人工知能学会研究会資料 SIG-SWO-A1001-07

サジェスト機能によるゆるやかなオントロジー構築を可能にするシステムの提案

Proposal of the System for Community-driven Loose Ontology Engineering Using Suggestion

濱崎雅弘 [1,2*]
Masahiro Hamasaki[1,2]

[1] 産業技術総合研究所
[1] Advanced Industrial Science and Technology (AIST)
[2] 科学技術振興機構 CREST
[2] JST, CREST

Abstract: In this paper, we propose a prototype system for community-driven loose ontology engineering using tagging and suggestion. Ontologies are key parts to realize the Semantic Web and CGM (Consumer Generated Media) is one of the most important information resources. So, it is important to support constructing ontologies for not only individuals or small teams but also many participants. Our proposed system based on "property tagging" and "property suggestion" can be used as an ontology development platform, reducing entry barriers for the participations in the creation and maintenance of ontologies.

1 はじめに

近年，RDF 形式で記述したデータを公開する Linked Data が注目を集めている．Tim B. Lee は Linked Data を (1) 名前として URI を用いる，(2) 人々が HTTP で URI にアクセスできる，(3) URI にアクセスすると RDF で記述された有用な情報が提供される[1]，(4) 他の URL が含まれており，さらなることを知ることができる，と定義している [Lee 09]．つまり Linked Data は機械可読な構造化データをアクセス可能にし，さらに，他のデータとリンクさせることが条件となる．これにより，そのデータ単体で，または，データ所有者ではできなかった価値創出が行われる可能性が生じる．特に他のデータとの組み合わせによる新たなサービスの実現（マッシュアップ）が注目されている．オープン化されたデータを Linked Data 化するプロジェクトも進められている．そうして公開された Linked Data は Linking Open Data (LOD) と呼ばれ，その量は 2009 年 11 月時点で 130 億トリプルを数える．

情報資源を Linked Data 化する場合，主な課題は条件 1 に対する「URI を与えるリソースの定義」と条件 3 に対する「トリプルによるデータの記述」となる．これは情報資源が対象とするドメインのオントロジーを生成し，それに基づいてリソースを記述することといえる．なお，条件 4 の「他のデータとのリンクの生成」も重要な課題であるが，程度問題が含まれてくるのでここでは割愛する．[2]

本研究の目標は，情報資源として注目される CGM(Consumer Generated Media) に適した Linked Data 生成環境の構築である．CGM の魅力の一つは，データの構造さえもユーザが生成していくことにより，既存の情報資源ではカバーできなかった広範で曖昧な知識をデータ化できた点にある．そして CGM の基本は，知識や能力を持つユーザが自身の動機に基づいて自身ができることをやり，それが共有され蓄積されていくことで大きな価値を生み出すというものである．よって知識を

*連絡先：産業技術総合研究所情報技術研究部門
〒 101-0021 東京都千代田区外神田 1-18-13 秋葉原ダイビル 11F
E-mail: hamasaki@ni.aist.go.jp

[1]現在は書式を RDF ではなく「RDF*または SPARQL」に変更している

[2]著名な Linked Data の一つである DBPedia [Bizer 09] の URI は「dbpedia.org/resource/用語」となっているため，文字列マッチングだけでテキストコーパスから DBPedia へのリンク作成は容易にできる．しかし語の揺れや同音異義語も含めて解決するとなると困難になる．また，DBPedia のような巨大なデータになると，内部で十分なリンクがあれば外部へのリンクがなくとも「さらなることを知ること」ができる．このように条件 4 に関しては，どの程度でもってクリアしたと判断するかが難しいため，課題には含めないでおく．

持っている本人が共有のためのデータ化を行う必要がある．このことから，(1) データ作成は負荷が低く誰でもできること，(2) ユーザが構造を作れること，が望ましいと考えられる．

知識を持っている本人にとっても，その知識の背景にある構造がどのようなものであるかを理解するのは容易ではない．オントロジーどころか分類や階層といったタキソノミーですら困難である．ではまったく構造を持っていないかというと，そうではないこと，ほとんどの人々が現に情報を何かしらの指針のもとに整理していることからもわかる．それぞれの個人が情報を整理するための弱い構造を持っており，かつ，それらは外在化さらには共有化されていない状態にあると考えられる．

では，どのような方法であれば，その弱い構造を外在化し，さらには共有化できるだろうか．ソーシャルブックマークの普及から，情報から想起される情報（タグやコメント）ならば多くのユーザができることがわかる．ここで対象となる情報とタグ・コメントとの関係性を記述すれば，リソース・属性名・属性値の関係になる．これはユーザにとって簡単で，かつ，データが構造化されるという点で，先ほどの二つの要件を満たす方法である．本研究ではこれを**属性タギング**と呼ぶ．しかし属性タギングだけでは語彙が発散して構造化が進まない可能性がある．そこですでに入力されたデータを元に属性をサジェストすることで，ユーザの負担を減らしつつ，データ全体の構造化を促す．本研究ではこれを**属性サジェスト**と呼ぶ．これら属性タギングと属性サジェストにより，ユーザはオントロジーのような知識の全体構造を意識せずにデータ（知識）の入力が可能になると考えられる．

本論文では CGM 的な Linked Data 生成手法としての属性タギングと属性サジェスト，およびそれらを実現するシステム・Social Infobox の提案を行う．既存の情報資源を用いて提案手法の有効性を示し，プロトタイプシステムを用いたユーザスタディから提案手法の可能性について考察を行う．

本論文の構成は以下の通りである．まず次章にて知識構築手段における関連研究を述べる．3 章にて提案手法の概要を述べ，4 章にてその詳細を述べる．5 章にて提案手法の実装に必要な計算モデルについて述べる．6 章にて既存の情報資源を用いた調査について述べ，7 章にてプロトタイプシステムを用いた予備実験の結果について述べる，最後に 8 章にて本論文をまとめる．

2 関連研究

2.1 オントロジー構築

オントロジーとは対象ドメインに現れる概念と概念間の関係とを明示的に示し，意味定義を与えたものである [溝口 97]．既存の知識表現は全体性，網羅性，体系性に対して断片的な配慮しかなされておらず，全てをカバーしていなかった．オントロジーはこれら 3 点全てを配慮することに加えて，特に概念間の関係性の詳細化（構造化）に重きを置く点が異なる．武田はこれらの条件に加え，形式的に記述されており推論が可能（形式性），目的の明示化により合意が可能（合意可能性）をオントロジーの特徴として挙げている [武田 01]．

オントロジー構築のための開発論は，数多く提案されている．オントロジー構築の一般的な手順は，対象となる概念を洗い出し，それらを体系化しクラスおよびプロパティ（属性）を定義するというものである．Protege を用いた [Noy 01] らのオントロジー開発手順を挙げると，(1) スコープ決定，(2) 再利用検討，(3) 用語列挙，(4) クラス定義，(5) プロパティ定義，(6) 制約定義，(7) プロパティの制約を決める，(8) インスタンス作成，となっている．これらの定義は対象ドメインの専門家とオントロジー設計者によって行われることを想定している．しかしながら CGM には特定のエキスパートによる構造の設計という方法はそぐわない．これは不特定多数の種々のエキスパートによる構造の創発こそが CGM の要件だからである．

CGM 的なアプローチでオントロジー構築をしようという試みもある．OntoWiki は Wiki のように誰でも自由に編集できるオントロジーエディタである [Hepp 06]．提案手法と考え方としては近いが，提案手法は特にユーザが簡単にデータを入力し，厳密な構造化を目的としていない点が異なる．

提案手法はユーザが自由に関係とリソースを追加していくという点で，マインドマップ [Buzan 05] やダブルイメージマップに似ている．これら発想支援手法とは，いかに人の知識を引き出すかという点では軌を一にするため，類似手法が存在することは十分に考えられる．しかし，これら発想支援手法は主に思考を展開させていくことを目的としているのに対し，提案手法は知識の構造化のために過剰な発散を防ぐことを目的としている点が異なる．

2.2 既存の情報資源からのオントロジー構築

既存の情報資源を Linked Data 化する場合，その情報資源の構造がどのようになっているかでアプローチが異なる．情報資源が明確に構造化されたデータベースであるならば，データベーススキーマとオントロジー

とのマッピングが主な課題となる．その際にはオントロジーマッチング [市瀬 07] やオントロジー検索 [Ding 04] などが重要な技術となる．逆に，まったく構造化されていないフリーテキストからであるならば，情報抽出の問題となる．これについては NLP やウェブマイニングの分野で多くの研究がなされている．

CGM として代表的なものに Wikipedia や Social Bookmark が挙げられるが，これらは弱い構造をもった情報資源であるといえる．この場合，構造化されている部分だけを利用したり，複数の情報資源を組み合わせたりしながら，前述の技術を組み合わせているケースが見られる．DBPedia では Wikipedia の Infobox を利用している [Bizer 09] ．YAGO は Wikipedia のカテゴリに加えて WordNet を利用して高い精度を出している [Suchanek 08] ．Wu らは Wikipedia の Infobox と WordNet を利用したアプローチを提案している [Wu 08]. また，Social Bookmark のユーザ・タグ・URL という三つ組から構造化するアプローチも提案されている [Mika 07] [Specia 07].

3 ゆるやかなオントロジー構築

CGM 的にクラスおよび属性を生成するにはどうすればよいだろうか．ここでクラスそして属性とは何かを再考する．属性とはリソースを分類・特徴付けするために人が付与するものである．例えば小沢一郎のタグとして思いつくのは，政治家，民主党，幹事長などである．これを属性名＝属性値に変換すると，職業＝政治家，政党＝民主党，役職＝幹事長となる．このとき，血液型や利き腕などは思いつかない．これは小沢一郎を他のリソースと比較する上で特徴として不要だからと考えられる．一方で野球選手ならば「利き腕」は必要な特徴となる．

ここから導かれるのは，暗に比較対象となりうる（比較する機会がある，比較することに価値がある）リソース集合の存在である．これがクラスの正体であると考えると，クラスとは共通する属性名を持ち，その属性値によって効率的に分類できるリソースの集合を指すものといえる．つまり，コミュニティのメンバーが共有したい情報を説明するために必要と思われる属性名（と属性値）を追加していけばそのコミュニティにとって自然なクラスが浮かび上がってくるのではないかと考えられる．現にオントロジーにおける概念はスロットの集合として表現されており，このようなアプローチによって概念定義を生み出すことは可能である．

しかしながら属性を部分的に共有するリソース集合があったとして，その集合全体に共通する属性的特徴があることは保証されない．つまり，このようなアプローチでは明確なクラス定義が生成されることは保証されず，むしろデータが増える（クラスに属するリソースが増える）ほど生成されることはまれになると考えられる．Wittgenstein は著書「哲学的探求」において，あるカテゴリに属するインスタンスは部分的な類似性（家族的類似性）をもとにしており，全てに共通する特徴は存在しないと述べた．この考えに基づくと，明確なクラス定義の存在は知の構造化にとって必要条件ではないといえる．

ここで情報共有という点から考えると以下の 2 点が導き出せる．

- 知の構造化には，リソースや属性値よりもむしろ属性名の共有を促すことが重要である
- 属性名の共起関係から（クラス的な）類似関係を求めることができる

以上の議論を踏まえて，本研究では情報構造化手法として属性タギングと属性サジェストを提案する．この二つの手法により，ユーザはクラス定義のような知識の全体構造を意識せずに知識（構造化データ）の入力が可能になる．また，属性サジェストを通して弱い構造化が漸進的に行われることで，ゆるやかなオントロジー構築が可能となる．

- **属性タギング**：属性名と属性値のペアで入力する．これにより入力した時点でトリプルができる．また，入力データの単位が小さくなり，入力が容易になる．なお，ユーザは必要な属性を追加することができる．これにより自分なりの整理の仕方で情報の入力が可能となる．
- **属性サジェスト**：入力済みデータを用いた属性名・属性値のサジェストをする．追加された構造を共有することで全体の構造化を進める．サジェスト機能によりユーザの負担を減らす．

4 提案システム

4.1 システム概要

提案システムは Wiki のように誰でもデータ入力および編集が可能となっている．データはリソース名，属性名，属性値の三つ組みで構成される．リソースは複数の属性名と属性値のペアを持ち，1 のように表記できる．ユーザはリソース名，属性名，属性値，いずれも自由に入力することができる．一つのリソースには複数の属性を持つことができ，Social Bookmark におけるタグのように属性情報を付与する（属性タギング）．このような形式であるため，容易に RDF 化が可能である．

（リソース名）

アーノルド・シュワルツネッガー	
誕生日	1940/11/27
職業	政治家
所属政党	共和党
主演映画	ターミネーター (1973)
（属性名）	（属性値）

図 1: データの例.

しかし自由に属性情報を追加していくだけでは，全体の構造化が進まない．そこですでに入力されたデータを元に，属性名および属性値がサジェストされる（属性サジェスト）．ユーザの負荷を減らすと共に，全体の構造化を促進するのが狙いである．

属性サジェストはユーザインタフェース的には直接提示と比較提示の 2 種類が存在する．直接提示によるサジェストは，リソース情報に基づく推薦度により追加すべき属性情報を提示する．比較提示によるサジェストは，他のリソースとの比較表示により追加すべき属性情報を提示する．

編集中のリソース

アーノルド・シュワルツネッガー	
誕生日	1940/11/27
職業	政治家
所属政党	共和党
主演映画	ターミネーター (1973)

サジェストされる属性名

性別, 生誕地, 血液型, 国籍, 両親,
役職, 政治思想, デビュー作品, 代表作, etc.

図 2: 属性名のサジェストの例.

編集中のリソース

アーノルド・シュワルツネッガー	
誕生日	1940/11/27
職業	政治家
所属政党	共和党
主演映画	ターミネーター (1973)
血液型	???

サジェストされる属性値

A 型, B 型, AB 型, O 型, A 型 RH+, etc.

図 3: 属性値のサジェスト

4.2 候補提示によるサジェスト

属性情報（属性名と属性値）が入力候補としてサジェストされる．2 は，属性名がサジェストされている例である．1 のリソースに対して，すでに持っている属性情報を元に，属性名がサジェストされる．例えば誕生日という属性を持つならば，性別や生誕地といった属性を持つ可能性が高い．同様に，職業があれば役職が，所属政党があれば政治思想が，主演映画があればデビュー作品があると考えられる．ユーザはこのようにして提示された属性名を見て，自分が知っている情報を入力すればよい．穴埋め問題となるので，ユーザにとって知識入力の負荷は自由文と比べて小さくなる．また，穴埋め問題のような不完全情報を与えると，人は補完したくなるという心理学的知見（ツァルガイック効果）もある．そういった面からも提案手法は知識外化手段として有効であると考えられる．サジェストされた属性名に意図するものが無ければ，自由に追加することができる．

属性が選択されると，次に入力すべき属性値がサジェストされる．3 は，選択された血液型という属性に対する属性値がサジェストされている．血液型の場合，属性値となりうるデータは限定されるので，それらが列挙されれば，ユーザは選択肢を選ぶだけで属性値の入力が可能になる．両親やデビュー作品といった属性では，取り得る属性値の幅が広く，サジェストされた属性値に目的の属性値が含まれる可能性は低い．しかし，提案手法はユーザが知っている知識を外化するのが狙いであるため，必ずしもユーザが意図する属性値が出る必要はない．それよりもむしろ，他では

4.3 比較提示によるサジェスト

3 章にて，属性情報の根底にはリソースの比較があると述べた．この考えに基づき，他のリソースとの比較表示により追加すべき属性情報を提示する．図 5 は比較提示によるサジェストの例である．属性情報（属性名と属性値）が比較情報として表示される．ユーザはこれを見て属性値が欠落した属性情報を容易に発見でき，また，比較情報が表示されることで，どういった属性値をいれるべきかがわかる．

比較したリソース同士が同じ属性を持つとは限らないため，全ての属性情報を埋められるわけではない．例えばロナルド・レーガンは米国大統領であるため「対日政策」という属性を持ちうるが，同じ政治家であっても州知事であるアーノルド・シュワルツネッガーには不要である．属性情報が増えてくると，比較提示において不要な属性が多くなってくる可能性がある．これについては，より一般的であったり必要と思われる属性名を上位に表示する，属性ランキングによって対応可能であると考える．

比較表示中のリソース

	アードルド・シュワルツネッガー	ロナルド・レーガン
フルネーム	—	ロナルド・ウィルソン・レーガン
国籍	—	米国
誕生日	1940/11/27	1911/02/06
職業	政治家	政治家
所属政党	共和党	共和党
代表作	ターミネーター (1973)	—
麻薬政策	—	「麻薬との戦争 (War on Drugs)」政策を推進.
対日政策	—	安全保障上のパートナーとして友好関係を結ぶ.

サジェストされる属性値

ロナルド・レーガン, バラク・オバマ, シルヴェスター・スタローン, etc.

サジェストされるリソース

ロナルド・レーガン, バラク・オバマ, シルヴェスター・スタローン, etc.

図 5: 比較提示によるサジェスト

編集中のリソース

アーノルド・シュワルツネッガー	
誕生日	1940/11/27
職業	政治家
所属政党	共和党
主演映画	ターミネーター (1973)

サジェストされるリソース

ロナルド・レーガン, バラク・オバマ, シルヴェスター・スタローン, etc.

図 4: リソースのサジェスト

5 アルゴリズム

本章では属性サジェストのためのアルゴリズムについて述べる．候補提示によるサジェストには，リソースに対し追加すべき属性名を求める属性名推薦と，属性に対し追加すべき属性値を求める属性値推薦の 2 種類がある．比較提示によるサジェストには，比較提示によるサジェストには，比較すべきリソースを求めるリソース推薦がある．いずれも「クラスとは属性名を共有するリソース集合であり，その属性値によって効率的に分類できるリソースの集合を指すもの」という仮説に基づいて上記を計算する．

5.1 リソース推薦

これまでの議論から，同一クラスに属するリソースは類似した属性集合を持つと考えられる．このことから，あるリソース n_a が与えられたとき，すでに持っている属性 $\{a_1, a_2, \ldots, a_m\}$ から，同一クラスに属するリソース n_b が推定できると考えられる．そこで属性集合 $\{a_1, a_2, \cdots, a_m\}$ を持つリソース n_a が，属性集合 $\{b_1, b_2, \ldots, b_l\}$ を持つリソース n_b と同じクラスに属する確率 $p(n_b|n_a)$ を求めて推薦リソースを選択する．

問題を簡略化するために $\{a_1, a_2, \cdots, a_m\}$ および $\{b_1, b_2, \cdots, b_m\}$ はそれぞれ互いに独立と考える．ベイズの定理により

$$\begin{aligned} p(n_b|n_a) &= \Pi_i p(b_i|n_a) \\ &= \Pi_i \frac{p(a_1, a_2, .., a_m|b_i)p(b_i)}{p(a_1, a_2, .., a_m)} \\ &\propto C + \Sigma_i(logp(b_i) + \Sigma_j \log p(a_j|b_i)) \end{aligned}$$

なお，全リソースを W，属性 x を持つリソース集合を X，属性 y を持つリソース集合を Y とすると，$p(x) = \frac{|X|}{|W|}$ および $p(y|x) = \frac{|Y \cap X|}{|X|}$ である．

5.2 属性名推薦

これまでの議論から，同一クラスに属するリソースは類似した属性集合を持つと考えられる．つまり，属性間には共起パターンが存在すると考えられる．このことから，あるリソース n_a が与えられたとき，すでに持っている属性 $\{a_1, a_2, \ldots, a_m\}$ から，次に追加すべき属性 b_x が推定できると考えられる．そこで属性集合 $\{a_1, a_2, \cdots, a_m\}$ を持つリソース n_a が，属性 b_x を持つ確率 $p(b_x|n_a)$ を求めて推薦する属性名を選択する．

問題を簡略化するために $\{a_1, a_2, \cdots, a_m\}$ は互いに独立であると考える．ベイズの定理により，

$$p(b_x|n_a) = \frac{p(n_a|b_x)p(b_x)}{p(n_a)}$$

$$
\begin{aligned}
&= \frac{p(a_1, a_2, .., a_m|b_x)p(b_x)}{p(a_1, a_2, .., a_m)} \\
&\propto C + \log p(b_x) + \Sigma_i \log p(a_i|b_x))
\end{aligned}
$$

なお，全リソースを W，属性 x を持つリソース集合を X，属性 y を持つリソース集合を Y とすると，$p(x) = \frac{|X|}{|W|}$ および $p(y|x) = \frac{|Y \cap X|}{|X|}$ である．

5.3 属性値推薦

属性には 2 種類あると考えられるリソース n_1 の特徴を示す属性 a_1 と，リソース n_2 が属するクラスを示すメタ属性 a_2 である．これまでの議論から，属性 a_1 は同一クラスのリソースが持つ属性値との差異が重要であると考えられる．一方で属性 a_2 は同一クラスが持つ属性値と同一である考えられる．いずれの場合においても，同一クラスのリソースが持つ属性値を示すことがユーザにとって有用であるといえる．よって，同一クラスのリソースが属性名 a に対して属性値 x を持つ確率の期待値を求めることで推薦すべき属性値 x を選択する

リソース m が属性 a に対して属性値 x を持っていたら 1, そうでなければ 0 を取る関数 $f(m, a, x)$ を定義し，全リソース数を N としたとき，属性 a に対する属性値 x の期待値 $E(x)$ は以下で求められる

$$
E(x) = \frac{1}{N}\Sigma(p(m|n)f(m, a, x))
$$

6 既存システムの調査

提案手法では，データをリソース名，属性名，属性値の三つ組でユーザに入力してもらう．代表的な CGM である Wikipedia には Infobox と呼ばれるデータがあり，これは表形式で属性名と属性値を入力するものである．自由に記述できるが，属性名のリストが「政治家」「俳優」といったテンプレートとして用意されている．これはプロパティベースの提案手法と異なり，クラスベースの CGM 的なオントロジー構築といえる．

Wikipedia の Infobox がオントロジー構築にとって有用であることは関連研究にて示されているが，ここではクラスベースのオントロジー構築がどのように行われているかを調べる．分析データには 2009 年の日本語版 Wikipedia ダンプデータを用いた．

10 リソース以上に適用されているテンプレートは 488 個，これらのテンプレートを少なくとも一つ以上持つリソースが 221,230 リソース，これらのテンプレートに含まれるユニーク属性名は 148,22 であった．798%のリソースが 1 つのテンプレートを用いており，複数のテンプレートを組み合わせて利用するケースは多くないことがわかる．

リソース，属性名，属性値のトリプルを 1 レコードと数えた場合，全部で 4,524,537 レコードであり，そのうち 1,244,444 レコードが属性値が空白であった．属性を持つリソースの数は 222,058 であり，空白属性を一つ以上持つリソースの数は 151,285 であった．つまり約 68%のリソース少なくとも一つ以上の空っぽの属性を持つリソースを持つ属性名の数は 16,909 であり，空白属性を一つ以上持つ属性名の数は 7,311 であった．つまり約半数の属性が属性値が空白のリソースを持つ．データを見ると，属性値を持つリソースが一つしかない属性名が 3,206，2 つしかない属性名を合わせると 4,277 で，全属性名の約 30%を占める．また，一つのテンプレートにしか含まれていない属性名は 11,555 と全体の約 78%を占めている．このことから，クラスベースではあるが，クラス（テンプレート）に基づいて入力される属性は一部であり，あとは各インスタンスに必要な属性が多く追加されているという，プロパティベースなアプローチがとられていることが伺える．

7 予備実験

本章では，提案手法を実装したプロトタイプシステムを用いたユーザスタディについて述べる．プロトタイプシステムの利用は開発メンバーを含む一部のユーザに限定されており，まだ十分に検証された知見は得られていない．しかしながらシステムの実装および利用を通して興味深い考察が得られたので，これらを本章にて述べる．

7.1 プロトタイプシステム

ウェブアプリケーションとしてプロトタイプシステムを構築した．図 6 はシステムのスクリーンショットである．一つのリソースに対して一つのページが割り当てられる．図 6 は「ナイーブベイズ」のページである．左中央に「ナイーブベイズ」の属性情報が表形式で表示されている．右中央は比較リソースとして「ID3」が表示されている．左下にはサジェストされた属性名の，右下にはサジェストされたリソースの一覧が表示されている．

属性名をクリックするか，新規属性名を入力すると，属性値の入力ができる．属性値にはテキストを入力するが，リソースかリテラルかを選択することができる．リソースを選択した場合，入力された属性値を名前としたリソースが自動生成される．なお，リソース A の属性 B の属性値としてリソース C が入力された場合，「リソース A，属性 B，リソース C」という三つ組だけ

図 6: プロトタイプシステム

でなく,「リソース C，属性 B’，リソース A」という三つ組も生成する．この，向きを逆にした三つ組は，画面中には表示されないが，サジェストを計算する際に用いられる．

入力された属性値は，リソースであるならば該当リソースのページへのハイパーリンク付きで表示される．リテラルである場合も，文中にリソース名が出現していた場合は自動的にハイパーリンクを追加する．

7.2 予備実験結果

システムを用いて生成されたデータをオントロジーという観点から見てみる．なお，以下ではクラス定義のようなものが生成されたと述べているが，明確なクラス定義が生成されたわけではなく，ゆるい属性集合が生成されただけである．

まず，共通する属性集合を多く持つリソース群が生成された．例えば属性「所属」「専門」などを持つリソース「武田英明」「濱崎雅弘」「大向一輝」である．これらのリソースに共有されている属性集合は，内包的なアプローチで生成されたクラス定義ともいえる．

また，特定の属性名の属性値として現れるリソース群が生成された．例えば属性「国際会議」の属性値に現れるリソース「WWW」「IJCAI」「AAAI」である．これらのリソースに共有されている属性集合は，外延的なアプローチで生成されたクラス定義，そして属性名はそのクラス名といえる．

他には，リソース群の分類において情報量の高い，つまり属性値の種類が比較的少ない属性が生成された．例えば属性「サービスの種類」である．これの属性値には「ソーシャルネットワーキングサイト」や「動画共有サイト」などがある．この属性において同一の属性値を持つリソースに共有されている属性集合は，外延的なアプローチで生成されたクラス定義，そして属性値はそのクラス名といえる．

8 まとめ

本研究では，属性サジェスト機能によりゆるやかなオントロジー構築を可能にするシステムを提案した．属性サジェストは UI 的には直接提示と比較提示との 2 種類があり，提示候補の計算的には属性名推薦，属性値推薦，リソース推薦の 3 種類のアルゴリズムによって構成されている．本稿では提案手法の概要を述べ，プロトタイプシステムの紹介と予備実験結果について報告した．

情報共有は長く議論されている課題であるが，情報処理技術の普及と共にその対象を広げており，多様な技術が求められている．提案手法は明確な知識構造化を牽引する設計者を必要とせず，知識を持つメンバーそれぞれが自身の考えに基づいて情報構造化を進めていくことで結果的に全体構造化を促す仕組みを有しており，CGM に適した Linked Data 生成環境の基盤技術となりうると考える．

今後は，より多くのデータ・参加者においても提案手法が有効に機能するかどうか実証実験を通して確かめたい．

参考文献

[Bizer 09] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S.: DBpedia - A Crystallization Point for the Web of Data, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, pp. 154?–165 (2009)

[Buzan 05] Buzan, T. and Buzan, B.: ザ・マインドマップ, ダイヤモンド社 (2005)

[Ding 04] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J.: Swoogle: a search and metadata engine for the semantic web, in *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pp. 652–659, ACM (2004)

[Hepp 06] Hepp, M., Bachlechner, D., and Siorpaes, K.: OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis,

in *Proceedings of the 2006 international symposium on Wikis* (2006)

[Lee 09] Lee, T. B.: Linked Data (2009), http://www.w3.org/DesignIssues/LinkedData.html

[Mika 07] Mika, P.: Ontologies are us: A unified model of social networks and semantics, *Web Semant.*, Vol. 5, No. 1, pp. 5–15 (2007)

[Noy 01] Noy, N. F. and McGuinness, D. L.: Ontology Development 101: A Guide to Creating Your First Ontology, in *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05* (2001)

[Specia 07] Specia, L. and Motta, E.: Integrating Folksonomies with the Semantic Web, in *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*, pp. 624–639, Springer-Verlag (2007)

[Suchanek 08] Suchanek, F. M., Kasneci, G., and Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet, *Web Semant.*, Vol. 6, No. 3, pp. 203–217 (2008)

[Wu 08] Wu, : Wikipedia Infobox + WordNet, in *hoge*, pp. 0–1 (2008)

[溝口 97] 溝口 理一郎, 池田 満：オントロジー工学序説：内容指向研究の基盤技術と理論の確立を目指して, 人工知能学会誌, Vol. 12, No. 4, pp. 559–569 (1997)

[市瀬 07] 市瀬 龍太郎：情報の意味的な統合とオントロジー写像, 人工知能学会誌, Vol. 22, No. 6, pp. 818–825 (2007)

[武田 01] 武田 英明：人工知能におけるオントロジーとその応用, 情報知識学会研究報告会講演論文集, No. 9, pp. 1–12 (2001)