述規則では記述する対象に合わせて、使用するメタデータタームやそのメタデータタームの制約条件等を定義する。この2つを組み合わせることによって記述対象に合わせたメタデータの構造を設計することができる。そしてメタデータスキーマを基に作られたメタデータをインスタンスと呼ぶ。図 6 にメタデータスキーマとインスタンスの関係図を示す。図 6 のように、メタデータ語彙(同一の名前空間を持つ語彙定義の集合)から選ばれたメタデータタームの使用方法を制約によって定義するのが記述規則である。2.3.1 項で語彙定義を、2.3.2 項で記述規則を説明する。

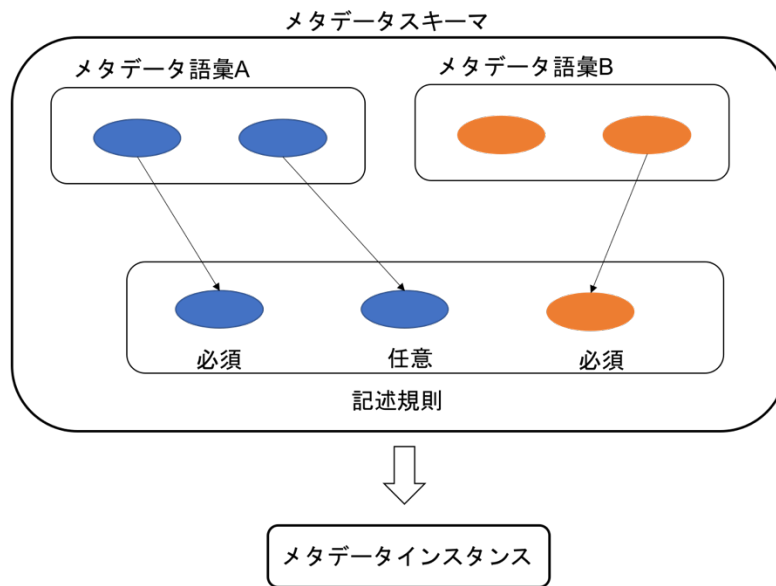


図 6：スキーマとインスタンスの関係図

2.3.1 語彙定義

様々なコミュニティによってメタデータタームの語彙定義が作成されている。語彙定義は基本的にライセンスフリーであるため、新しくメタデータを作る際には目的に合わせて既存の語彙定義を利用することができる。

語彙定義の例として表 1 に FOAF [9] の語彙定義の一部を示す。FOAF は Friend of a friend の略で、人や組織の情報に関するメタデータを記述するために作られた語彙である。表 1 の名称欄では「foaf:」と表記されているがこれは「<http://xmlns.com/foaf/0.1/>」の名前空間の接頭辞である。原則としてメタデータタームは HTTP URI で名前を与えられており、参照解決可能でなければならない。次に分野を問わずに汎用的にメタデータの記述に使われる DCMES (Dublin Core Metadata Element Set [10]) の語彙定義の一部を表 2 に示す。このように特定の目的のためではなく汎用的な利用を目的とした語彙も存在する。なお「dc」は「<http://purl.org/dc/elements/1.1/>」の接頭辞である。

RDF 記述上ではメタデータターム自身は URI のみで構成され、その種類や説明といった情報は表 1、表 2 のように語彙定義で記述される。したがってメタデータターム利用者は語彙定義を参照し、そのタームの基本的な情報を確認する。

語彙定義はコンピュータによる処理を容易にするために、それ自身も RDF で記述される。したがって語彙定義を記述するためのメタデータタームが存在する。RDFS (RDF Vocabulary Description Language, RDF Schema [11]) や OWL (Web Ontology Language [12])、SKOS (Simple Knowledge Organization System [13]) などが挙げられる。簡単に説明すると SKOS は概念階層を表現するための語彙であり、RDFS と OWL はクラスとプロパティを用いて概念モデルを記述するための語彙を提供する [14]。これらの語彙を利用す

ることで、メタデータタームを定義できる。メタデータを作成する際には既存の語彙を再利用することも、自ら定義することも可能である。目的に合わせて選択することが求められるが、既存のものを再利用することがメタデータの相互運用性の観点から推奨されている。

表 1：FOAF の語彙定義の一部

名称	種類	説明
foaf:Agent	クラス	Agent を表す。人やグループ、ソフトウェア等
foaf:Person	クラス	人物を表す。存命中、故人、架空を問わない
foaf:Organization	クラス	組織を表す。
foaf:Document	クラス	文書を表す。
foaf:name	プロパティ	事物の名前を指す。
foaf:title	プロパティ	称号を指す。Mr, Mrs, Dr. 等
foaf:interest	プロパティ	関心のある事物に関する文書を指す。

表 2：DCMES の語彙定義の一部

名称	種類	説明
dc:contributor	プロパティ	リソースに貢献している個人、組織を指す。
dc:coverage	プロパティ	リソースが関わる範囲、領域を指す。
dc:creator	プロパティ	作成した責任者を指す。
dc:date	プロパティ	リソースの出来事に関する期間を指す。
dc:description	プロパティ	リソースの説明を指す。
dc:language	プロパティ	リソースの言語を指す。
dc:publisher	プロパティ	リソースの公開に責任を持つ個人、組織を指す。
dc:subject	プロパティ	主題を指す。
dc:title	プロパティ	与えられた名前を指す。
dc:type	プロパティ	リソースの性質やジャンルを指す。

2.3.2 記述規則

記述規則とはメタデータの記述対象とその記述対象が持つ項目に対する制約の定義のことである。つまり記述規則は構造的制約を表す。前項で述べた語彙定義だけでは記述するメタデータの構造や各メタデータタームの利用方法を詳細に理解することは難しいため、記述規則が作成される。記述規則を参照することでスキーマに関する理解を補うことができる。定義される制約の例を以下に挙げる。

- 記述項目名
 - 記述項目の名前. 「タイトル」「著者」など
- メタデータターム
 - 記述に利用するメタデータターム
- 出現回数
 - 項目ごとの出現回数を決める
- 値制約
 - メタデータタームの目的語が文字列であった場合, 従うべきシンタックスやデータ型を決める

記述規則を表現するための形式として DSP (Description Set Profile [15])が挙げられる. DSP は DCMI (Dublin Core Initiative [16])によって提唱されている. DSP では機械可読を考慮して, 記述規則を XML あるいは RDF を用いて記述を行う. 図 7 に XML 形式の DSP の例を示す. この例ではあるリソース(ID が person となっているもの)に対して, foaf:name プロパティが必須で 1 回の記述が必要なこと, その時の目的語はリテラルであることの制約を記述している.

```
<?xml version="1.0" ?>

<DescriptionSetTemplate xmlns="http://dublincore.org/xml/dc-dsp/2008/03/31">

  <DescriptionTemplate ID="person" minOccurs="1" maxOccurs="1" standalone="yes">

    <ResourceClass>http://xmlns.com/foaf/0.1/Person</ResourceClass>

    <StatementTemplate minOccurs="1" maxOccurs="1" type="literal">

      <Property>http://xmlns.com/foaf/0.1/name</Property>

    </StatementTemplate>

  </DescriptionTemplate>

</DescriptionSetTemplate>
```

図 7 : XML 形式の DSP の例 [15]

DSP を作成することは XML や RDF に関する専門性が必要になり、馴染みのない人には難しい。この問題に対処するために、メタデータ情報基盤構築事業では簡易 DSP と呼ばれる誰でも DSP が作成できる記述方法を提唱している [8]。簡易 DSP ではテンプレートに値を記入していく形で DSP が作成可能である。図 8 に簡易 DSP の記述例を示す。DSP よりも構造が把握しやすく、記述が容易なことがわかる。

最後に実際に利用されている記述規則の例として表 3 に DPLA(Digital Public Library of America [17])の記述規則の一部を示す。この記述規則は `dpla:sourceResource` クラスに属するリソースの記述規則である。`dpla:sourceResource` クラスに属するリソースが「Alternate Title」という項目をプロパティ「`dc:alternative_sourceResource.alternative`」で記述していること、その使い方、使用できる回数が 0 から任意の回数であること等が読み取れる。

[@NS]						
<code>dcndl</code>						<code>http://ndl.go.jp/dcndl/terms/</code>
<code>ndlsh</code>						<code>http://id.ndl.go.jp/auth/ndlsh/</code>
<code>bsh</code>						<code>http://id.ndl.go.jp/auth/bsh/</code>
<code>ndlbooks</code>						<code>http://iss.ndl.go.jp/books/</code>
<code>@base</code>						<code>http://ndl.go.jp/dcndl/dsp/biblio</code>
[MAIN]						
#項目規則名	プロパティ	最小	最大	値タイプ	値制約	説明
書誌 ID	<code>foaf:Document</code>	1	1	ID	<code>ndlbooks:</code>	文書の ID
タイトル	<code>dcterms:title</code>	1	1	構造化	#構造化タイトル	文書の表題
著者	<code>dcterms:creator</code>	0	1	構造化	<code>foaf:Agent</code>	文書の作者
発行日	<code>dcterms:issued</code>	1	1	文字列	<code>xsd:date</code>	文書の発行日
主題	<code>dcterms:subject</code>	0	-	参照値	<code>ndlsh: bsh:</code>	文書の主題
[構造化タイトル]						
#項目規則名	プロパティ	最小	最大	値タイプ	値制約	説明
リテラル値	<code>xl:literalForm</code>	1	1	文字列		タイトル自身
読み	<code>dcndl:transcription</code>	0	1	文字列		タイトルの読み

図 8：簡易 DSP の記述例

表 3 : DPLA の記述規則の一部

Label	Source	Property	Sub-property of	Range	Usage	Vocab Schema	Syntax Schema	Obligation
Alternate Title	P	dcterms:alternative .sourceResource.alternative	dc:title		Any alternative title of the described resource including abbreviations and translations. Literal.			0 - n
Collection*	D	dcterms:isPartOf, .sourceResource.collection	dc:relation	dc:mimeType: Collection	Collection or aggregation of which described resource is a part. Ref.			0 - n
Contributor	D	dcterms:contributor .sourceResource.contributor		edm:Agent	Entity responsible for making contributions to described resource. Ref.			0 - n
Creator*	D	dcterms:creator .sourceResource.creator		edm:Agent	Entity primarily responsible for making described resource. Ref.			0 - n
Date*	P	dc:date .sourceResource.date		edm:Time Span	Date value as supplied by Data Provider. Ref.			0 - n

2.4 Linked Open Data

セマンティックウェブを実現するための取り組みに Linked Data を普及する活動がある [18]. Linked Data は標準化されたデータ表現(RDF)とアクセス方式(SPARQL [19])を持ち、互いにリンクされたメタデータのデータセットである。Linked Data は下記の基本原則に従う [20].

1. *Use URIs as names for things.*
「事物には URI で名前をつけること」
2. *Use HTTP URIs so that people can look up those names.*
「参照解決のために HTTP URI を利用すること」
3. *When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).*
「URI が参照された時, RDF や SPRAQL 等の標準技術を使用して情報を提供すること」
4. *Include links to other URIs. so that they can discover more things.*
「より多くの事物を発見できるように, リンクを含めること」

Linked Data かつオープンライセンスで公開されているデータを LOD (Linked Open Data)という。2006 年ごろから LOD を公開する取り組みは始まっており, 2013 年の G8 サミットでは LOD として政府データを公開することを推奨する”オープンデータ憲章”が合意されるなど, LOD はデータ公開の標準として期待をされている [2]. LOD cloud (Linked Open Data Cloud) [1]という公開した LOD を登録するサイトがあり, 有名な LOD データ

セットはほとんどが登録されている。LOD cloud では登録されるデータセット数は増加しており、2018年12月現在で1231件のデータセットが登録されている。図9にLOD cloudに登録されているデータセットのリンク関係を可視化したグラフを示す。図9のグラフではノードはデータセット、エッジはデータセット間のリンク関係を表す。このグラフからたくさんのデータセットがリンク関係にあるのがわかる。地理的情報、政府が公開する情報、生命科学など様々な分野のメタデータがLOD cloudでは登録されており、直接ないし他のデータセットを介してリンク関係にあることがわかる。

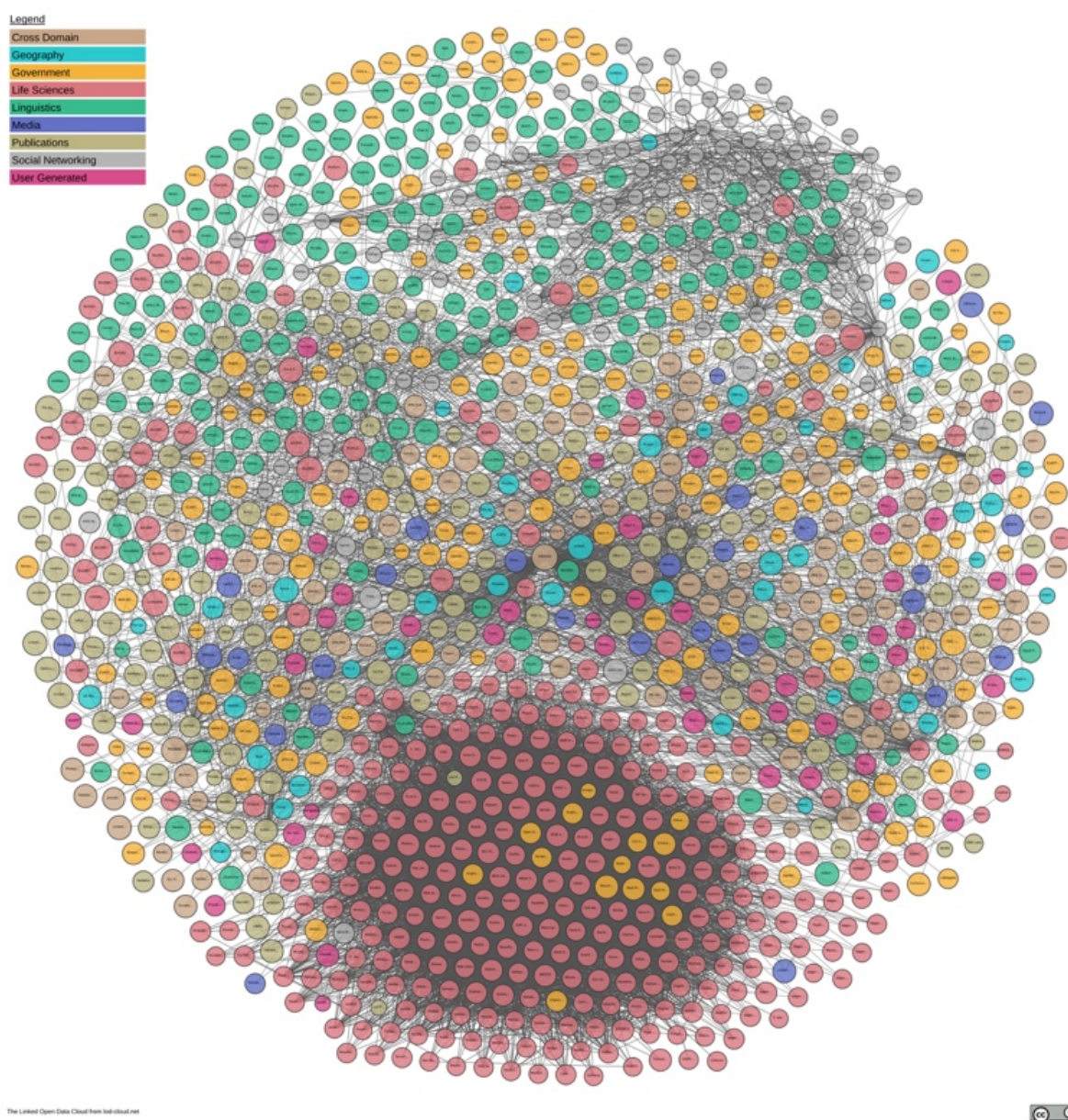


図9：LOD cloud グラフ [1]

3 メタデータタームの探索と用法理解

LOD を作成する時, あるいは利用するときのメタデータタームの問題について述べる.

3.1 LOD 作成時のメタデータターム探索における問題

2.3.1 項で述べたように, メタデータを作成するときには目的に即したメタデータタームを既存の語彙から再利用, あるいは独自で定義する必要がある. メタデータの相互運用性⁴の観点では既存の語彙を再利用し, メタデータタームを共有することが好ましい. しかし既存の語彙を再利用する場合, メタデータタームの探索が難しいという問題がある. 語彙定義は作成された後ウェブ上に公開されるのみで, 特定の機関に対しての登録は義務付けられていない. したがって語彙定義はウェブ上に散在しており, メタデータターム探索者は語彙定義を自力で見つけなければならない. Google 等のウェブ検索では見つけることが難しく, その探索を支援するシステムに LOV (Linked Open Vocabularies) [21]がある.

LOV はウェブ上に公開されている語彙定義を収集したデータベースとなっており, キーワード検索を備えている. 例えば”name”をキーワードとしてプロパティ検索すると”rdfs:label”⁵, ”foaf:name”の名前付けのためのメタデータタームが結果として出てくる. しかし文献 [4]によると, 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必要最小限しか記述されていないものが存在するため, 検索キーワードによっては適切なメタデータタームの候補が結果に出ない場合がある.

また LOV に登録されているメタデータタームだけでも 2018 年 12 月現在 61588 個のメタデータタームが登録されており, 全てのメタデータタームを確認することは現実的ではない. さらには語彙定義を作成する上で標準となっているガイドラインが存在しないため, 語彙定義の品質にはバラツキがある. 必要最低限度のラベルしか提供しないものから, 利用方法の詳細なコメントを記述するものまである. その他にもメタデータターム探索者がラベルとして適切だと考える単語が, メタデータタームのラベルとして適切ではないケースもある. こうした要因から目的に即したメタデータタームの探索は, 知識のある専門家にも難しい作業である.

⁴ メタデータの相互運用性とは様々な組織の垣根を超えてそのメタデータを利用できる能力に関する特性のこと

⁵ ”rdfs:”は”<http://www.w3.org/2000/01/rdf-schema#>”の接頭辞

3.2 LOD 利用時のメタデータタームの用法理解における問題

公開されている LOD を情報源として利用する際にもまた、メタデータスキーマが公開されていないためメタデータタームの用法理解が難しいという問題が存在する。メタデータタームの用法とは、語彙定義や記述規則に記述されるそのメタデータタームの使い方のことである。プロパティの場合、そのプロパティの目的語がリソースなのかリテラルなのか、リソースであった場合、何のクラスに所属するリソースなのかといった情報である。例えば foaf:name プロパティは、名前を記述するのに使用され、目的語にはリテラルを持つ。

通常 RDF データの利用には RDF 標準クエリ言語である SPARQL が用いられる。SPARQL はグラフパターンによるパターンマッチによって問い合わせを行うため、事前にスキーマを知っておかなければ適切なクエリは書けない。したがって使われる各メタデータタームの用法を事前に理解しておく必要がある。本来であれば2.3節で述べたメタデータスキーマが用法理解のための役割を担っている。しかし現状ではメタデータスキーマを公開していない LOD データセットは多く存在する。ほとんどの場合、LOD 利用者が自力でメタデータタームの使い方を把握しなければならない。自力でメタデータタームの用法を理解することは、繰り返し SPARQL クエリを実行し1つ1つ用法を理解していく、専門性と慣れが必要な煩雑な作業である。

SPARQL クエリの例として図 10 に人の名前を問い合わせる SPARQL クエリを示す。このクエリは「foaf:Person クラスに所属するリソースの rdfs:label プロパティの目的語の集合」をグラフパターンで問い合わせている。このクエリを書くには「人という実態が foaf:Person クラスに所属することで表現されていること、人の名前を表すプロパティには rdfs:label が利用されており目的語に文字列で名前が与えられること」を事前に知っておかなければならない。

また記述規則は公開されていなくても、語彙定義には参照解決可能な場合がある。その場合新たに別の問題として、語彙定義とインスタンスでの用法が異なることがある。すると元の語彙定義に記述される用法ではなく、LOD におけるそのメタデータタームの用法を理解する必要がある。


```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?o
WHERE{
    ?s rdf:type foaf:Person.
    ?s rdfs:label ?o.
}

```

図 10：人の名前を問い合わせる SPRAQL クエリ例

3.3 メタデータターム間類似度による問題の解決支援

メタデータタームの類似関係がわかると探索と用法理解の役に立つことが期待できる。メタデータタームの探索における問題では類似する複数のメタデータタームを提示することで、目的に即したメタデータタームの発見に繋がりがやすくなる。例えばメタデータターム探索者があるターム A を記述するメタデータのためのターム候補としているとする。この時ターム A と類似度の高い複数のターム{B,C, ...K}を探索者に提示することができれば、探索者にとって、よりメタデータ記述に適切なタームの発見につながる可能性がある。また用法理解における問題でも、似た意味をもつすなわち似た用法で利用されているメタデータタームを提示することで理解を助けることができる。例えば利用者があるタームの用法理解が困難であった時、先程同様に似た類似度の複数のタームを提供することで、比較が可能になり用法理解を支援することができる。

上記の理由から本研究ではメタデータターム間の類似関係がわかると探索と用法理解の役に立つと考えている。従ってターム間の類似度を算出し、それを利用者へ提示することでタームの探索と用法理解の問題の解決支援が行える。

また本研究では分布仮説に基づく意味的な類似関係を扱う。分布仮説とは単語の意味はその単語が出現した際の周囲の単語によって決まるという考えである。これをメタデータタームに置き換えると、あるタームの意味は周囲のノードやエッジによって決まると考えることができる。したがって周囲のノードやエッジが類似しているターム同士は意味的な類似度が高いと考えることができる。図 11, 図 12 に類似するトリプルの例を示す。これらのトリプルを比較すると、プロパティ「write」とプロパティ「make」は、ラベルは異なるが周囲のノードやエッジが類似しており、タームの意味は似ていると考えることができる。

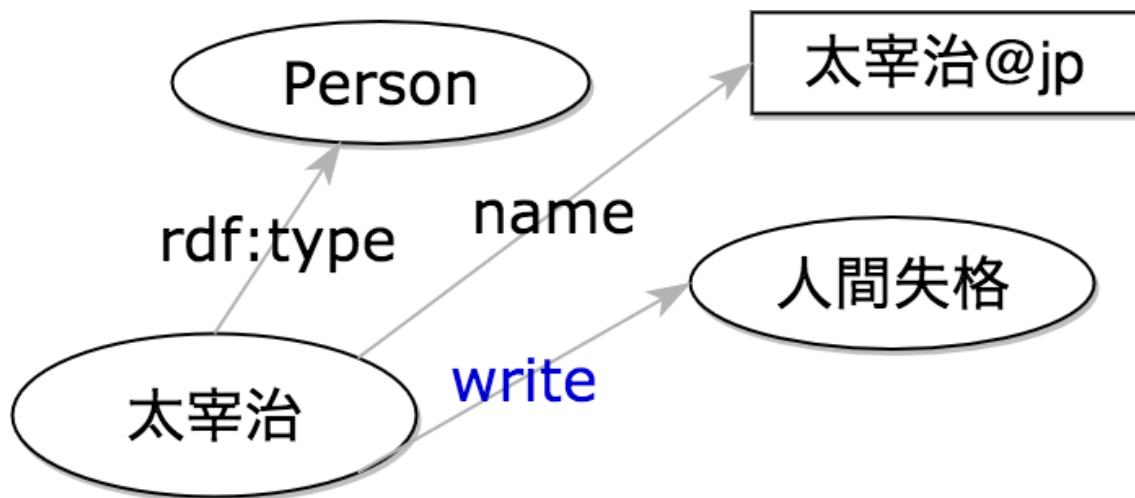


図 11：トリプル例 1

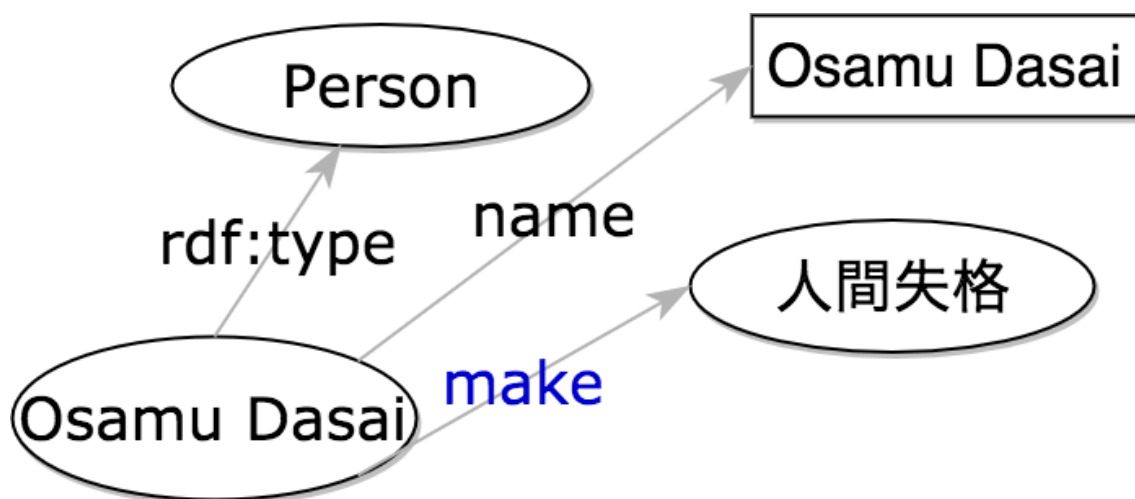


図 12：トリプル例 2

4 グラフ構造を利用した類似度算出手法の提案

本研究ではメタデータタームの探索, 用法理解の問題を解決するためにメタデータターム間の類似度算出手法を提案する. 2組の LOD を入力としてそこで利用されるメタデータターム間の類似度を算出する. LOD すなわちメタデータインスタンスのグラフ構造に注目し, メタデータターム間の類似度を算出する. また類似度を算出するにあたってメタデータターム間だけではなくエンティティ⁶の類似度も併せて算出する. 本手法は入力が複数の LOD の場合に対しても適用可能であるが, 今回は入力を2つとしている.

4.1 提案手法概要

本手法は一部ノード, エッジの共有を伴って異なるメタデータインスタンスを結合し, グラフノードの分散表現を取得できるエンベディング手法を適用することで, 各メタデータターム, エンティティに対応する表現ベクトル取得する. そして取得した表現ベクトル間のコサイン類似度を算出し, それをメタデータターム間, エンティティ間の類似度とする.

エンベディング手法とはグラフ構造を保存して, グラフのノードに対する表現ベクトルを取得する手法である. 本研究では分布仮説⁷に基づき, メタデータタームの意味は周囲のノードによって決まると考えた. つまり似た意味を持つメタデータタームは周囲のノードが似ていると考えた. そうであれば2つの LOD を一部ノードの共通を伴って結合すれば, 似た意味をもつメタデータタームは周囲のノードを共有するはずである. 4.1.1 項で後述するが近傍のノードを共有している状態でエンベディングされたノード組は, 表現ベクトルの類似度が高くなる. したがって結合した LOD にエンベディング手法を適用することでメタデータターム, エンティティ間の類似度を算出できる.

図 13 に表現ベクトル取得までの提案手法の概要図を示す. 図 13 からわかるように本手法では, LOD の結合前に重み付きグラフへと変換している. これは不要なリテラルを削除することや, ノード間の関係によって重み付けを行うなどの狙いがある. またプロパティの表現ベクトルを取得するために, エッジからノードへと変換している. 入力する2つの重み付きグラフは1つの重み付きグラフへと結合する. その際に共有されるノードを増やすために既存のマッチング手法を適用する. その後グラフエンベディング手法である LINE 手法(後述)を適用し, ノードの分散表現を取得する. 各ノードに対応する表現ベクトル間でコサイン類似度を算出し, それをメタデータターム間, エンティティ間の類似度とする.

⁶ 何らかのクラスに所属するリソースを指す.

⁷ 単語の意味はその単語が出現した際の周囲の単語によって決まる

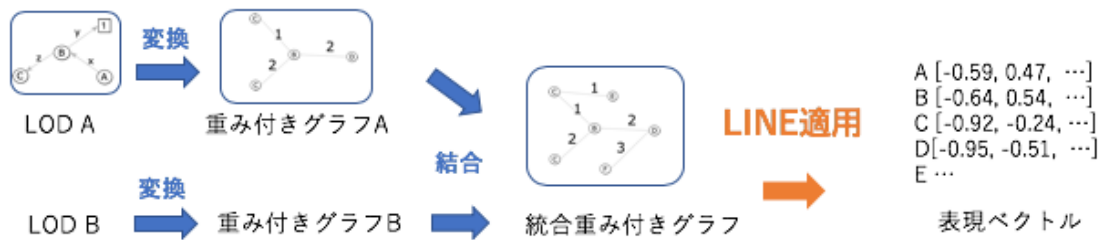


図 13：提案手法の概要図

4.1.1 エンベッディング手法について

エンベッディング手法とはグラフのノードの分散表現を得る手法である。またエンベッディング手法はその手法ごとに異なるグラフ構造を保存して表現ベクトルを取得する [22]。本研究は LINE [23]手法を利用する。LINE 手法では *first-order proximity* と *second-order proximity* を保存し、ネットワーク(重み付きグラフ)をベクトル空間へと埋め込む。

first-order proximity とは 2 組のノード間の局所的な近さを表す尺度である。エッジを持つノード間の *first-order proximity* はそのエッジの重みとなり、エッジを持たないノード間の *first-order proximity* は 0 である。現実のネットワークにおいて *first-order proximity* は通常 2 つのノード間の類似度を表すといえる。例えばソーシャルネットワークにおいて友人関係を持つ 2 人の組は共通の趣味や関心事を持っている。

second-order proximity とは 2 つのノード間でどれだけ近傍(1Hop 先のノード集合)が似ているのかの尺度である。例えばソーシャルネットワークにおいてたくさんの友人を共有する 2 人は、共通の関心事を持っており、友人関係である可能性も高い。

形式的に説明すると、グラフを $G = (V, E)$, V をノードの集合, E をノード間のエッジの集合, それぞれのエッジ $e \in E$ を順序組 (u, v) , その重みを $w_{u,v}$ と表すとする。すると 2 つのノード u, v に対して重み $w_{u,v}$ がノード u, v 間の *first-order proximity* となる。また $p_u = (w_{u,1}, \dots, w_{u,|V|})$ が u の他の全ノードへの *first-order proximity* を表すとする。図 14 のネットワークグラフの例ではノード 6 とノード 7 の *first-order proximity* が高く、ノード 5 とノード 6 の *second-order proximity* の高いことが予想できる。

first-order proximity を保存するエンベッディング手法では、*first-order proximity* の高いノード組の表現ベクトルは距離が近くなる。つまりそのノード組のコサイン類似度が高くなる。またその時の重みが大きいほど関係は保存され、表現ベクトルの類似度は高くなる。同様に *second-order proximity* を保存する手法では、*second-order proximity* の高いノード組すなわち近傍のノード集合が似ているノード組ほど、エンベッディング後の類似度は高くなる。LINE 手法では *first-order proximity* と *second-order proximity* を保存するエンベッディング手法が別々で実装されており、最後に両実装で取得した表現ベクトルを結合する。

本研究では *second-order proximity* のみを保存したエンベッティングを行いたいのでLINE手法のうち *second-order proximity* を保存する実装のみを扱う. この理由は元が異なるLOD をマッチングによって結合することから 等価関係にあるノードがエッジでつながることは無く別のノードを介してリンクされるであろうと, 予想するためである.

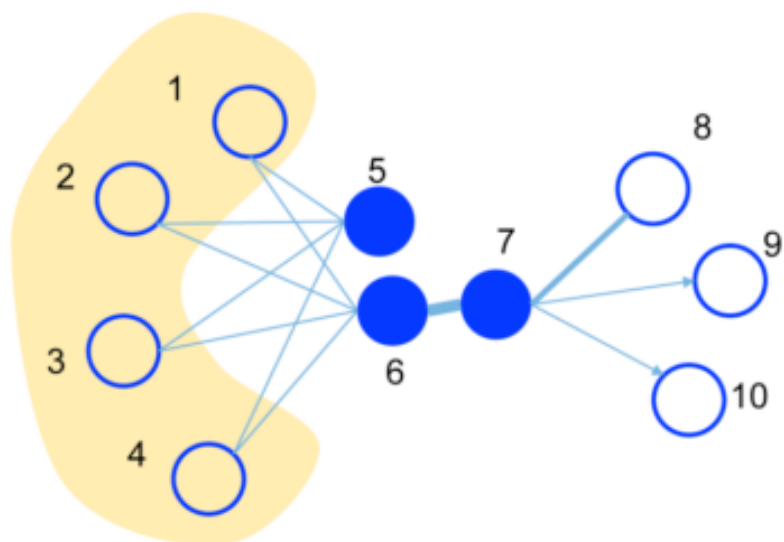


図 14: ネットワークグラフの例([23]より)

4.2 重み付きグラフの作成

変換前の LOD について, R を元の RDF データ, R_p を R に含まれるプロパティの集合, R_c を R に含まれるクラスの集合, R_e を R に含まれるエンティティの集合とする. ただし $R_e = \{s | (s, rdf:type, o) \in R \wedge s \notin R_p \wedge s \notin R_c\}$ とする. 一方で作成する重み付きグラフを $G = (V, E, W, g)$ と表し, V をノードの集合, E をエッジの集合とする. また枝接続関数 g を $E \rightarrow \{(i, j) | i \in V, j \in V\}$ とし, W をエッジ $e_{i,j} \in E$ に対応する重み $w_{i,j}$ の集合とする. 下記にまとめる.

変換前の LOD

R : RDF データ

R_p : R に含まれるプロパティの集合

R_c : R に含まれるクラスの集合

R_e : R に含まれるエンティティの集合,

ただし $R_e = \{s | (s, rdf:type, o) \in R \wedge s \notin R_p \wedge s \notin R_c\}$

作成する重み付きグラフ

$G = (V, E, W, g)$

V : ノードの集合

E : エッジの集合

$g : E \rightarrow \{(i, j) | i \in V, j \in V\}$ の枝接続関数

W : エッジ $e_{i,j} \in E$ に対応する重み $w_{i,j}$ の集合

LOD から重み付きグラフを作成するアルゴリズムを図 15 に示す. 重み付きグラフを作成する上での工夫を述べる. 重み付きグラフの作成では極力グラフ構造を変更しないように, エッジであるプロパティをノードへと変換している. これによってプロパティも表現ベクトルを取得できるようになり, 類似度が算出できるようになる. またリテラルノードは削除している. 本研究ではリソースノードと比較してリテラルノードはノードの意味を形成するにあたって重要度が低いと考えた. リテラルノードを削除することでグラフ全体のノード数を減らし, 手法適用にかかる時間を短くする狙いがある. エッジに与える重みは, クラス関係のものを大きく, リソース間のものを中くらいに, プロパティ関係のものを小さく設定した. 1~14 行までが上述の処理を行っている. 15~18 行は 1 つノードから出る重みの総和が 1 になるような正規化を行っている. エンティティごとのトリプル数にバラツキがあることを考慮している. 最後に 19 行目で存在する全てのエッジに対してその逆方向である, ターゲットからソースへのエッジに同じ重みを与えることで, 有向グラフを無向グラフへと変更している. これはプロパティが主語エンティティを共有できるようにするためである.

```

1   $W = 0$ 
2  each entity in Re
3      n = number of edges
4      each pair of property and object
5          if property == rdf:type
6               $w_{entity, object} += n$ 
7              break
8          else if object  $\in R_c$  or  $R_p$ 
9               $w_{entity, object} += n$ 
10         else if object  $\in R_e$ 
11              $w_{entity, object} += n/2$ 
12         else if object == URI
13              $w_{entity, object} += n/3$ 
14          $w_{entity, property} += 1$ 
15 each i in V
16      $sum = \sum_{(i,k) \in E} w_{i,k}$ 
17     each j in (i,j)  $\in E$ 
18          $w_{i,j} = w_{i,j}/sum$ 
19  $w_{j,i} += w_{i,j} \in E$ 

```

図 15 : 重み付きグラフ作成のアルゴリズム

4.3 事前のマッピングとグラフの結合

重み付きグラフ作成後はそのグラフを結合する。結合の際には元のグラフが異なるノード同士が近傍を共有できるように一部のノード(プロパティ, クラス, エンティティ)の等価関係のマッピングを行う。本手法では既存のマッピング手法を適用することを想定している。類似度を算出する前の作業なので再現率よりも精度の高い手法が好ましい。マッピング手法は適用する LOD によって結果が大きく異なるため、適宜好ましいマッピング手法を選びたい。本研究の評価実験においては、URI の最後の単語文字列の完全一致によりマッピングを行っている(図 16)。文字列の完全一致を利用するマッピング手法は、適用する LOD に結果が大きく左右されず、再現率は期待できないが高い精度が期待できる。(5.1.1 項後述)。

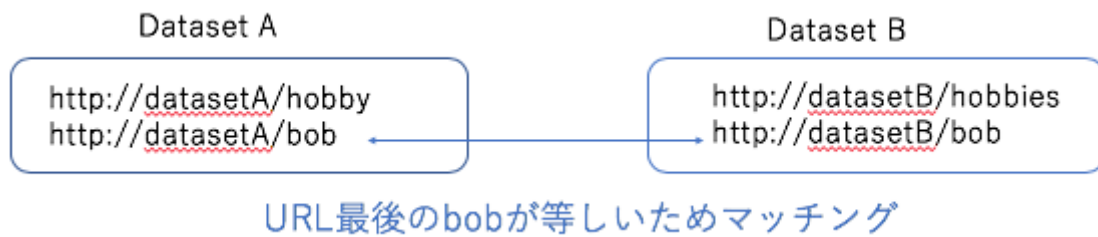


図 16 : URI の最後の単語文字列の完全一致によるマッチング

4.4 類似度の算出方法

LINE 手法適用後に取得した表現ベクトルから類似度を算出する。最初に 2 つの LOD のそれぞれのクラス、プロパティ、エンティティごとに表現ベクトルを分ける。その後、一方の LOD のクラスの表現ベクトルともう片方の LOD のクラスの表現ベクトル同士で、掛け合わせてコサイン類似度を算出し、それをクラス間の類似度とする。同様にプロパティ間、エンティティ間でもコサイン類似度を算出し類似度とする。具体的に説明する。一方の LOD データセットを A、もう一方の LOD データセットを B とした時、データセット A のプロパティの表現ベクトル集合を $PA = \{p_{a1}, p_{a2}, \dots, p_{a|PA|}\}$ と表記する。同様にデータセット A のクラスの表現ベクトル集合を CA 、エンティティのベクトル集合を EA とする。データセット B のプロパティ、クラス、エンティティのベクトル集合も同様にそれぞれ PB , CB , EB とする。コサイン類似度は $\{PA, PB\}$ 間と $\{CA, CB\}$ 間, $\{EA, EB\}$ 間で算出する。本手法ではクラス、プロパティ間の類似度の算出等を行わない。

5 関連研究

本章では関連研究であるマッチング手法と先行研究を紹介する。マッチング手法はメタデータターム、エンティティの等価関係を発見する手法である。メタデータターム間の関係を扱う研究ではマッチング手法が有名である。しかしマッチング手法では1対1の関係を発見することになるため、メタデータタームの探索や用法理解においては、類似度の高いものを複数提示したほうがよいと本研究では考えた。またマッチング手法の中には何らかの類似度を算出し、それを基に等価関係を見つける手法もあり、その点において本研究とマッチング手法は非常に似ている。なおグラフエンベディングを利用したマッチング手法は現在確認できないため、本研究の新規性は十分にあると考えている。

5.1 マッチング手法

マッチング手法は1対1の等価関係を発見する手法である。特にメタデータタームの等価関係を発見するものをスキーママッチングと呼ぶ。またエンティティのみの等価関係を扱うものをインスタンスマッチングと呼ぶ。マッチング手法には OWL や SKOS など詳細に記述された語彙定義に適用するものも多いが、本研究ではそうした語彙定義に参照解決できないような既存の LOD への適用を考えているため、そうした手法の紹介はあまり行わない。また OWL など記述されるオントロジ(木構造を持った概念階層)を対象としたマッチング手法も紹介を控える。LOD は木構造を持つとは限らない。本説は主に文献 [24] [25]を参考にしている。

5.1.1 文字列を基にしたマッチング手法

最も一般的に用いられているマッチング手法。ノードやエッジなどの要素単位で行う。つまりメタデータタームやエンティティに与えられている URI や、特定のプロパティ(foaf:name 等)の指すラベルの文字列を比較する方法。単純に文字列が一致しているのかどうかだけではなく、編集距離を算出し閾値を超えるかどうかでマッチングの判定を行う手法もある。編集距離以外にも接頭辞一致、接尾辞一致、n-gram などの指標を用いることができる。また事前にステミングを行い、単語を標準形に戻して指標を用いて判定する方法がある。その他にソーラスや概念辞書、翻訳辞書などの知識資源を利用することができる。また語彙定義などに記載されている複数の単語列(文)に対しても、単語ごとに指標を算出することで比較が可能になる。本研究の評価実験ではリソースの URI に与えられている最後の単語文字列が完全に一致しているかどうかでマッチングの判断を行っている。

5.1.2 グラフ構造を利用したマッチング手法

グラフが木構造を持っていた場合、グラフマッチの手法も利用される。これは一方のグラフがもう一方のグラフのどこに対応するかを計算し、ノード間の対応を発見する手法である。グラフ全体の対応関係ではなく、局所的なグラフ構造を利用する事も可能である。近隣のノードやエッジの類似度が高ければそのノード対の類似度が高いと判定するやり方である。クラス同士を比較する際に、そのクラス対の上位クラスや下位クラスの類似度を利用するやり方もこちらに含まれる。本研究は局所的なグラフ構造を利用しているといえる。

5.1.3 ドメイン知識を活かしたマッチング手法

ドメイン知識を利用したマッチング手法を紹介する。地理情報の LOD である Linked Geo Data [26]では同じく地理情報を扱う DBpedia [27], Geonames [28] へのリンク生成のためのマッチングにラベルの編集距離と緯度経度から算出される距離を組合わせた指標を用いている [29]。また出版物のマッチングを行う上で ISBN のようなキーを利用することで精度の高いマッチングが期待できる。

5.2 先行研究

弊研究室で行われたメタデータターム間の類似度を算出する研究に [4]がある。この研究では語彙定義から代表となる単語列を抽出し、Word2Vec [30]を利用して表現ベクトルを取得、そのコサイン類似度をメタデータターム間の類似度としている。メタデータタームに関わる文字列の意味から類似度を算出しており、インスタンスでの局所的なグラフ構造から類似度を算出する本研究とは異なる。LOD ではインスタンスがスキーマを遵守しないケースや、スキーマに参照解決できないケースが時折確認される。本研究はそうしたケースに対処すべく、インスタンスだけから類似度を算出することが狙いのひとつである。

6 評価実験

本章では主に提案手法で算出されるクラス、プロパティ、エンティティの類似度の妥当性を確かめるための実験について述べる。また提案手法の実験に加えて、提案手法に変化を加えた場合の実験も同様に行う。これによって重み付きグラフ作成等での工夫の良し悪しを判断する。具体的には(1)重み付けを行わなかった場合、(2)重みの正規化を行わない場合、(3)前処理を一切行わなかった場合、以上の3つの条件を提案手法に加えて実験する。(1)では提案手法で作成する重み付きグラフのノードとエッジの接続関係は変えず、エッジの重みのみ全て1にする変更を加える。(2)は図15のアルゴリズムの15~18行目の重みの正規化の処理を行わない場合である。また(3)は、ノードの事前のマッチングを行わずに結合したLODデータ組を、リテラルを含むトリプルの削除のみ行い、エッジの重みが全て1の重み付きグラフへ変換した場合である。エッジの接続関係はLODから変更していない。つまりプロパティはノードへと変換していないため、プロパティの類似度は算出できない。

提案手法を改善するための実験はこれら以外にも行っており、それについては考察で触れる。

6.1 評価方法

マッチング手法の評価用データセットを用いて、等価関係にある組の類似度が、それ以外の組み合わせと比較したときに上位10件以内に入っているかどうかの再現率を算出する。具体的には一方のLODデータセットをA、もう一方のLODデータセットをBとした時、データセットAのプロパティの表現ベクトル集合を $PA = \{p_{a1}, p_{a2}, \dots, p_{a|PA|}\}$ 、クラスの表現ベクトル集合をCA、エンティティのベクトル集合をEAとする。同様にデータセットBのプロパティ、クラス、エンティティのベクトル集合をそれぞれPB, CB, EBとする。また p_{ai} とPBの類似度集合を $U_{pai} = \{sim(p_{ai}, p_{b1}), sim(p_{ai}, p_{b2}), \dots, sim(p_{ai}, p_{b|PB|})\}$ とする。その他の表現ベクトルに対しても同様にUを使って、もう一方のLODの表現ベクトル集合との類似度集合を表現する。正しい等価関係の組 $\{p_{ai}, p_{bj}\}$ が与えられた時、その類似度 $sim(p_{ai}, p_{bj})$ がそれぞれ U_{pai} と U_{pbi} の中で10番以内に高いかどうかで再現率を判断する。1つの正解組で2個の再現の可否を判断する。

評価用データセットはOAEI(Ontology Alignment Evaluation Initiative [31])が提供しているknowledge graphトラックの評価用データセットを利用する。このデータセットはDbkWikiプロジェクトで、Fandomと呼ばれる特定のトピックに絞ったwikiサイトから生成されたデータセットである。3分野9つのデータセットから構成され、同一分野ごとにデータセット間の組み合わせでの正しい等価関係が与えられる。表4にKnowledge graphトラックを構成するデータセットを示す。実験ではGame, Entertainment分野のデータセットを利用する(実験で利用するデータセットの詳細は付録参照)。なおEntertainment分野はクラス数が少ないが、クラスの等価関係が提供されないため、再現率の問題には含まれない。

い. また事前のマッピングではタームの最後の文字列の完全一致によりマッピングを行った.

表 4 : knowledge graph トラックのデータセット

データソース	分野	#クラス	#プロパティ	#エンティティ
RuneScape Wiki	Game	106	1,998	200,605
Old School RuneScape Wiki	Game	53	488	38,563
DarkScape Wiki	Game	65	686	19,623
Memory Alpha	Entertainment	0	326	63,240
Star Trek Expanded Universe	Entertainment	3	201	17,659
Memory Beta	Entertainment	11	413	63,223
Marvel Database	Book	2	99	56,464
Hey Kids Comics Wiki	Book	181	1,925	158,234
DC database	book	5	177	128,495

6.2 実験環境

手法の実装はプログラミング言語 python(v 3.6.1)を利用して行った. LOD データは RDF ストア Fuseki⁸に格納し, SPRAQL を利用してデータの取得を行った. LINE 手法の適用は著者が公開しているプログラム⁹を利用した. また LINE 手法のパラメータとして 25 億回の学習数で 100 次元へとエンベッティングを行った.

6.3 結果

表 5 に提案手法による上位 10 以内の再現率の結果を示す. 再現率の項目は算出された類似度による比較のみによる再現率を示している. また全体の再現率の項目は文字列によるマッチングも含めた再現率である. またクラス, プロパティ, エンティティの共有率は Jaccard 係数 $J(A, B) = |A \cap B| / |A \cup B|$ により求めている. また比較のために重み付けの工夫を行わない提案手法の結果を表 6 に示す. この手法では基本的に図 15 で示したアルゴリズムに従いグラフを作成するが, エッジの重みは全て 1 としている. 次に正規化を行わない提案手法の結果を表 7 に示す. この手法はエッジの重みの正規化を行わないことのみ提案手法と異なる. さらに比較対象として表 8 に事前のマッピングやプロパティのエッジ化, 重み付け等を行わず, LOD を直接結合し LINE 手法を適用した結果をベースラインとして示す.

⁸ <https://jena.apache.org/documentation/fuseki2/>

⁹ <https://github.com/tangjianpku/LINE>

表 5：提案手法による上位 10 以内の再現率

LOD の組	再現率, 正解数/問題数 (算出された類 似度から判定)	全体の再現率, 正解数/問題数 (文字列のマッ チングも含む)	クラス 共有率	プロパティ 共有率	エンティティ 共有率
Dark, Old	60%, 6/10	97%, 138/142	49%	29%	15%
Rune, Dark	25%, 1/4	98%, 192/195	54%	24%	9 %
Rune, Old	16%, 2/12	93%, 142/152	38%	16%	9 %
Malpha, Mbeta	28%, 4/14	82%, 47/57	0%	14%	11%
Mbeta, Startrek	33%, 2/6	90%, 36/40	0%	17%	5%
Malpha, Startrek	0%, 0/4	88%, 32/36	0%	13%	4%

表 6：重み付けを行わない提案手法の上位 10 以内の再現率

LOD の組	再現率, 正解数/問題数 (算出された類 似度から判定)	全体の再現率, 正解数/問題数 (文字列のマッ チングも含む)	クラス 共有率	プロパティ 共有率	エンティティ 共有率
Dark, Old	30%, 3/10	95%, 135/142	49%	29%	15%
Rune, Dark	0%, 0/4	98%, 191/195	54%	24%	9 %
Rune, Old	16%, 2/12	93%, 142/152	38%	16%	9 %
Malpha, Mbeta	14%, 2/14	79%, 45/57	0%	14%	11%
Mbeta, Startrek	0%, 0/6	85%, 34/40	0%	17%	5%
Malpha, Startrek	0%, 0/4	88%, 32/36	0%	13%	4%

表 7：正規化を行わない提案手法の上位 10 以内の再現率

LOD の組	再現率, 正解数/問題数 (算出された類 似度から判定)	全体の再現率, 正解数/問題数 (文字列のマッ チングも含む)	クラス 共有率	プロパティ 共有率	エンティティ 共有率
Dark, Old	30%, 3/10	95%, 135/142	49%	29%	15%
Rune, Dark	25%, 1/4	98%, 192/195	54%	24%	9 %
Rune, Old	33%, 4/12	94%, 144/152	38%	16%	9 %
Malpha, Mbeta	0%, 0/14	75%, 43/57	0%	14%	11%
Mbeta, Startrek	0%, 0/6	85%, 34/40	0%	17%	5%
Malpha, Startrek	0%, 0/4	88%, 32/36	0%	13%	4%

表 8：前処理を行わなかった場合の上位 10 以内の再現率

LOD の組	全体の再現率, 正解数/問題数
Dark, Old	3%, 4/142
Rune, Dark	5%, 9/195
Rune, Old	5%, 8/152
Malpha, Mbeta	0%, 0/57
Mbeta, Startrek	0%, 0/40
Malpha, Startrek	0%, 0/36

7 考察

考察では最初に評価実験に関しての考察を行う(1). 次にその評価方法が正しかったのか評価方法に関しての考察をする(2). そして今回提案した類似度算出手法とその改善についての考察を行う(3). また LINE 手法のパラメータに対する考察も行う(4). 最後に現実の LOD での本手法の適用について述べる(5).

(1) 評価実験の結果に関して

表 5, 表 8 から前処理を行う提案手法は算出される類似度が改善されることがわかる. 実験全体において, 評価用データセットが提供している正しい等価関係の組が, 文字列のマッチングにより再現されているため全体の再現率が高くなっている. 同時に算出された類似度を評価するための問題数が少なくなってしまう. 表 6 から重み付けを行わない場合, 提案手法よりも結果が悪くなっていることが確認できる. したがって単にノード間をエッジで結ぶよりも, 元の LOD のノード関係やプロパティに応じてエッジの重み付けを工夫することで, 算出される類似度の質が良くなることがわかる. また表 7 から重みの正規化を行ったほうが, 結果が良くなる事がわかる. また表 5 からエンティティの共有率が高いほど, 再現率が高い傾向にあることがわかる. これはエンティティ共有率が高いと等価関係にあるリソースが, 近傍(1Hop)のノードを共有できる可能性が高くなるためだと考えられる. またクラスやプロパティの共有率があまり影響しないのは表 4 からわかるようにエンティティと比べて数が少ないためだと考えられる. 表 9 に LOD の組み合わせとクラス, プロパティ, エンティティを合わせた共有率を示す. このようにエンティティの共有率と大きく変わらないことがわかる. 最後に評価用データセット組{Dark, Old}の提案手法による重み付きグラフを調査したところ, 上位 10 位以内に再現できた等価関係にあるリソース組は, 何らかのノード(単数または複数)を共有していることを確認した. 一方で再現できなかったリソース組はノードを共有していないことを確認した.

(2) 評価方法に関して

実験では上位 10 以内の等価関係にある組の再現率を算出した. この 10 という数字は著者が恣意的に定めた値であり, 論理的に導かれた数字ではない. この数字はメタデータタームの探索あるいは用法理解に取り組む人へ, 類似するメタデータターム群を提供した時に, 等価関係にあるメタデータタームを上位 10 以内に含めることができれば役に立つのではないかと考え定めた数字である. 参考までに表 10 に提案手法による上位 n 以内の場合の再現率をそれぞれ示す. 評価方法は今後の課題である.

考えられる他の評価方法の候補は算出された類似度からさらにマッチングを行うことである. マッチング手法としてならば, OAEI のコンペティションに提出されたオントロジーマッチング手法と比較が可能になる. そのコンペティションでは今回実験で使用した knowledge graph トラックにオントロジーマッチングの手法が適用されている. しかしその

比較はオントロジーマッチング手法との比較になるため適切とは言い難い。なぜならばオントロジーは概念階層をメタデータで表現したものであり、木構造を持つなど LOD とは性質が大きく異なる。例えばオントロジーではクラスやプロパティの上位下位の関係は必ず記述されており、プロパティがデータタイププロパティなのか、オブジェクトタイププロパティなのかといった情報も記述される。しかし今回実験で利用した knowledge graph トラックは現実の LOD であるため、そのような情報は記述されておらず、インスタンスの推論から判断しなければならない。そしてインスタンスにおいて、あるプロパティの目的語にリソースとリテラルの両方が来ること¹⁰があるため推論が難しい。またオントロジーマッチングの手法はプロパティのタイプを判断できないとマッチングを行わない。このような事情から OAEI に提出されたオントロジーマッチングの手法の中で knowledge graph トラックのプロパティを1個でもマッチングできた手法は14のうちわずか1つの手法である¹¹。以上を踏まえてオントロジーマッチング手法は比較対象として適切ではないと考えている。代わりに文字列によるマッチングのみを扱うベースラインを設定し比較対象とすると良いかもしれない。その場合は本手法で算出した類似度をどのように利用しマッチングを行うのか考える必要がある。最も単純な方法は類似度が一番高いもの同士をマッチングさせる方法である。

(3) 手法とその改善について

今回の手法では *second-order proximity* を保存するエンベッティング手法を利用しているが、*first-order proximity* の利用も手法の改善になるかもしれない。*first-order proximity* を使っていない理由は、元が異なる LOD であることから等価関係にあるメタデータターム組が直接エッジを持たないためである。別のノードを介してエッジでつながることはあるため、今回は *second-order proximity* のみを扱った。一方であるノード B を介して等価関係にあるメタデータターム組 {A,C} がつながっている時に、*first-order proximity* を保存することで $u_A \cong u_B, u_B \cong u_C$ が成り立つベクトル空間への埋め込みを行えば、 $u_A \cong u_C$ となりうる可能性があるため *first-order proximity* は検討の余地はある。またより広範囲なノード間のグラフ構造を保存するエンベッティング手法の利用も検討できる。例えば [32] の手法では、隣接するノード組をシーケンスとする LINE 手法とは異なり、ランダムウォークによって skip-gram モデル [30] に入力するシーケンスを決定するためランダムに、1~kth order proximities を保存することができる。それによって、より広範囲でノード間の類似度が高くなる埋め込みができる。なお *k-order proximity* についての説明は文献 [33] の定義を参考にしたい。

提案手法は実験結果や原理から、事前マッチングがどれだけできるかで、算出される類似度の質が決まることは明らかである。つまりなるべく高い精度で多くのノードを共有す

¹⁰ 現実の LOD データセットではスキーマが守られていないことが度々確認される

¹¹ <http://oaei.ontologymatching.org/2018/results/knowledgegraph/index.html>

ることで、残りのノードの類似度の質が良くなる。提案手法ではクラス、プロパティ、エンティティのマッチングは行ったが、実はそれ以外のリソースのマッチングは行っていない。それらリソースのマッチングを行うことで更に結果が良くなることが期待できる。さらにリテラルを利用することも1つの手段である。そのLODデータセットに依存するがリテラルは通常リソース数の数倍から数十倍程度の数がある。計算時間を短くするため提案手法では利用せずに重み付きグラフへと含めなかったが、マッチングできるリテラルは含めたほうが算出される類似度が良くなる可能性はある。また全てのリテラルをノードへ組み込むのではなく、ランダムなアルゴリズムで少数のノードを組み込む等の工夫も考えられる。

(4) 実験のパラメータについて

実験では100次元のベクトルへと埋め込んでいるが、この次元数を変更することで結果が変化する。表11に正規化を行わない提案手法の次元数ごとの再現率を示す。表11の結果以外の実験でも、25次元では再現率が落ち50次元、100次元では似たような結果となった。そのため実験では100次元を採用した。またサンプリング数についても同様にいくつかのサンプリング数で実験を行い、250億回以上増やしても結果に改善が見られなくなったのを確認し、250億回としている。エッジの重みについても同様に複数のパターンの重み付けルールの中から最も結果の良かったものを提案手法として紹介している。

(5) 現実のLODへの適用について

最後に本手法の実用性、すなわち現実のLODへの適用について、まずは手法のスケール性について述べる。LINE手法は数十億規模のエッジ数のグラフに対しても1台のマシンで無理なく動作するとされている。2018年12月現在、LOD cloudに登録されていたデータセットのうち外れ値が含まれない1143件を調査したところ、数百規模のエッジ数のデータセットからエッジ数が90億のデータセットまで幅広くあった。図17、図18にLODデータセットのヒストグラム、LODデータセットの累積ヒストグラムをそれぞれ示す。いずれも横軸はエッジ数となっている。これらの図からほとんどのLODのエッジ数は数百万以下であることがわかる。したがってLINE手法の処理時間は問題ない。また実験で利用したような文字列を使ったマッチング手法であれば、前処理にも時間はあまりかからないため、スケール性という意味で提案手法は現実のLODデータセットへ十分に適用できる。しかし表8の結果が悪いうように、多くのリソースの共有がなければ手法は上手く機能しないため、現実のLODへ適用する場合は全く異なる分野のLOD組への適用は難しいことが予想できる。

表 9：LOD 組と合わせた共有率

LOD の組	再現率, 正解数/問題数 (算出された類似度から判定)	クラス + プロパティ+ エンティティ 共有率
Dark, Old	60%, 6/10	15%
Rune, Dark	25%, 1/4	9 %
Rune, Old	16%, 2/12	10 %
Malpha, Mbeta	28%, 4/14	11%
Mbeta, Startrek	33%, 2/6	6%
Malpha, Startrek	0%, 0/4	4%

表 10：提案手法による上位 n 以内の再現率

LOD の組	n = 1 再現率, 正解数/問題数	n = 5	n = 10	n = 20
Dark, Old	20%, 2/10	50%	60%	70%
Rune, Dark	0%, 0/4	25%	25%	25%
Rune, Old	8%, 1/12	16%	16%	25%
Malpha, Mbeta	14%, 2/14	21%	28%	50%
Mbeta, Startrek	0%, 0/6	16%	33%	33%
Malpha, Startrek	0%, 0/4	0%	0%	0%
合計	10%, 5/50	24%, 12/50	30%, 15/50	40%, 20/50

表 11：正規化を行わない提案手法の次元数ごとの再現率

LOD の組	25 次元 再現率, 正解数/問題数	50 次元	100 次元
Dark, Old	30%, 3/10	40%	40%
Rune, Dark	0%, 0/4	0%	0%
Rune, Old	16%, 2/12	25%	16%
Malpha, Mbeta	0%, 0/14	14%	14%
Mbeta, Startrek	0%, 0/6	0%	0%
Malpha, Startrek	0%, 0/4	0%	0%
合計	10%, 5/50	18%, 9/50	16%, 8/50

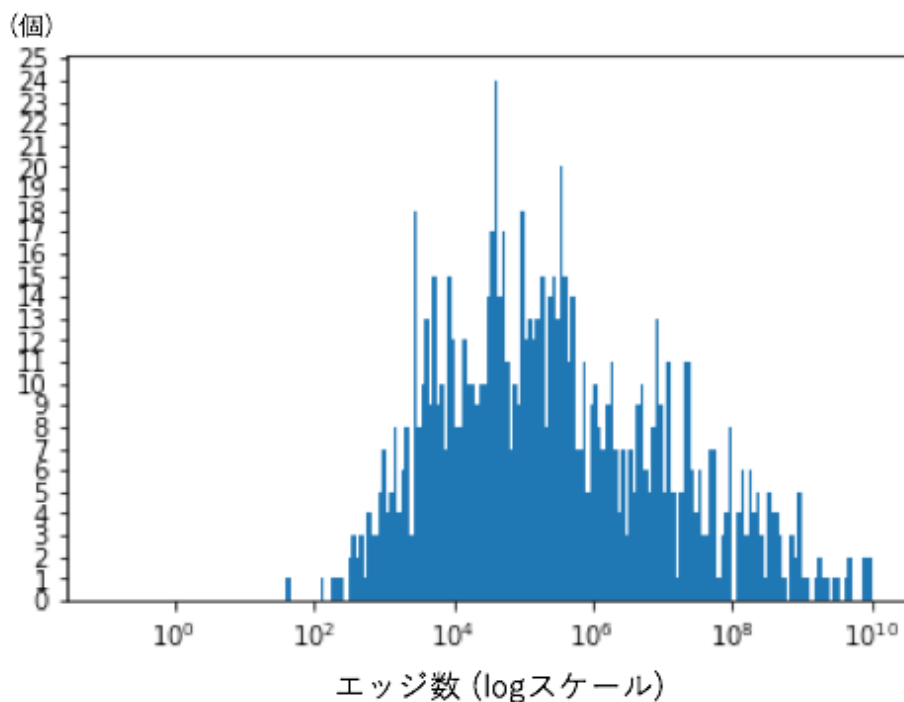


図 17：LOD データセットのヒストグラム

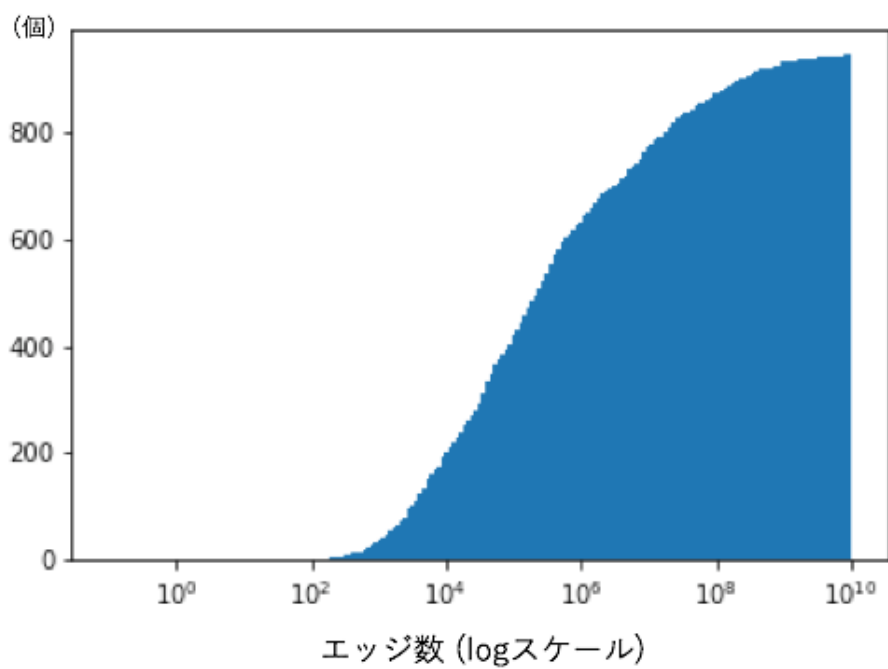


図 18 : LOD データセットの累積ヒストグラム

8 おわりに

本研究ではメタデータタームの探索と用法理解を支援するために、局所的なグラフ構造の一致からメタデータターム間の類似度を算出した。併せてエンティティの類似度の算出も行った。局所的なグラフ構造を保存するエンベディング手法で分散表現を取得し、そのベクトル間のコサイン類似度を計算することでメタデータタームの類似度を実現した。また LOD をそのまま LINE 手法の入力とするのではなく、リテラルの削除やプロパティのエッジ化、ノード間の関係に応じたエッジの重み付けなどの工夫を行った。それによって計算時間を減らし、プロパティの類似度が算出可能になり、類似度の質が向上した。提案手法はスケール性が高く、規模の大きなデータセットへの適用も可能である。そして評価実験では等価関係にある組の類似度が上位 10 件以内に入っているのかどうかを評価した。しかしこの評価方法は適切とは言えず、提案した手法の有用性を確かめるに至っていない。また原理的にも実験結果からも、提案手法は事前にノードのマッチングがどれだけできるかで、算出される類似度の質が決まると考えられる。したがってクラス、プロパティ、エンティティ以外のリソースとリテラルのマッチングの追加等を検討しなければならない。最後に現実の LOD への適用については、LOD の大きさに関しては問題無いが、分野が大きく異なる LOD 組への適用は難しいだろうと結論付けた。

謝辞

本研究を進めるにあたり，ゼミでの議論や論文の添削など様々な場面でご指導していただいた杉本重雄先生，永森光晴先生，三原鉄也先生に感謝申し上げます．併せて本研究に関して多くのご意見と助言をくださった坂口哲男先生，森嶋厚行先生に感謝いたします．また杉本・永森研究室の皆様にも多くの助言をいただきました．心より感謝の意を表します．

参考文献

- [1] “The Linked Open Data Cloud,” [オンライン]. Available: <https://lod-cloud.net/>. [アクセス日: 17 12 2018].
- [2] 国立国会図書館, “使う・つなげる : 国立国会図書館の Linked Open Data (LOD) とは , ” [オンライン]. Available: <http://www.ndl.go.jp/jp/dlib/standards/lod/index.html>. [アクセス日: 17 12 2018].
- [3] 田中圭, “既存メタデータスキーマを用いたメタデータインスタンスからのスキーマ推定手法,” 筑波大学 図書館情報メディア研究科, 2014.
- [4] 二十歩亮介, “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.
- [5] T. Berners-Lee, “Plenary at WWW Geneva 94,” 9 1994. [オンライン]. Available: <https://www.w3.org/Talks/WWW94Tim/>. [アクセス日: 19 12 2018].
- [6] T. Berners-Lee, “Semantic Web Road map,” 10 1998. [オンライン]. Available: <https://www.w3.org/DesignIssues/Semantic.html>. [アクセス日: 19 12 2018].
- [7] 神崎正英, セマンティック HTML/XHTML, 毎日コミュニケーションズ, 2009.
- [8] 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,” 28 3 2011. [オンライン]. Available: http://www.soumu.go.jp/main_content/000132512.pdf. [アクセス日: 19 12 2018].
- [9] L. M. Dan Brickley, “FOAF Vocabulary Specification 0.99,” 14 1 2014. [オンライン]. Available: <http://xmlns.com/foaf/spec/>. [アクセス日: 19 12 2018].
- [10] D. C. M. Initiative, “Dublin Core Metadata Element Set, Version 1.1: Reference,” 14 6 2012. [オンライン]. Available: <http://dublincore.org/documents/dces/>. [アクセス日: 19 12 2018].
- [11] R. G. Dan Brickley, “RDF Schema 1.1,” 25 2 2014. [オンライン]. Available: <https://www.w3.org/TR/rdf-schema/>. [アクセス日: 19 12 2018].
- [12] F. v. H. Deborah L. McGuinness, “OWL Web Ontology Language,” 10 2 2004. [オンライン]. Available: <https://www.w3.org/TR/owl-features/>. [アクセス日: 19 12 2018].
- [13] S. B. Alistair Miles, “SKOS Simple Knowledge Organization System,” 18 8 2009. [オンライン]. Available: <https://www.w3.org/TR/skos-reference/>. [アクセス日: 19 12 2018].
- [14] C. B. Tom Heath, Linked Data Evolvind the Web into a Global Data Space, 近代科学社, 2013.

- [15] M. Nilsson, “Description Set Profiles: A constraint language for Dublin Core Application Profiles,” 31 3 2008. [オンライン]. Available: <http://dublincore.org/documents/dc-dsp/>. [アクセス日: 19 12 2018].
- [16] D. C. M. Initiative, “DCMI HOME,” [オンライン]. Available: <http://dublincore.org/>. [アクセス日: 19 12 2019].
- [17] “Digital Public Library of America,” [オンライン]. Available: <https://pro.dp.la/>. [アクセス日: 4 1 2019].
- [18] T. Heath, “Linked Data - Connect Distributed Data across the Web,” [オンライン]. Available: <http://linkeddata.org/>. [アクセス日: 9 1 2019].
- [19] G. Steve Harris, “SPARQL 1.1 Query Language,” 21 3 2013. [オンライン]. Available: <https://www.w3.org/TR/sparql11-query/>. [アクセス日: 19 12 2018].
- [20] T. Berners-Lee, “Linked Data - Design Issues,” 27 07 2006. [オンライン]. Available: <https://www.w3.org/DesignIssues/LinkedData.html>. [アクセス日: 19 12 2018].
- [21] Ontology Engineering Group, “Linked Open Vocabularies (LOV),” [オンライン]. Available: <https://lov.linkeddata.es/dataset/lov>. [アクセス日: 19 12 2018].
- [22] E. F. Palash Goyal, “Graph embedding techniques, applications, and performance: A survey,” Knowledge-Based Systems Volume 151, Pages 78-94 , 2018.
- [23] J. Tang, “LINE : Large-scale Information Network Embedding Categories and Subject Descriptors,” ACM World Wide Web, 1067–1077, 2015.
- [24] P. S. Euzenat, “A Survey of Schema-Based Matching Approaches,” In: Spaccapietra S. (eds) Journal on Data Semantics IV. Lecture Notes in Computer Science, vol 3730. Springer, Berlin, Heidelberg, 2005.
- [25] 市瀬龍太郎, “情報の意味的な統合とオントロジー写像,” 人工知能学会 22 卷 6 号, 2007.
- [26] C. Stadler, “ LinkedGeoData.org, ” [オンライン]. Available: <http://linkedgeodata.org/About>. [アクセス日: 3 1 2019].
- [27] “DBpedia,” [オンライン]. Available: <https://wiki.dbpedia.org/>. [アクセス日: 3 1 2019].
- [28] “GeoNames,” [オンライン]. Available: <https://www.geonames.org/>. [アクセス日: 3 1 2019].
- [29] S. Claus, “ LinkedGeoData: A core for a web of spatial open data,” Semantic Web - On linked spatiotemporal data and geo-ontologies archive Volume 3 Issue 4, Pages 333-354, 2012.

- [30] K. C. Tomas Mikolov, “Efficient Estimation of Word Representations in Vector Space,” Conference: Proceedings of the International Conference on Learning Representations (ICLR 2013), 2013.
- [31] “Ontology Alignment Evaluation Initiative,” [オンライン]. Available: <http://oaei.ontologymatching.org/>. [アクセス日: 4 1 2019].
- [32] R. A.-R. Bryan Perozzi, “DeepWalk: online learning of social representations,” KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining Pages 701-710 , 2014.
- [33] M. S. Cheng Yang, “Fast Network Embedding Enhancement via High Order Proximity Approximation,” Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence Main track. Pages 3894-3900., 2017.

付録

実験で使した各データセット¹²のトリプル例を示す.

1.DarkScape

```
@prefix ns0: <http://dbkwik.webdatacommons.org/ontology/>.
@prefix ns1: <http://dbkwik.webdatacommons.org/darkscape/property/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dc: <http://purl.org/dc/terms/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<http://dbkwik.webdatacommons.org/darkscape/resource/Silver_bar>
  a
    <http://dbkwik.webdatacommons.org/darkscape/class/Recipe>,
    <http://dbkwik.webdatacommons.org/darkscape/class/Item>;
  ns0:abstract "This is one of two normal metal bars which cannot be made into the
standard weapons (e.g. swords, battleaxes, etc.), armour (e.g. shields, platebodies,
helmets etc.) and tools (e.g. pickaxes, hatchets, etc.), the other being the gold
bar. Silver bars however, can be made into silver sickles, which can be wielded as
weapons although they are not very effective in combat."@en;
  ns1:monster
    "Gnoeals"^^rdf:langString, "Skeleton heavy"^^rdf:langString,
    "Small Lizard"^^rdf:langString, "Lizard"^^rdf:langString,
    "Ferocious barbarian spirit"^^rdf:langString, "Flesh Crawler"^^rdf:langString,
    "Baby impling"^^rdf:langString, "Skeleton thug"^^rdf:langString,
    "Icefiend"^^rdf:langString, "Desert Lizard"^^rdf:langString,
    "Skeleton warlord"^^rdf:langString, "Dried zombie"^^rdf:langString,
    "Skeleton"^^rdf:langString ;
  ns1:exchange "GEMW"^^rdf:langString ;
  ns1:store "No"^^rdf:langString ;
```

¹² <http://oaei.ontologymatching.org/2018/knowledgegraph/index.html>

```

ns1:name "Silver bar"^^rdf:langString ;
ns1:combat
    132, 8493, "N/A"^^rdf:langString, 8994110, 5658, 50120, 131718132,2224468410,
    507090, 110, 166, 30;
ns1:level 20 ;
ns1:update "Improved prayer system and more!"^^rdf:langString ;
ns1:skillxp 1.370000e+1 ;
ns1:quantity 2, 3, 1 ;
ns1:stackable "No"^^rdf:langString ;
ns1:high 90 ;
dc:subject
    <http://dbkwik.webdatacommons.org/darkscape/resource/Category:Silver>,
    <http://dbkwik.webdatacommons.org/darkscape/resource/Category:Crafting_items>;
ns1:rarity
    "Rare"^^rdf:langString, "Uncommon"^^rdf:langString, "Common"^^rdf:langString ;
rdfs:label "Silver bar"@en ;
ns0:thumbnail
    <http://darkscape.wikia.com/wiki/Special:FilePath/Silver_bar.png?width=300>;
ns1:tradeable "Yes"^^rdf:langString ;
rdfs:comment "This is one of two normal metal bars which cannot be made into the
standard weapons (e.g. swords, battleaxes, etc.), armour (e.g. shields, platebodies,
helmets etc.) and tools (e.g. pickaxes, hatchets, etc.), the other being the gold
bar. Silver bars however, can be made into silver sickles, which can be wielded as
weapons although they are not very effective in combat."@en ;
ns1:members "No"^^rdf:langString ;
foaf:depiction <http://darkscape.wikia.com/wiki/Special:FilePath/Silver_bar.png> ;
ns1:gevalue "gemw"^^rdf:langString ;
ns1:skill "Smithing"^^rdf:langString ;
ns1:storevalue 45 ;
ns1:equipable "No"^^rdf:langString ;
ns1:alchvalue 90 ;
ns1:weight 1.810000e+0 ;
ns1:mat "Silver ore"^^rdf:langString ;
ns1:examine "It's a bar of silver."^^rdf:langString ;
ns1:release "2001-07-12"^^xsd:date ;
ns1:quest "No"^^rdf:langString ;

```

```
ns1:destroy "Drop"^^rdf:langString ;  
ns1:mat1price "gemw"^^rdf:langString ;  
ns1:low 60.
```

2.Old School RuneScape

```
@prefix ns0: <http://dbkwik.webdatacommons.org/oldschoolrunescape/property/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix ns1: <http://dbkwik.webdatacommons.org/ontology/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<http://dbkwik.webdatacommons.org/oldschoolrunescape/resource/Silver_bar>
  a
    <http://dbkwik.webdatacommons.org/oldschoolrunescape/class/Recipe>,
    <http://dbkwik.webdatacommons.org/oldschoolrunescape/class/Item> ;
  ns0:monster
    "Lizard"^^rdf:langString, "Flesh Crawler"^^rdf:langString,
    "Skeleton thug"^^rdf:langString, "Skeleton warlord"^^rdf:langString,
    "Enraged barbarian spirit"^^rdf:langString, "Skeleton brute"^^rdf:langString,
    "Ferocious barbarian spirit"^^rdf:langString, "Skeleton heavy"^^rdf:langString,
    "Angry barbarian spirit"^^rdf:langString, "Small Lizard"^^rdf:langString,
    "Desert Lizard"^^rdf:langString ;
  ns0:quantity 3, 2, 1 ;
  ns0:seller "Ardougne Silver StallSilver merchant"^^rdf:langString ;
  ns0:level 20 ;
  ns0:combat 132, 42, 166, 24, 12, 283541 ;
  ns0:skillxp 1.370000e+1 ;
  ns0:destroy "Drop"^^rdf:langString ;
  ns0:matlim
<http://dbkwik.webdatacommons.org/oldschoolrunescape/resource/File:Silver_ore.png> ;
  ns0:stackable "No"^^rdf:langString ;
  foaf:depiction
<http://oldschoolrunescape.wikia.com/wiki/Special:FilePath/Silver_bar.png>;
  ns0:quest "No"^^rdf:langString ;
  ns0:equipable "No"^^rdf:langString ;
  ns0:matlqty 1 ;
  ns0:alchvalue 90 ;
```

```

ns0:members "No"^^rdf:langString ;
ns1:thumbnail
<http://oldschoolrunescape.wikia.com/wiki/Special:FilePath/Silver_bar.png?width=300>;
ns1:abstract "The primary use of a silver bar is to create jewellery with the
Crafting skill. To craft silver bars into jewellery players must use a silver bar on
a furnace with a proper mould in their inventory. If players wish to craft jewellery
with gems inset, the cut gems must be present in the player's inventory before
crafting the jewellery. Alternatively, it may be bought from a silver merchant, such
as the one in Ardougne market. Unlike gold bars, which cannot be made into weapons,
silver bars can be made into silver sickles. It can also be ground to silver dust in
the Ectofuntus grinder for use as an ingredient in the guthix balance potion."@en ;
ns0:rarity
    "Uncommon"^^rdf:langString, "Rare"^^rdf:langString, "Common"^^rdf:langString ;
ns0:storevalue 150 ;
ns0:mat "Silver ore"^^rdf:langString ;
ns0:name "Silver bar"^^rdf:langString ;
rdfs:label "Silver bar"@en ;
rdfs:comment "The primary use of a silver bar is to create jewellery with the
Crafting skill. To craft silver bars into jewellery players must use a silver bar on
a furnace with a proper mould in their inventory. If players wish to craft jewellery
with gems inset, the cut gems must be present in the player's inventory before
crafting the jewellery. Alternatively, it may be bought from a silver merchant, such
as the one in Ardougne market."@en ;
ns0:exchange "gemw"^^rdf:langString ;
ns0:skill "Smithing"^^rdf:langString ;
ns0:update "Improved prayer system and more!"^^rdf:langString ;
ns0:examine "It's a bar of silver."^^rdf:langString ;
ns0:weight 1.800000e+0 ;
ns0:release "2001-07-12"^^xsd:date ;
ns0:high 90 ;
ns0:low 60 ;
ns0:store 150 ;
ns0:tradeable "Yes"^^rdf:langString .

```

3.RuneScape

```
@prefix ns0: <http://dbkwik.webdatacommons.org/runescape/property/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ns1: <http://dbkwik.webdatacommons.org/ontology/> .

<http://dbkwik.webdatacommons.org/runescape/resource/Soul_tiara>
  a
    <http://dbkwik.webdatacommons.org/runescape/class/Item>,
    <http://dbkwik.webdatacommons.org/runescape/class/Recipe>,
    <http://dbkwik.webdatacommons.org/runescape/class/Bonuses> ;
  ns0:release "2017-06-05"^^xsd:date ;
  ns0:gevalue "No"^^rdf:langString ;
  ns0:value 100 ;
  ns0:class "None"^^rdf:langString ;
  rdfs:label "Soul tiara"@en ;
  ns0:mat "Tiara"^^rdf:langString, "Soul talisman"^^rdf:langString ;
  rdfs:comment "Soul runecrafting was released on 5 June 2017 as a reward from the
'Phite Club quest. The Soul tiara can be used on the wicked hood, giving the hood two
free teleports to the soul altar, and 5 free soul runes every day. This will consume
the tiara."@en ;
  foaf:depiction <http://runescape.wikia.com/wiki/Special:FilePath/Soul_tiara.png> ;
  ns0:quest "No"^^rdf:langString ;
  ns0:tradeable "Yes"^^rdf:langString ;
  ns1:thumbnail
    <http://runescape.wikia.com/wiki/Special:FilePath/Soul_tiara.png?width=300> ;
  ns1:abstract "Soul runecrafting was released on 5 June 2017 as a reward from the
'Phite Club quest. The Soul tiara can be used on the wicked hood, giving the hood two
free teleports to the soul altar, and 5 free soul runes every day. This will consume
the tiara."@en ;
  ns0:destroy "Drop"^^rdf:langString ;
  ns0:update "Menaphos - The Gates are Open"^^rdf:langString ;
  ns0:members "Yes"^^rdf:langString ;
```

```
ns0:id 5551 ;
ns0:mat2price "gemw"^^rdf:langString ;
ns0:weight 1 ;
ns0:level 1, 90 ;
ns0:accuracy 0 ;
ns0:mat1qty 1 ;
ns0:kept "reclaimable"^^rdf:langString ;
ns0:mat1price "gemw"^^rdf:langString ;
ns0:skill "Runecrafting"^^rdf:langString ;
ns0:disassembly "Yes"^^rdf:langString ;
ns0:equipable "Yes"^^rdf:langString ;
ns0:life 0 ;
ns0:category "magic 6"^^rdf:langString ;
ns0:strength 0 ;
ns0:mat2qty 1 ;
ns0:slot "head"^^rdf:langString ;
ns0:noteable "Yes"^^rdf:langString ;
ns0:stackable "No"^^rdf:langString ;
ns0:armour 0 ;
ns0:ranged 0 ;
ns0:name "Soul tiara"^^rdf:langString ;
ns0:store "No"^^rdf:langString ;
ns0:prayer 0 ;
ns0:damage 0 ;
ns0:requirements "None"^^rdf:langString ;
ns0:exchange "gemw"^^rdf:langString ;
ns0:examine "A tiara infused with the properties of the soul."^^rdf:langString ;
ns0:tier "n/a"^^rdf:langString ;
ns0:tier "n/a"^^rdf:langString ;
ns0:magic 0 ;
ns0:skillxp 55 .
```


4.Memory Alpha

```
@prefix ns0: <http://dbkwik.webdatacommons.org/ontology/> .
@prefix ns1: <http://dbkwik.webdatacommons.org/memory-alpha/property/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://dbkwik.webdatacommons.org/memory-alpha/resource/Kate_Mulgrew>
  ns0:wikiPageExternalLink
    <http://www.totallykate.com/ollady/ollady.html>,
    <http://variety.com/2015/film/awards/sag-award-nominations-2016-nominees-full-
list-1201657169/>,
    <http://tv.ign.com/articles/117/1175324p1.html>,
    <http://www.totallykate.com/blackcom/blackcom.htm>,
    <http://www.playbill.com/news/article/117230.html>,
    <http://www.signaturetheatre.org./press/7_9_07.htm>,
    <http://www.playbill.com/news/article/118970.html>,
    <http://www.totallykate.com/aristocr/aristocr.htm>,
    <http://www.totallykate.com/ourtown/ourtown.htm>,
    <http://www.totallykate.com/tvguides/tvtimes.html>,
    <http://www.totallykate.com/latenews.html>,
    <http://www.totallykate.com/lovespel/lovespel.html>,
    <http://baltimore.broadwayworld.com/article/Kate_Mulgrew_to_Star_in_Master_
Class_at_Paper_Mill_Playhouse_20010101>,
    <http://www.ibdb.com/production.asp?ID=4573>, <http://www.totallykate.com>,
    <http://www.totallykate.com/lajolla/lajolla.htm>,
    <http://www.totallykate.com/othello/othello.htm> ;
  ns1:image "Kathryn Janeway, 2377.jpg"^^rdf:langString ;
  ns1:name "Kate Mulgrew"^^rdf:langString ;
  foaf:depiction
    <http://memory-alpha.wikia.com/wiki/Special:FilePath/Kate_Mulgrew.jpg> ;
  ns0:thumbnail <http://memory-alpha.wikia.com/wiki/Special:FilePath/Kate_
```

```

Mulgrew.jpg?width=300> ;
  dc:subject <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:
Saturn_Award_nominees>,
  <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:Saturn_
Award_winners>,
  <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:Satellite_
Award_winners>,
  <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:Performers>,
  <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:Video_
game_performers>,
  <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:
Film_performers>,
  <http://dbkwik.webdatacommons.org/memory-alpha/resource/Category:VOY_
performers> ;
  ns1:characters "Kathryn Janeway ; Shannon O'Donnel"^^rdf:langString ;
  ns1:gender "Female"^^rdf:langString ;
  ns1:birthName "Katherine Kiernan Maria Mulgrew"^^rdf:langString ;
  ns1:placeOfBirth "Dubuque, Iowa, USA"^^rdf:langString ;
  ns1:dateOfBirth "--04-29"^^xsd:gMonthDay ;
  rdfs:label "Kate Mulgrew"@en .

```

5.Memory Beta

```
@prefix dc: <http://purl.org/dc/terms/> .
@prefix ns0: <http://dbkwik.webdatacommons.org/memory-beta/property/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ns1: <http://dbkwik.webdatacommons.org/datatype/> .
@prefix ns2: <http://dbkwik.webdatacommons.org/ontology/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://dbkwik.webdatacommons.org/memory-beta/resource/Oslo_class>
  dc:subject
    <http://dbkwik.webdatacommons.org/memory-beta/resource/Category:Heavy_escort_
classes>,
    <http://dbkwik.webdatacommons.org/memory-beta/resource/Category:Federation_
starship_classes> ;
  ns0:cruspeed "Warp 7"^^rdf:langString ;
  ns0:affiliation
    <http://dbkwik.webdatacommons.org/memory-beta/resource/United_Federation_
of_Planets>,
    <http://dbkwik.webdatacommons.org/memory-beta/resource/Federation_Starfleet> ;
  ns0:type
    <http://dbkwik.webdatacommons.org/memory-beta/resource/Escort>,
    "fed"^^rdf:langString ;
  rdfs:label "Oslo class"@en ;
  ns0:length "371.0"^^ns1:metre ;
  ns2:wikiPageExternalLink
    <http://web.archive.org/web/20091212070342/http://www.startrekonline.com/ships/
oslo_class> ;
  ns0:width "228.0"^^ns1:metre ;
  ns2:thumbnail <http://memory-beta.wikia.com/wiki/Special:FilePath/USS_Oslo.
jpg?width=300> ;
  ns0:armaments 5 ;
  ns0:maxspeed "Warp 9.75"^^rdf:langString ;
  ns0:mass 625000 ;
  ns0:defences "Deflector shield, rodimium-alloy hull plating"^^rdf:langString ;
```

```
ns0:height "54.0"^^ns1:metre ;  
ns0:className "Oslo-class"^^rdf:langString ;  
foaf:depiction <http://memory-beta.wikia.com/wiki/Special:FilePath/USS_Oslo.jpg> ;  
ns0:servicePeriod 24 ;  
ns0:crew 100 .
```

6. Star Trek

```
@prefix dc: <http://purl.org/dc/terms/> .
@prefix ns0: <http://dbkwik.webdatacommons.org/ontology/> .
@prefix ns1: <http://dbkwik.webdatacommons.org/stexpanded/property/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://dbkwik.webdatacommons.org/stexpanded/resource/USS_Quintillus_(NCC-590)>
  dc:subject
    <http://dbkwik.webdatacommons.org/stexpanded/resource/Category:Federation_
starships>,
    <http://dbkwik.webdatacommons.org/stexpanded/resource/Category:Orion_Press>,
    <http://dbkwik.webdatacommons.org/stexpanded/resource/Category:Hermes_class_
starships>;
  ns0:abstract "The USS Quintillus (NCC-590) was a Hermes-class scout in service in
the mid 23rd century. Ensign Christopher Warren was posted to Quintillus in the 2240s
and rose quickly through the ranks, though he failed to qualify for a heavy cruiser
assignment. (Starfleet Supplemental Communiqué 1:9) Quintillus 's mission commander
in the 2260s was Commodore Richard Broughton. (Starship Farragut: ¥"The Captaincy¥")
Under the command of Captain Christy, Quintillus was assigned to Task Force Four
during the Kelvan War in 2285. (Orion Press: In Harm's Way)"@en ;
  ns1:ncc "NCC-590"^^rdf:langString ;
  rdfs:comment "The USS Quintillus (NCC-590) was a Hermes-class scout in service in
the mid 23rd century. Ensign Christopher Warren was posted to Quintillus in the 2240s
and rose quickly through the ranks, though he failed to qualify for a heavy cruiser
assignment. (Starfleet Supplemental Communiqué 1:9) Quintillus 's mission commander
in the 2260s was Commodore Richard Broughton. (Starship Farragut: ¥"The Captaincy¥")
Under the command of Captain Christy, Quintillus was assigned to Task Force Four
during the Kelvan War in 2285. (Orion Press: In Harm's Way)"@en ;
  ns0:thumbnail
    <http://stexpanded.wikia.com/wiki/Special:FilePath/USS_Quintillus.jpg?
width=300> ;
  foaf:depiction
```

```
<http://stexpanded.wikia.com/wiki/Special:FilePath/USS_Quintillus.jpg> ;  
ns1:name "USS Quintillus"^^rdf:langString ;  
rdfs:label "USS Quintillus (NCC-590)"@en ;  
ns1:class "Hermes"^^rdf:langString ;  
ns1:type "Scout"^^rdf:langString .
```