テキストを含む構造化データに対する知識ベースを用いたOLAP分析

中野茉里香† 天笠 俊之†† 北川 博之††

† 筑波大学 情報学群情報科学類 〒305–8573 茨城県つくば市天王台 1–1–1
†† 筑波大学 計算科学研究センター 〒305–8573 茨城県つくば市天王台 1–1–1
E-mail: †marikan@kde.cs.tsukuba.ac.jp, ††{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし 近年，オンラインメディアの普及によりテキストを含む構造化データが増加している．データベースに蓄積された大量のデータから多次元的な集計・分析を行う手法としては，OLAP(Online Analytical Processing) 分析が挙げられるが，テキストに対しては扱える語彙やトピックに限りがあり，また語義の曖昧性や表記揺れなどの問題に対応できないといった課題があった．一方で，事物に関する一般的な知識を構造化データとして表現し，機械処理に利用可能な知識ベースが注目され，様々な分野で利用されている．そこで本研究では，構造化データと知識ベースとをリンクさせることによって，より高度なOLAP分析を行う手法を提案する．テキスト中の単語を知識ベースのエンティティに対応させることで，語義の曖昧性や表記揺れの問題の解消を行うことができる．また，知識ベースにおける概念階層を利用することで，事前に人手による辞書の構築を行う必要なく分析を行うことが可能となる．ニュースコーパスを用いた実験によって，従来手法と比較して提案手法が低コストな前処理で，より高精度なテキストのOLAP分析を実現していることを示した．

キーワード OLAP分析, 知識ベース, テキスト処理, RDF

1 はじめに

近年，オンラインニュースメディアやECサイトの普及によって，構造化データとともにテキストを持つデータが多く蓄積されている．一般的に，時間情報や位置情報，製品情報といったデータは数値等を用いて形式的に表現されるため，構造化データと呼ばれる．これに対してテキストデータは，構造的に表現することのできない，非構造化データに当たる．

これまでに，構造化データのみに対する分析は多く行われてきている．このような分析を効率的に行う技術としてはOLAP(Online Analytical Processing) 分析 [1] が存在する．OLAP分析はデータベースに蓄積された大量のデータから多次元的な情報の集約を行う手法であり，ユーザは対話的な分析を行うことができる．OLAP分析は分析の対象となるデータであるメジャーと分析の軸となる次元から多次元モデルを構築することで，大規模なデータに対しても素早く結果を返すことを可能とする．

しかし，OLAP分析は数値データのような構造化されたデータに特化した技術であるため，従来のOLAP分析システムではテキストに含まれる情報についての分析は対象としていなかった．レビューデータのようなデータにおいては，構造化されたデータだけでなく，テキストにも多くの有益な情報が含まれている．これらの情報も扱うことで，より詳細な分析を行うことが可能になると考えられる．

そこで，OLAP分析を構造化されたデータだけでなくテキストデータにも適応させる研究が行われてきた．これらの研究では，テキストから得られた単語やトピックを新たに次元として用いることで，テキストから得られた情報についても利用した，よりリッチな分析を可能にしている．しかし，これまでのテキストに対するOLAP分析の研究では，テキストに含まれる単語を単に文字列として捉えているため，多義語についての曖昧性や同一の概念を表す単語の表記揺れ等の問題に対処することができなかった．また，これらの研究では，テキストから抽出する単語や単語の持つ階層構造を事前に定義しておく必要があるため，分析を行うには分野ごとに人手による前処理を行う必要があった．

一方で，事物に関する一般的な知識を構造化データとして表現し，機械処理に利用可能な知識ベースが発達を遂げている．知識ベースは，様々な分野の単語と単語の意味的な階層構造を内包する．大規模知識ベースの代表例としては，DBPedia[1]やWikiData[2]等が挙げられ，大きな注目を集めている．これらの知識ベースはあらゆる分野についてのデータが体系的に記述されている上，誰もが利用可能であるように公開されている．また，既存のテキスト中の単語をこれらの知識ベース中のエンティティと結びつける技術であるエンティティリンキングが近年注目され，これを活用した様々な研究が進められている．[2]

そこで，本研究では知識ベースに対するエンティティリンキングによってテキストに含まれるエンティティを捉えることで，テキストに対するOLAP分析を可能にする手法を提案する．エンティティリンキングでは，テキストからエンティティ名を抽出するのみでなく，文脈を考慮したエンティティの抽出を行う．これにより，語義の曖昧性の解消や表記揺れの解消を行うことができるため，従来のテキストのOLAP分析システムでは成し得なかったより正確な分析が可能となる．また，テキス

1 : http://wikidata.dbpedia.org/
2 : https://www.wikidata.org/wiki/Wikidata:Main_Page

トから抽出したエンティティについて知識ベースに含まれる概念階層のようなエンティティの持つ付加的な情報を用いることで，ユーザが事前に単語や単語の階層構造の定義を行う必要なく，分析を行うことが可能となる．

評価実験により，従来手法が分析を行うための前処理に膨大な時間とストレージコストを必要とする一方で，提案手法はより低いコストで分析を行うことが可能であると示された．また，従来手法の単語ベースと提案手法のエンティティベースのそれぞれの方法で抽出された単語またはエンティティを比較した実験によって，提案手法が表記揺れや語義の曖昧性の解消を行い，より正確な分析を可能としていることが示された．

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

2 前提知識

2.1 OLAP (Online Analytical Processing)

OLAP(Online Analytical Processing) 分析 [1] とは，データベースに蓄積された大量のデータから多次元的な集計・分析を行う手法である．様々な粒度についての集約が可能であり，ユーザは大規模なデータに対して対話的な分析を行うことができる．

OLAP 分析は収集したデータから多次元データモデルであるキューブを生成する．多次元データモデルとは「製品」「期間」「地域」といった複数の分析の軸を持つデータモデルである．

キューブを構築するために，一つの事実表と複数の次元表からなるスタースキーマを用いる．事実表には分析対象の数値データであるメジャーと次元表に関連づけられるキーが格納され，次元表には階層構造を持つ属性が格納されている．スタースキーマの例を図 1 に示す．このスタースキーマは，事実表である sales と次元表である dates，product，store から構成されている．事実表 sales は，売り上げの情報を示すスキーマであり，メジャーとなる sales と共に dates，product，store への外部キーを持つ．また，次元表 store は主キーである store_key と共に city と nation を持つ．city と nation からなる store 次元の階層構造の例を図 2 に示す．city のそれぞれの値は nation のいずれかの値に属することとなる．

スタースキーマから構築されたキューブは，分析対象となる数値を格納したメジャーとそれに関連付けられた複数の分析の軸で構成される．図 1 のスキーマを用いて構築されたキューブが図 3 の左のキューブとなる．このキューブは dates，product，store の三つの軸からなり，各セルにはメジャーである sales が格納されている．

このキューブに対して，ロールアップ，ドリルダウンなどの操作を適用することで，様々な分析が可能となる．ロールアップとはある次元についてより荒い粒度で集約を行う操作であり，ドリルダウンとはある次元についてより詳細な粒度の集約を行う操作である．ロールアップとドリルダウンの例を図 3 に示す．左のキューブでは store の次元の粒度が city であるが，store の階層構造を用いてロールアップの操作を行うことによって，store の次元の粒度が nation である右のキューブに集約することができる．

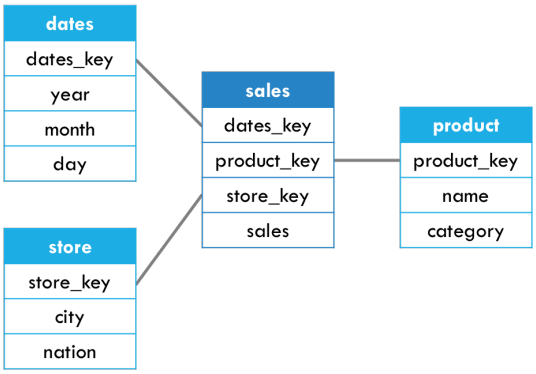

図 1　スタースキーマ

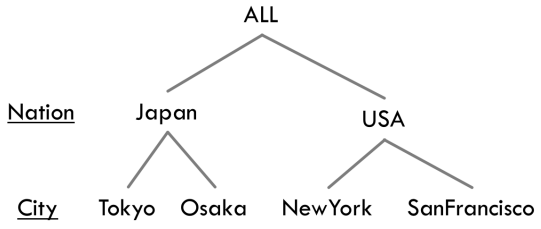

図 2　store の階層構造

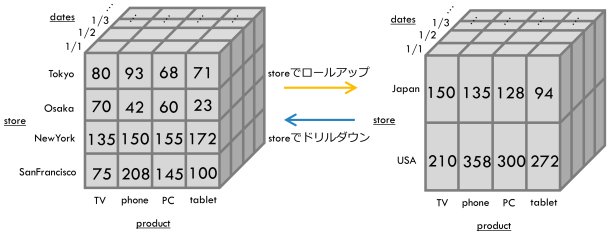

図 3　ロールアップとドリルダウン

2.2 知識ベースと RDF

2.2.1 知識ベース

知識ベースとは様々な分野の知識を組織化し，蓄積したデータベースである．大規模な知識ベースの代表例としては Wikipedia 等がある．知識ベースの中でも特に，構造化され，コンピュータによる読み取りが可能な知識ベースが増えている．このような知識ベースは RDF のようなデータ形式を用いて論理的に一貫した形式で記述がなされている．構造化された知識ベースの代表例としては，DBPedia，WikiData などが存在する．

2.2.2 RDF (Resource Description Framework)

知識ベース等の様々な資源を構造的に記述するためのデータ形式として，RDF(Resource Description Framework) [3] がある．RDF とは，Web 上のリソースのメタデータを記述するための枠組みである．

RDF データは最小単位として，主語 (Subject)-述語 (Predicate)-目的語 (Object) からなるトリプルを持つ．トリプルの各要素は，基本的には Web 上のデータの識別子である URI(Uniform Resource Identifier) を用いて表現される．URI 以外の表現方法としては主語，述語，目的語の各要素ごとに制約が異なる．主語は URI または空白ノードで構成される．述語は URI を持つ．目的語は URI またはリテラルか空白ノードで構成される．リテラルとは，URI のように形式化された識別子ではなく文字列をそのまま記述したものを示す．

RDF トリプルの例を図 4 に示す．この例は，「dbpedia:サンタクロース」は「dbpedia-owl:Person」に対して「rdf:type」という性質を持つ，ということを表している．

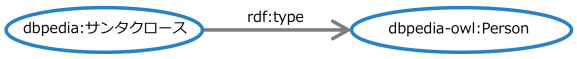

図 4　RDF トリプルの例

この RDF トリプルを組み合わせることによって RDF グラフが構成される．RDF グラフは，主語と目的語がノード，述語がエッジを示すラベル付き有向グラフとして表現される．以後の表記として，URI を表すノードは楕円で，リテラルを表すノードは長方形で表現することとする．

RDF グラフの例を図 5 に示す．この例では，二つのトリプルがリテラルを持つ．

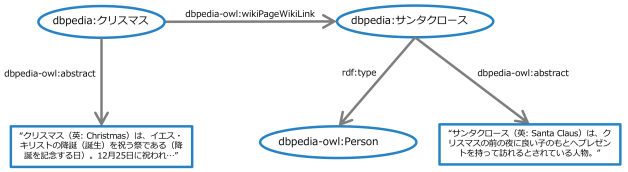

図 5　RDF グラフの例

3　関 連 研 究

3.1　単語を次元として扱うテキスト OLAP 分析

TextCube [4] や CXT-Cube [5] は，単語の階層構造を用いることでテキストに対する OLAP 分析を実現する．単語の階層構造の例を図 6 に示す．この例では，「Memory」「CPU」「Disk」のような同じカテゴリに属する単語を統合することで対象の単語についての階層構造を構築している．このように予め定義しておいた単語の階層構造を利用することで，テキストから抽出した単語の次元に対しても，通常の OLAP 分析と同様にロールアップやドリルダウンのような集約操作を行うことが可能となった．

しかし，テキスト中に含まれる単語について，文脈を考慮していないため，語義の曖昧性などの問題に対処することができなかった．また，分析したい分野ごとに単語の階層構造を定義しなければいけない上，扱うことのできる語彙数に限りがあった．

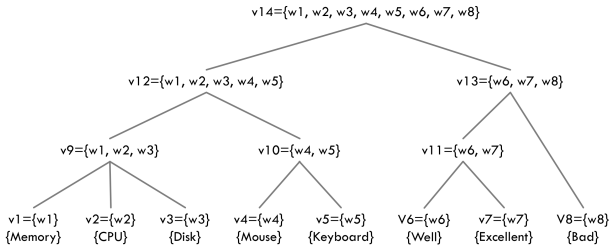

図 6　単語の階層構造

3.2　トピックを次元として扱うテキスト OLAP 分析

TopicCube [6] は，トピックの次元を生成することでテキストに対する OLAP 分析を実現している．テキストに対して潜在的トピックモデルである PLSA [7] を適応し，トピックを抽出する．得られたトピックごとの単語や文書の出現確率を用いることで，新たにトピックについての次元を生成し，テキストの意味を考慮した OLAP 分析を可能にしている．さらに，図 7 に示すようなトピックの階層構造を表すトピックツリーを利用することで，トピックに対する集約演算を行うことができる．このトピックツリーを上にたどる操作を行うことでトピックに対するロールアップを，下にたどる操作を行うことでトピックに対するドリルダウンを実現する．

しかし，TopicCube ではテキストの内容を包括的に捉えたトピックのみを対象としているため，テキストに含まれる個別の単語について考慮した，より詳細な分析を行うことはできない．また，各トピックについてロールアップなどの集約操作を行うために，分析をする分野ごとにトピックツリーを定義しなければならないといった問題点も挙げられる．

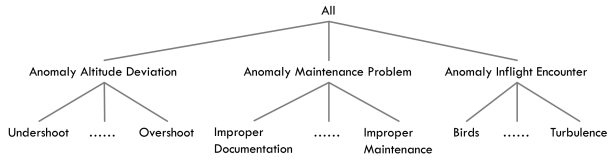

図 7　トピックツリー

4　提 案 手 法

4.1　提案手法のアイディア

本研究では，テキストを含む構造化データに対して，テキストに含まれるエンティティに着目し，それを知識ベースと関連づけることで，エンティティに着目した OLAP 分析を可能に

する手法を提案する．エンティティと知識ベースとの関連づけには，既存のエンティティリンキングの手法を用いる．

エンティティリンキング [8] とは，テキスト中の単語を知識ベース内のエントリと結びつける技術である．与えられたテキスト中の何らかのエンティティを参照する記述であるメンションを検出し，知識ベース内の対応するエンティティと結びつけることができる．エンティティリンキングでは，テキストからエンティティ名の抽出を行うだけでなく，テキストの文脈を考慮してメンションを知識ベースに結びつけることでエンティティの曖昧性の解消まで行う．これによって，テキスト中の単語について，表記の揺れや語義の曖昧性を回避してエンティティのクラスやエンティティ間の関係を正確に扱うことが可能となる．なお本研究では，エンティティリンキングの手法は既存のものを用いることとし，その詳細には立ち入らない．

また，テキストから抽出したエンティティについて，知識ベースがもつ概念階層や，知識ベースに記述されたエンティティに対する付加的な情報を用いることで，構造化データだけでは不可能な OLAP 分析が可能となる．すなわち，テキストから抽出したエンティティの次元に対してもロールアップのような集約の操作が可能となる．

4.2 データモデル

以下に，提案するモデルの持つデータ構造を定義する．

4.2.1 知識ベース

本研究で用いる知識ベースは RDF で記述されているものを想定し，RDF グラフとみなすことができる．知識ベースを表す RDF グラフ G_{KB} について，グラフ内のノード集合を V，エッジ集合を E，エッジに張られるラベルを ρ としたとき，次のように表現することができる．

$$G_{KB} = (V, E, \rho)$$

ここで，$V = V_{ENT} \cup V_L \cup V_B$ であり，V_{ENT} は URI で表されるエンティティを持つノードを，V_L はリテラルを持つノードを，V_B は空白ノードを示す．これらのノードの関連を示すエッジ集合 E は URI か空白ノードを持つ主語と，URI またはリテラルか空白ノードを持つ目的語間の関係を表すため，$E \subseteq (V_{ENT} \cup V_L) \times V$ となる．また，G_{KB} に含まれる全ての述語であるプロパティの集合を P としたとき，ρ は $E \to P$ となり，関連からプロパティへの写像を表す．

4.2.2 エンティティサブグラフ

本手法では，知識ベースの RDF データのうち，エンティティリンキングで関連づけられたエンティティと，そのエンティティについての性質が記述されたドキュメントを用いる．トリプルのうち目的語に関してはリテラルという文字列を持つことができる．目的語がリテラルであるトリプルの多くが，あるエンティティについての概要や説明文を表している．これを利用するために，各エンティティについてリテラルを持つものをまとめて持っておくエンティティサブグラフを定義する．

知識ベースに含まれるエンティティの集合 V_{ENT} のうち i 番目のエンティティについてのエンティティサブグラフを考える．i 番目のエンティティが持つ全てのトリプルのうち，リテラルを目的語に持つトリプルを探索する．これを i 番目のエンティティについてのエンティティサブグラフとし，エンティティサブグラフ i と呼ぶこととする．エンティティサブグラフ i の例を図 8 に示す．この図において，「dbpedia:サンタクロース」というエンティティのエンティティサブグラフは，三つのリテラルの値を持つ．

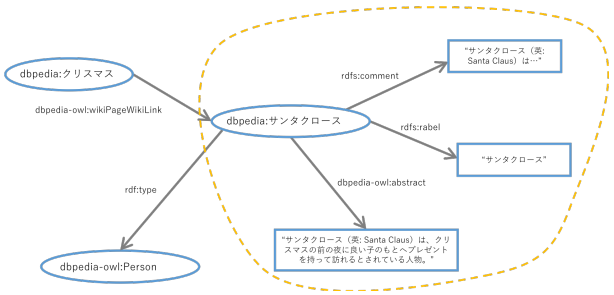

図 8　エンティティサブグラフの例

エンティティサブグラフ i の定義は次の通りになる．

$$G_{ES}^{(i)} = (V_{ES}^{(i)}, E_{ES}^{(i)}, \rho_{ES}^{(i)})$$

ここで，$V_{ES}^{(i)} = \{v^{(i)}\} \cup \{v \mid v \in V_L \ s.t. \ (v^{(i)}, v) \in E\}$ であり，$E_{ES}^{(i)} \subseteq \{v^{(i)}\} \times \{v \mid v \in V_L \ s.t. \ (v^{(i)}, v) \in E\}$ となる．また，G_{KB} に含まれる全プロパティの集合 P に対して，$\rho_{ES}^{(i)}$ は $E_{ES}^{(i)} \to P$ となり，関連からプロパティへの写像を表す．

エンティティ $v^{(i)}$ において，$p \in P$ であるプロパティ p を介して参照されるリテラルを $v^{(i)}.p$ と表記することとする．

4.2.3 エンティティサブグラフ・グラフ

G_{KB} に含まれる全てのエンティティ間の関係を表したグラフをエンティティサブグラフ・グラフ G_{ES} とする．あるエンティティの性質を表すリテラルを持つトリプルについては各エンティティのエンティティサブグラフで扱い，エンティティサブグラフ・グラフ G_{ES} ではリテラルを除いたエンティティ間の関係のみを考える．エンティティサブグラフ・グラフの定義は次の通りになる．

$$G_{ES} = (V_{ES}, E_{ES}, \rho_{ES})$$

ここで，エンティティサブグラフ・グラフのノードはリテラルでないエンティティのみを考えるため $V_{ES} = V_{ENT}$ であり，$E_{ES} \subseteq V_{ENT} \times V_{ENT}$ となる．また，G_{KB} に含まれる全プロパティの集合 P に対して，ρ_{ES} は $E_{ES} \to P$ となり，関連からプロパティへの写像を表す．

4.2.4 入力データ

入力としてリレーションスキーマ $R_s(A_1, A_2, ..., A_k, T, N)$ のデータを考える．ここで，A_k はテキストを含まない数値等で表される通常の属性である．T はテキスト属性であり，最低一つは存在することを想定する．本研究では，単一属性を対象とする．N は OLAP 分析の対象となるメジャー属性であり，基本的には数値属性となる．

4.3 エンティティリンキング

以下に，データモデルに対するエンティティリンキングの定義を示す．

4.3.1 エンティティリンキングの定義

リレーションスキーマの各タプルは $j = (a_1^{(j)}, a_2^{(j)}, ..., a_k^{(j)}, t^{(j)}, n^{(j)}) \in R_s$ として表すことができる．このうち，テキスト属性 $t^{(j)}$ に対してエンティティサブグラフ・グラフを用いてエンティティリンキングを適用すると，複数のメンション $M^{(j)} = \{m_1^{(j)}, m_2^{(j)}, ..., m_l^{(j)}\}$ が検出され，この各メンションに対してエンティティが関連づけられる．各タプルのテキスト属性 $t^{(j)}$ に対するエンティティリンキングの定義としては次の通りになる．

$$l^{(j)} : M^{(j)} \rightarrow V^{(j)}$$

各タプルにおいて検出されたメンションの集合 $M^{(j)}$ に対して，関連づけられたエンティティとして $V^{(j)} = \{v_1^{(j)}, v_2^{(j)}, ..., v_l^{(j)}\}$ が得られる．エンティティリンキングの例を図 9 に示す．この例では，入力として与えられたテキストから「クリスマス」「プレゼント」「母」の三つのメンションを検出し，それぞれについてエンティティサブグラフ・グラフの「クリスマス」「贈り物」「母親」のエントリと関連づけている．エンティティリンキングでは，テキストの文脈を考慮してメンションを知識ベースに結びつけるため，エンティティの表記揺れや語義の曖昧性の問題を解決する．図 9 の例から，「母」「お母さん」など複数の表記を持つエンティティを「母親」という一つの概念として捉えられることが示されている．

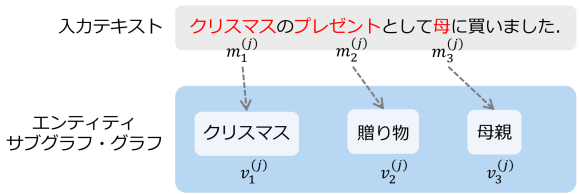

図 9　データモデルに対するエンティティリンキングの例

4.3.2 エンティティのカテゴリ分類

抽出されたエンティティの数は非常に膨大であり，この全てをそのまま次元として扱った分析も可能ではあるが，よりデータを集計した分析も必要になってくる．そこで，本研究では抽出したエンティティのカテゴリ分類を行いエンティティについての階層構造を構築しておくことで，エンティティの次元に対するロールアップを可能にする．これを実現するために，エンティティを分類する二つの方法を提案する．一つ目は知識ベースにおけるエンティティの概念階層を利用する方法，二つ目はエンティティの持つリテラルの値を利用する方法である．それぞれについて以下で詳細を述べる．

a）　エンティティの概念階層の利用

抽出したエンティティの集約を行うために知識ベースにおけるエンティティの概念階層を利用する．知識ベースにはあらゆるエンティティ間の関係が記述されている．このうち，エンティティのカテゴリ区分を表す概念階層を示している関係を抽出することで，エンティティの階層構造を構築することができる．

エンティティサブグラフ・グラフ G_{ES} において，概念階層を表すプロパティを用いることでエンティティの概念階層を抽出する．扱う対象となるエンティティ $v \in V_{ENT}$ からリンクが張られている全エンティティのうち，より上位の概念を表すプロパティを持つトリプルのエンティティについては対象エンティティを包含するグループであるとする．このようにして，対象エンティティ v の上位概念を表すエンティティを含むトリプルを抽出することで，v を頂点とするツリーを抽出することができる．このツリーを対象エンティティ v についての概念階層とする．分析で扱う全てのエンティティに対してこの概念階層を抽出しておくことで，エンティティのカテゴリ分類を行うことが可能となる．

b）　エンティティサブグラフの利用

抽出したエンティティの集約を行うためにエンティティが持つリテラルの値，つまりは，各エンティティのエンティティサブグラフを利用する．ここで，エンティティが持つリテラルの値をそのエンティティの属性と定義する．i 番目のエンティティについてのエンティティサブグラフ $G_{ES}^{(i)}$ は，エンティティ $v^{(i)}$ が参照する属性を持つ．属性はあるエンティティの性質を表すため，これを利用することでエンティティの集約を行うことができる．例えば，抽出されたエンティティのうち，それぞれのエンティティサブグラフにおいて，ある特定のプロパティを持つトリプルの属性が一致するものを一つのグループとして集約することができる．つまり，エンティティ $v^{(i)}$ と $v^{(j)}$ について，$p \in P$ であるプロパティ p を介して参照される属性である $v^{(i)}.p$ と $v^{(j)}.p$ が一致する場合，この二つのエンティティは同じグループに属するとみなすことができる．このようにして，分析で扱う全てのエンティティのエンティティサブグラフを利用することで，エンティティの階層構造を構築することが可能となる．

4.4 操　　作

抽出されたエンティティとエンティティの階層構造を用いて OLAP 分析の操作を行う．

4.4.1 スキーマ生成

得られたエンティティリンキングの結果と元の入力データからスキーマを作成する．入力データ内のテキストでない通常の属性 A_i については従来のデータキューブと同様に，そのものを次元として用いる．テキスト属性 T についてはエンティティリンキングで得られたエンティティを次元として用いる．エンティティの概念階層を用いることで，エンティティ次元の階層構造を構築する．

4.4.2 キュービング

ユーザが選んだ次元を使って実際にキュービングを行う．各セルは $(a_1, a_2, ..., a_n, e : D, F(D))$ という形式で表される．各次元値に対応するエントリの集合がそれぞれのセルに割り当てられる．また，このエントリの集合 D に対してカウントや平均などの操作を行った結果である $F(D)$ についても同様にメ

ジャーとして収められる．図 10 に，入力されたレビューデータについて実際にキュービングを行なった例を示す．例で示されているキューブは各セルにそれぞれの次元の値に当てはまるレビューの集合が割り当てられ，レビューのカウント数がメジャーとして収められている．

ユーザは得られたキューブに対してロールアップやドリルダウンなどの操作を実行できる．通常の属性 A_i から得られた次元についてはそこに内在されている階層構造を元に粒度を変えた分析を行う．テキスト属性 T から得られたエンティティについては，エンティティの階層構造を用いることで粒度を変えた分析が可能となる．

レビューの集合 D

ID	性別	年齢	レビューテキスト
1	その他	31	クリスマスプレゼントに買いました
8	その他	38	おかげで良いクリスマスを過ごせた
19	その他	35	クリスマスシーズンにちょうどいい
⋮	⋮	⋮	⋮

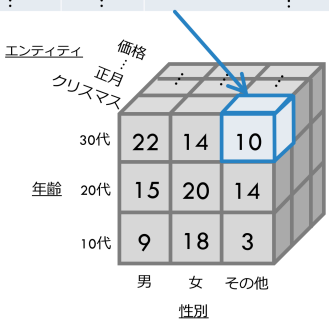

図 10　提案手法で構築されたキューブの例

5　システムアーキテクチャ

本章では，本研究で提案する知識ベースを用いたテキストの OLAP 分析を行うシステムのアーキテクチャを示す．このアーキテクチャでは，テキストを含む構造化データを分析対象とする．

図 11 に本研究で提案するシステムアーキテクチャの概要を示す．入力としては，分析対象となるデータと分析に用いる知識ベースが与えられる．まず，入力の分析対象のデータを格納する．また，入力で選択された知識ベースからエンティティ間の関係についての情報を抽出したエンティティサブグラフ・グラフを生成しておく．次に，エンティティリンカーが分析対象のデータのうちテキスト属性であるデータに対してエンティティサブグラフ・グラフを用いてエンティティリンキングを行う．この結果得られた，各レコードに含まれるエンティティとそれに対応するメンションの情報，そしてエンティティサブグラフの情報を格納しておく．入力データとエンティティリンキングの結果を用いることで，データの集約を行う．

また，本研究では，日本語のテキストに対してエンティティリンキングを行うエンティティリンカーとして word2vec-wikification-py を用いる．[3] このエンティティリンカーでは，入力テキストの形態素解析を行い，その結果得られた各単語に対して Wikipedia 記事の候補を作成する．それぞれの単語の記事の組み合わせのうち記事間の類似度を考慮して最も意味が近い組み合わせを選択することによって，文脈を考慮したエンティティの抽出を行うことが可能となる．記事間の類似度の計算には，鈴木らの日本語 Wikipedia エンティティベクトル [4] [9] を用いている．

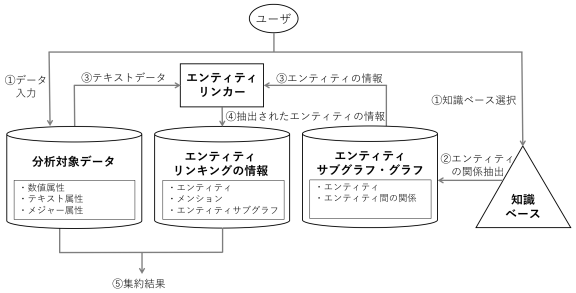

図 11　システムアーキテクチャの概要

6　評価実験

6.1　実験概要

本実験では，提案手法がエンティティの表記揺れや語義の曖昧性の解消を行い，単語を文字列として扱う従来手法に比べて少ないコストで，より正確な分析が可能となることを示す．

6.2　実験方法

6.2.1　データセット

二つのデータセットを対象に実験を行なう．

一つ目として，NHN Japan 株式会社が運営する「livedoor ニュース」のうちクリエイティブコモンズライセンスが適応されるニュース記事を収集したコーパスである，livedoor ニュースコーパス [10] を用いる．ニュースデータには多くの人名が含まれるが，これらの多くは表記が統一されていない．そこで，このデータセットをエンティティの表記揺れについての実験の評価に用いる．

二つ目として，「楽天データ公開」において公開されている，EC サイト「楽天市場」に投稿された商品レビューのデータ [11] を使用する．このデータは，投稿者の年齢・性別，商品名，評価ポイント，レビューなどテキスト属性を含め 17 属性を持つデータになっている．レビューデータは非常に多様な単語を含んでいる．その中には，服飾名と作品名の二つの意味を持ち合わせる「ワンピース」のような，多義語が多く存在している．そこで，このデータセットをエンティティの語義の曖昧性についての実験の評価に用いる．

6.2.2　実験環境

本実験の実行環境を表 1 に示す．入力データを用いて大規模知識ベースである DBPedia に対してエンティティリンキングを行う．エンティティリンカーとしては word2vec-wikification-py

3：https://github.com/Kensuke-Mitsuzawa/word2vec-wikification-py

4：http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

を用いる．エンティティリンキングの結果得られたエンティティについて，そのエンティティの DBPedia における概念階層を用いることでエンティティをカテゴリごとに分類する．エンティティリンキングを行い，カテゴリを抽出する部分は Python 3.7.3 で実装を行なった．入力データとエンティティリンキングを行なった結果を用いて SparkSQL で集約演算を行う．

表 1　実 行 環 境

CPU	2.3GHz Intel Core i5
Memory	16GB
OS	macOS High Sierra 10.13.6

6.2.3 比 較 手 法

比較を行う対象として，従来手法の，事前に定義した辞書を用いることでテキストから単語の抽出を行う方法を用いる．分析対象となる単語とその単語の意味的なカテゴリ階層を定義することで，事前に辞書を構築する．この辞書に含まれる単語が分析対象となるテキストに含まれる場合に，これを検出する．

6.3 実 験 結 果

6.3.1 実行コストと精度の比較

比較手法である単語ベース手法と提案手法について，実行コストと精度についての比較を行う．livedoor ニュースコーパスのうち，スポーツに関するニューステキスト 100 件の分析を行う．単語ベース手法における辞書は，対象となるデータに関係するデータとして Wikipedia における「日本のスポーツ選手」カテゴリ下の記事を MediaWiki API[5]を用いて全て収集することで生成する．

まず，前処理と実行にかかる実行時間の比較を行う．単語ベース手法においては，辞書の構築にかかる時間を前処理の時間とする．提案手法については，事前に辞書の構築を行う必要はない．また，単語ベース手法における辞書を用いて単語抽出を行う時間，提案手法におけるエンティティリンキングを行い，抽出されたエンティティのカテゴリ分類を行う時間を実行時間とする．

比較を行なった結果を表 2 に示す．提案手法は辞書を構築する必要がないため，単語ベース手法よりも少ない時間で分析を実行できることが読み取れる．

表 2　実行時間の比較

	前処理	実行時間 (s)	前処理＋実行時間 (s)
単語ベース手法	1680	0.85	1681
提案手法	—	921.28	921

次に，抽出された単語の精度の比較を行う．事前に，対象となるデータセットの各記事に含まれるスポーツ選手名をその記事ついての正解となるエンティティとして真の結果を定義しておく．単語ベース手法と提案手法が，正解となるエンティティをどれだけ検出できているか，また検出した単語がどれだけ正解しているかを評価する．評価の指標として，表 3 の混合行列を用いて，Accuracy，Precision，Recall を次のように定義する．

表 3　混 合 行 列

予測結果＼真の結果	True	False
True	True Positive	False Positive
False	False Negative	True Negative

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

抽出された単語の精度を比較し結果を表 4 に示す．抽出された単語の Accuracy と Recall について，提案手法がより高い値となっている．提案手法ではエンティティの表記揺れや語義の曖昧性の問題を解消することができるため，高い精度を保ったまま，より多くの単語を抽出することが可能となっている．

表 4　抽出された単語の精度の比較

	Accuracy	Precision	Recall
単語ベース手法	98.6%	100.0%	46.0%
提案手法	99.4%	100.0%	78.0%

6.3.2 エンティティの表記揺れの解消についての精度評価

提案手法が従来手法と比較して精度が向上した一因として考えられるエンティティの表記揺れの解消についての評価を行う．

「中山雅史」というエンティティは「中山雅史」の他に,「ゴン中山」「中山」といった表記でも記述される．このような複数の表記を持つエンティティについて，全ての表記を一つの概念として捉えることができるかを検証する実験を行う．livedoor ニュースコーパスのうち，複数表記を持つエンティティを含むニューステキスト 50 件を対象とした．それぞれのニューステキストには事前に正解となるエンティティを割り当てる．ニューステキストに対して単語抽出またはエンティティリンキングを行い，得られた単語について精度を評価する．

実験結果を表 5 に示す．実験結果より，単語ベースの手法と比較して，提案手法の Recall が高く，より多くの表記揺れしている単語を同一エンティティとして捉えられている．単語ベースの手法では，事前に情報抽出を行なった辞書に含まれる単語のみを対象とする一方，提案手法では人物の本名だけでなくニックネームなども対象としてエンティティを抽出することができる．しかし，エンティティリンキングにおいて「中山」から「中山雅史」というエンティティに結びつけるように，名字のみから人名に結びつけることができない例も多く存在した．

5 : https://www.mediawiki.org/wiki/MediaWiki

表 5　表記揺れを含むテキストの分類精度

	Accuracy	Precision	Recall
単語ベース手法	88.8%	100.0%	44.0%
提案手法	94.4%	100.0%	72.0%

6.3.3　語義の曖昧性の解消についての精度評価

提案手法が従来手法と比較して精度が向上した一因として考えられる語義の曖昧性解消についての評価を行う．

「ワンピース (服飾)」と「ワンピース (作品)」というように，一つの単語が複数の語義を持つ際に，文脈から判断して最もふさわしい意味で捉えられているかを検証する実験を行う．商品レビューのデータのうち，複数の語義を持つ単語を含むレビューテキスト 50 件を対象とした．それぞれのレビューテキストには事前に正解となるエンティティを割り当てる．レビューテキストに対してエンティティリンキングを行い，抽出されたエンティティの精度を評価する．

実験結果を表 6 に示す．単語抽出を行う従来手法では，コンテキストから意味を判別することができないため，語義の曖昧性の解消を行うことは不可能である．しかし，提案手法においては，78.0%の Accuracy で曖昧性の解消ができている．

表 6　語義の曖昧性を含むテキストの分類精度

	Accuracy	Precision	Recall
単語ベース手法	NaN	NaN	NaN
提案手法	78.0%	88.9%	64.0%

7　まとめと今後の課題

本研究では知識ベースに対するエンティティリンキングによってテキストに含まれるエンティティを捉えることで，より高度なテキストの OLAP 分析を行う手法を提案した．提案手法では，テキストからエンティティを抽出することにより，語義の曖昧性の解消や表記揺れの解消を行うことができるため，従来のテキストの OLAP 分析システムでは成し得なかった，より正確な分析が可能となった．また，テキストから抽出したエンティティについて知識ベースに含まれる概念階層を用いることで，ユーザが事前に単語や単語の階層構造の定義を行う必要なく，分析を行うことが可能となった．評価実験により，提案手法が従来手法よりもより少ないコストで，分析を実行できることが示された．また，提案手法が表記揺れや語義の曖昧性の解消を行い，より正確な分析を可能としていることが示された．

今後の課題として，複数の知識ベースを組み合わせた分析を行う手法の検討が挙げられる．本研究では，エンティティリンキングを行う対象として一つの知識ベースのみを用いたが，複数の知識ベースを組み合わせることで，より多くの有用な情報を得ることが可能になると考えられる．そこで，今後は複数の知識ベースに対してエンティティリンキングを行い結果を集約する手法についての研究を進める．

謝　　辞

本研究では，NHN Japan 株式会社様から提供を受けた「livedoor ニュースコーパス」，並びに，楽天株式会社様から提供を受けた「楽天データセット」を利用しました．ここに記して謝意を表します．

文　　献

[1] Edgar F Codd, Sharon B Codd, Clynch T Salley, F Codd, and C Salley. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. 1993.

[2] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering, Vol. 27, No. 2, pp. 443–460, 2014.

[3] Graham Klyne. Resource description framework (RDF): Concepts and abstract syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, 2004.

[4] Cindy Xide Lin, Bolin Ding, Jiawei Han, Feida Zhu, and Bo Zhao. Text cube: Computing IR Measures for Multidimensional Text Database Analysis. In 2008 Eighth IEEE International Conference on Data Mining, pp. 905–910. IEEE, 2008.

[5] Lamia Oukid, Ounas Asfari, Fadila Bentayeb, Nadjia Benblidia, and Omar Boussaid. CXTcube: contextual text cube model and aggregation operator for text OLAP. In Proceedings of the sixteenth international workshop on Data warehousing and OLAP, pp. 27–32, 2013.

[6] Duo Zhang, Chengxiang Zhai, and Jiawei Han. Topic cube: Topic modeling for olap on multidimensional text databases. In Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 1124–1135. SIAM, 2009.

[7] Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 50–57, 1999.

[8] 乾健太郎，松田耕史，岡崎直観．日本語 wikification ツールキット:jawikify. 言語処理学会 23 回年次大会, pp.250-253, 2017.

[9] 鈴木正敏, 松田耕史, 関根聡, 岡崎直観, 乾健太郎. Wikipedia 記事に対する拡張固有表現ラベルの多重付与. 言語処理学会第 22 回年次大会, 2016.

[10] livedoor ニュースコーパス. http://www.rondhuit.com/download.html#ldcc.

[11] 楽天株式会社 (2014): 楽天データセット. 国立情報学研究所情報学研究データリポジトリ. (データセット). https://doi.org/10.32130/idr.2.0.