知識ベースを用いた検索語の汎化構造解析とそれに基づくエンティティ検索支援

桐谷雄介† 馬強†† 吉川正俊††

† 京都大学工学部情報学科
〒 606–8501 京都府京都市左京区吉田本町
†† 京都大学大学院情報学研究科
〒 606–8501 京都府京都市左京区吉田本町
E-mail: †y.kiritani@db.soc.i.kyoto-u.ac.jp, ††{qiang,yoshikawa}@i.kyoto-u.ac.jp

あらまし 近年，人物や場所などに関するエンティティ検索が注目されている．一方，現在の主流検索エンジンは，キーワードベースのものが多く，ユーザによる検索語の入力が要求される．ユーザが入力した検索語は，必ずしもユーザの求めるもののみを指すとは限らず，むしろ計算機上では複数の意味に取られるような語になってしまう方が多い．そのため，検索結果にはユーザの意図しなかった意味での検索語の結果が多く含まれ，さらなる処理が必要となり，ユーザの負担が大きい．そこで，本研究では，既存の知識ベースを利用した検索語の‘ type ’や‘ subClassOf ’といった汎化構造の解析手法と，キーワードベースのエンティティ検索における検索支援として，キーワードの曖昧性解消と検索結果のカテゴライズを試みる．

キーワード 知識ベース，エンティティ検索，最小包括クラス，汎化構造

Generalization Structure Analysis of Query Terms based on a Knowledge Base and its Application to Entity Retrieval

Yusuke KIRITANI†, Qiang MA††, and Masatoshi YOSHIKAWA††

† School of Informatics and Mathematical Science, Faculty of Enginnering, Kyoto University
Yoshidahonmachi, Sakyo-ku, Kyoto 606–8501, Japan
†† Department of Social Informatics, Graduate School of Informatices, Kyoto University
Yoshidahonmachi, Sakyo-ku, Kyoto 606–8501, Japan
E-mail: †y.kiritani@db.soc.i.kyoto-u.ac.jp, ††{qiang,yoshikawa}@i.kyoto-u.ac.jp

Abstract Recently, entity retrieval for finding information on certain persons or places from the Web becomes a new challenge. Since the keyword is ambiguous, current keyword-based search engine will return various Web pages rather than that about the certain entity expected by a user. we propose methods for removing query-sense ambiguity and search results clustering based on the generalization structure analysis of query-term. The generalization structure analysis is based on the *type* and *subClassOf* relationships between entities in the well-known knowledge-base YAGO. We also introduce a prototype system to demonstrate the feasibility of our methods.

Key words Knowledge Base, Entity Retrieval, Generalization Structure, Minimal Exhaustive Classes

1. 序 論

現在，一般に Web 上の情報を検索するのはページ検索が主流であるが，人や場所などの固有実体（エンティティ）に対する検索，いわゆる，エンティティ検索 [1] の要求も多い．既存の検索エンジンを用いたエンティティ検索では，検索語の曖昧性が生じやすい．例えば，ユーザが ‘washington’ というキーワードを入力したとき，‘Washington state’,‘Washington, D.C.’ というような地名，‘George Washington’ というような人名，‘The Washington Post’ というような新聞名などといった候補が多数出てきてしまい，ユーザの意図しなかった検索結果も多く含まれてしまう．そのため，さらなる処理が必要とされ，ユーザの負担が大きい．

従来，一般の Web ページ検索を対象とした曖昧性解消のた

めの質問拡張やクラスタリング手法が提案されているが，検索されたページに現れる語を利用しているため，人物などの実世界に存在するエンティティの特徴を十分に利用できてないので，エンティティ検索には不十分であり，さらなる工夫が必要である．

近年，インターネット上のフリー百科事典 Wikipedia [2] が注目されている．Wikipedia には，人物などエンティティに関する情報が多数収録されている．Wikipedia のページを対象に，エンティティの関係やカテゴリ情報を利用して，YAGO [3] や DBpedia [5] などの知識ベースが構築されつつある．

そこで本論文では，エンティティ検索の支援として，YAGO の知識ベースを利用して汎化構造を解析することによる曖昧性解消とクラスタリング手法を提案する．本論文では，YAGO の以下の４つのアイテムに注目して，エンティティの汎化構造を解析する．

- 人物・地名など固有名を表すエンティティ（以下，インスタンスと呼ぶ）
- エンティティを分類するエンティティ（以下，クラスと呼ぶ）
- インスタンスとクラス間の ‘type’ 関係
- ２つのクラス間の ‘subClassOf’（親子）関係

この汎化構造を利用して，インスタンスをクラスで分類することで，インスタンスの曖昧性解消と検索結果のクラスタリングを行う．そのため，インスタンス集合の要素それぞれの持つ全ての属性を最大限に表現するための最小のクラス集合となる極小包括クラス (Minimal Exhaustive Classes (MEC)) を定義する．一般的に極小包括クラスは，与えられたインスタンス（集合）の特性をを過不足なく，より具体的に表現できるクラス集合として定義される．

本研究では，MEC を用いた検索語の曖昧性解消手法と検索結果のクラスタリング手法を提案して，エンティティ検索の支援を行う．

本研究では，曖昧性解消は，ユーザにとって不確かで曖昧なインスタンスをその語の一部分（キーワード）から見つける処理である．まず，YAGO からキーワードを含むインスタンス集合が得られ，これを候補集合とする．ユーザの意図したインスタンスはこの候補集合に入っていると仮定する．この集合の各インスタンスは，MEC というクラス集合により全て分類されていることを利用して，MEC のクラスごとにそれの持つインスタンス集合を提示する．ユーザは候補集合を絞りながら意図したインスタンスを見つけ出す．

ユーザに検索語の候補集合を提示するため，われわれは，MEC に基づいて計算されるクラスのメジャー度を用いた．メジャー度はクラスの具体性と所有候補の総数で計算される．メジャー度を用いて，検索語の候補をメジャーとマイナーに分類することが可能となり，ユーザが自分の目的（メジャーな検索をしたいか，マイナーな検索をしたいか）に応じて検索語を指定できる．

本研究では，検索結果を MEC にあるインスタンスとの対応関係に基づいて分類する．YAGO から，キーワードを含むインスタンス集合が得られ，その各インスタンスを MEC というクラス集合の数だけクラスタに振り分けることができることを利用して，各インスタンスを分ける．そして，Web 検索結果の各ページはページ内に含むインスタンスが分けられたクラスタに Web 検索結果のクラスタリングを行う．

以下，本論文の構成を示す．２節では，関連研究としてエンティティ検索，検索支援について紹介する．３節では，汎化構造と極小包括クラス（MEC）について述べる．４節では，エンティティ検索支援のための MEC の計算方法を述べ，それに基づく検索支援手法を紹介する．５節では，プロトタイプシステムについて説明する．６節は，本論文のまとめである．

2. 関 連 研 究

エンティティ検索，特に，人物検索として，さまざまな方法が研究されている．文献 [8] では，知識ベースを用いて，文書間の類似度を測ることにより，文書をクラスタリングして人物と文書のマッピング手法を提案している．文献 [9] は属性に着目したオブジェクトの特定手法を網羅的に紹介している．文献 [10] は検索結果ページを人ごとにクラスタリングする手法を提案した．属性情報のほか，リンク情報とページの類似度を用いたことが特徴である．WebHawk [11] はジャンクページの排除，人物情報の抽出とクラスタリング，ラベル付けといった処理からなる人物検索結果の分類手法を提案した．欧米人の名前の特徴であるミドルネームがあることをうまく利用している．Bekkerman らは Social Network 内の人間関係を用いた同姓同名の分離手法を提案している [12]．また，木村らは検索結果に現れるページタイトルとスニペット内の関連オブジェクトと職業情報を用いた手法を提案した [13].

これらは，人物特定に関する手法である．それに対して本研究では，人物を含むエンティティ検索，特定のためのキーワード（検索語）の曖昧性の解消の手法を提案している．また，知識ベースに格納されているエンティティ（オブジェクト）の汎化構造を利用，解析している点も異なる．

質問拡張による検索支援は，検索結果から適切な語を抽出して行う手法が多い [14]．これに対して，Google サジェスト [15] や goo のキーワードアシスト [16] は辞書や検索履歴を利用している．また，文献 [17] は単語の出現頻度や単語間の共起関係を考慮して，追加キーワードを選定し，ユーザの検索支援手法を提案している．

これらの従来手法は，一般のページ検索における検索質問の曖昧性を解消するためのものである．本研究では，知識ベースにおけるエンティティの汎化構造を用いる点や，検索結果のカテゴリの先読みから候補を選別する点が異なる．[18]

汎化構造の解析において，田中は，インスタンスとクラスの関係から汎化階層を解析してオブジェクト集合を抽象化するために，MEC の概念とそれに基づく一般化，明確化と統合などの手法を提案している．本研究では，インスタンス集合を分類するものとして最小包括クラスというものを定義したが，類似する概念 MEC（極小包括クラス）を定義しているが，MEC に含まれるクラス数は可変である点は新しい．

Denzel W	*type*	petecostals
Denzel W	*type*	actor
Isaiah W	*type*	actor
	...	
Cheney, W	*type*	village
actor	*subClassOf*	entertainer
president	*subClassOf*	leader
	...	
physical entity	*subClassOf*	entity

図 1　ファクトの例

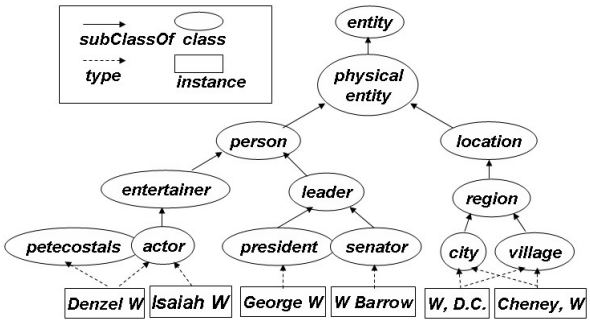


図 2　図 1 から得られる汎化構造の例

3. 最小包括クラス

本論文では，YAGO の ‘type’ や ‘subClassOf’ などのエンティティ間の関係を利用して検索語の汎化構造を解析する．

YAGO は，Wikipedia [2] と WordNet [4] から統合的に情報を抽出した知識ベースである [3]．現在 170 万個のエンティティと，1400 万個のファクト (fact) を有している．YAGO におけるエンティティは，固有名（e.g. ‘*Elvis Presley*’），数字や日付などの文字列定数（e.g. ‘1912’），エンティティを分類するクラス（e.g. ‘*singer*’），エンティティ間の関係を表す関係名（e.g. ‘$isPartOf$’）などからなる．一方，ファクトは，

エンティティ　関係　エンティティ

の 3 つ組から出来ている．例えば，Albert Einstein がノーベル賞を受賞したというファクトは

$Albert\ Einstein \quad hasWonPrize \quad Nobel\ Prize$

というようになる．また本論文では，固有名で表されるエンティティをインスタンスと呼ぶことにする．

本研究では，我々は，YAGO の以下の 4 つの項目（2 種類のエンティティとそれらに関する 2 種類のファクト）を用いて，検索語の汎化構造を解析する．

1. インスタンスを表すエンティティ
2. クラスを表すエンティティ
3. ‘type’ という関係を用いて「インスタンス ins のタイプはクラス c である」を意味するファクト
   $ins \quad type \quad c$
4. ‘subClassOf’ という関係を用いて「クラス c_1 はクラス c_2 のサブクラスである」を意味するファクト
   $c_1 \quad subClassOf \quad c_2$

例えば，インスタンス $I = \{Denzel\,W,\ Isalah\,W\ George\,W,\ W\ Barrow\,W,\ D.C.,\ Cheney,\ W\}$ とそのクラスに関する図 1 のファクトから図 2 の汎化構造が得られる．

また，本論文では，クラス間の親子関係を表す ‘subClassOf’ の関係を持つファクトの推移的閉包から得られる関係を ‘subClassOf∗’ で表す．すなわち ‘subClassOf∗’ はクラス間の子孫関係を表すことになる．例えば，図 2 のファクトから，新たに

$actor \quad subClassOf* \quad entertainer$
$leader \quad subClassOf* \quad entity$
$city \quad subClassOf* \quad location$

などのファクトが得られる．

3.1 集合の定義

本論文で用いる集合を定義する．以下 I と C は，それぞれインスタンス集合とクラス集合を表すとする．

インスタンス ins $(\in I)$ と ‘type’ 関係にあるクラスの集合を $\boldsymbol{type(ins)}$ とする．同様に，インスタンス集合 $I = \{ins_1, ins_2, \cdots, ins_n\}$ $(\subseteqq I)$ に対して，

$$type(I) = \bigcup_{i=1}^{n} type(ins_i)$$

と定義する．

クラス c $(\in C)$ に対して，

$c_{sub} \quad subClassOf* \quad c$
$c \quad subClassOf* \quad c_{sup}$

という ‘subClassOf∗’ 関係が成立するクラス c_{sub} $(\in C)$ の集合とクラス c_{sup} $(\in C)$ の集合をそれぞれ $\boldsymbol{subClassOf*(c)}$ と $\boldsymbol{supClassOf*(c)}$ とする．

また，c $(\in C)$ に対して，

$ins \quad type \quad c$
または
$c_{sub} \quad subClassOf* \quad c$
$ins \quad type \quad c_{sub}$

という関係が成立するインスタンス ins $(\in I)$ の集合を $\boldsymbol{instance(c)}$ とする．

同様に，クラス集合 $C' = \{c_1, c_2, \cdots, c_m\}$ $(\subseteqq C)$ に対して，

$$subClassOf*(C') = \bigcup_{i=1}^{m} subClassOf*(c_i)$$

$$supClassOf*(C') = \bigcup_{i=1}^{m} supClassOf*(c_i)$$

$$instance(C') = \bigcup_{i=1}^{m} instance(c_i)$$

と定義する．

例えば，図 2 のインスタンス集合 $I = \{Denzel\,W,\ Isalah\,W,\ George\,W,\ W\ Barrow,\ W\ D.C.,\ Cheney\,W\}$ から作られた汎化構造では，

$type(I) = \{petecostals, actor, president, city, state\}$

$supClassOf*(city) = \{region, location, physical\ entity, entity\}$

$subClassOf*(\{leader, region\}) = \{president, city, state, region\}$

$instance(person) = \{Denzel\ W, Isalah\ W, GeorgeW, W\ Barrow\}$

などとなる．

3.2 極小包括クラス

本研究では，YAGO における人物などのエンティティ（以下，インスタンス）の汎化構造を解析して，そのエンティティを表現できる最小包括クラス (MEC: Minimal Exhaustive Classes) を求め，それを用いて検索語の特定（曖昧性解消）と検索結果のクラスタリングを行う．直観的には，MEC とは，与えられたインスタンス（集合）の全ての属性を過不足なく表現できる，可能な限り下位のクラスの集合である．

m 個のインスタンスからなるインスタンス集合 $I = \{ins_1, ins_2, \cdots, ins_m\}$ が与えられたとき，集合 $type(I)$ の各クラスを，自身か下位集合として被覆するようなクラス集合で，過不足なく，可能な限り下位のクラスで表された集合を I の最小包括クラス (MEC) と呼ぶ．I に対して，以下のような条件を満たすクラス集合 $\{c_1, c_2, \cdots, c_n\}$ を $MEC(I, n)$ $(n = 1, 2, \cdots, |type(I)|)$ とする．

1. $\{c_1, c_2, \cdots, c_n\} \subseteqq type(I) \cup supClassOf*(type(I))$
2. $type(I) \subseteqq \bigcup_{i=1}^{n} \{c_i \cup subClassOf*(c_i)\}$
3. $c_i \neq c_j$ なる任意の c_i $(i = 1, 2, \cdots, n)$ と c_j $(j = 1, 2, \cdots, n)$ に対して $c_j \notin subClassOf*(c_i)$
4. 任意の c_i $(i = 1, 2, \cdots, n)$ と $c_{isub} \in subClassOf*(c_i)$ が存在するとき，c_i と c_{isub} を入れ替えると，2. が成り立たない．
5. 3. においてクラス c_i が $c_i \in type(I)$ である場合のみ，$c_j \in subClassOf*(c_i)$ を許す．

一般に，$MEC(I, n)$ が存在しない場合があり，その場合は空集合と定義して，$MEC(I, n) = \phi$ とする．

図 2 を例にとると，

$MEC(I, 1) = \phi$

$MEC(I, 2) = \{petecostals, physical\ entity\}$

$MEC(I, 4) = \{petecostals, actor, leader, region\}, \{petecostals, person, city, village\}$

$MEC(I, 6) = \{petecostals, actor, president, senator, city, village\}$

などになる．

一般に，インスタンス集合 I とある自然数 n に対して，$MEC(I, n)$ が一意に決まらない場合がある．例えば，図 2 のときは先の結果から $n = 4, 5$ で 2 つ以上の集合が得られる．

4. 検索支援

4.1 節では，多く抽出される極小包括クラスの中から，検索支援に適しているものを選んだり，簡略化したりするための計算方法を説明する．4.2 節では，MEC を用いた検索支援手法について説明する．

4.1 検索支援のための極小包括クラスの計算手法

4.1.1 極小包括クラスの一意性

図 2 のように，一般に $MEC(I, n)$ は複数得られる．そのため，実際のアプリケーションに応じて適切なものを選ぶ必要がある場合がある．

例えば，図 2 では MEC(I,4) として図 3 のように以下の 2 種類のクラス集合がありうる．

$C_1 = \{petecostals, person, city, village\}$

$C_2 = \{petecostals, actor, leader, region\}$

このような複数のクラス集合（MEC) から検索支援に用いる MEC を 1 つに絞るための基準として，本研究では，各クラスの持つインスタンス集合の重複度と標準偏差を用いる．

重複度とは，クラスのインスタンス集合間にどれほどのオーバーラップがどのくらいあるかを測るものある．具体的には，各クラス間で重複しているインスタンスの個数で計算される．本研究では，二つのクラス c_i と c_j の重複度 $overlap(c_i, c_j)$ を

$$overlap(c_i, c_j) = |instance(c_i) \bigcap instance(c_j)|$$

と定義する．また，クラス集合 $C = \{c_1, c_2, \cdots, c_k\}$ の重複度 $overlap(C)$ は次のように計算する．

$$overlap(C) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} overlap(c_i, c_j)$$

また，クラスの標準偏差 $\sigma(C)$ はクラスのインスタンス数の標準偏差で定義する．クラスのインスタンス数の平均 $E(C)$ を

$$E(C) = \frac{1}{m} \sum_{1=i}^{m} |instance(c_i)|$$

とすると，

$$\sigma(C) = \sqrt{\frac{1}{m} \sum_{1=i}^{m} (|instance(c_i)| - E(C))^2}$$

で計算する．

重複度が大きいと，その MEC のクラス集合は，同じインスタンスを被覆しているクラスが多いということである．そのような重複は少ない方がインスタンスを区別しやすいので，より重複度の小さいクラス集合を選ぶべきである．また，標準偏差が小さいと，各クラスの持つインスタンスの数のばらつきが小さいということである．ばらつきが小さいようなクラス集合のほうが，インスタンスを均等に分割でき，提示画面での選択がより容易になる場合があると考えられ，より標準偏差の小さいクラス集合を選ぶとする．

図 3 から明らかのように C_1 より C_2 のほうがインスタンスの重複が少なく，インスタンス数のばらつきが小さい．実際，C_1, C_2 の重複度は，$overlap(C_1) = 3$，$overlap(C_2) = 1$ となり，標準偏差は，$\sigma(C_1) = 1.58$，$\sigma(C_2) = 0.25$ となる．よってこの場合 MEC(I,4) として C_2 のほうが本研究における検索支援に適していると考えられ，選択される．

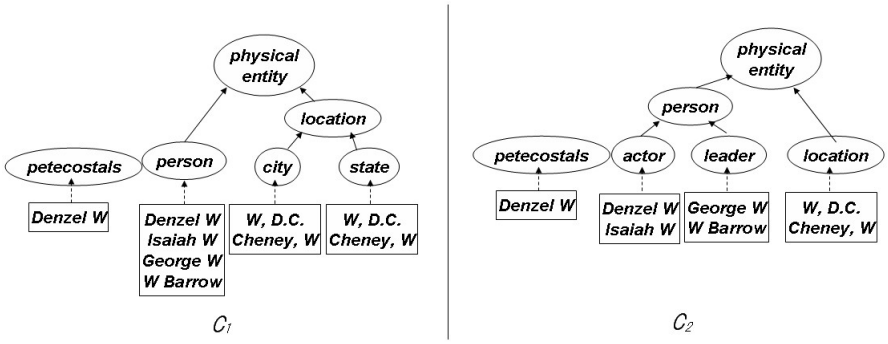


図 3　図 2 の一意に決まらない MEC の例

もし，重複度も分散値も等しいクラス集合の場合，どちらを選んでも同じような表現であると考えられる．

4.1.2　メジャー度を用いた MEC 表示

クラスの性質として，クラス c のインスタンス集合 I に対する関連の高さの尺度メジャー度 $major(c)$ を導入する．インスタンス数が多く，インスタンスまでの距離が近いものをメジャーなクラスと呼び，その逆をマイナーなクラスと呼ぶ．

メジャー度の計算には，クラスからインスタンスまでの平均距離 $h(c)$ というものを用いる．これは，クラス c と c の持つインスタンス集合 $instance(c) = \{ins_1, ins_2, \cdots, ins_m\}$ の各インスタンス ins の距離の平均で定義する．すなわちクラスの平均距離 $h(c)$ は，

$$h(c) = \frac{1}{m}\sum_{1=i}^{m} dis(c, ins_i)$$

となる．ただし，インスタンスの距離とは，クラス c とインスタンス ins 間の長さのことで，汎化構造におけるクラスとインスタンスの間の枝の数として計算され，$dis(c, ins)$ で表す．

そして，メジャー度 $major(c)$ はインスタンスの個数 m の増加に伴い増加し，距離 h の増加に伴い減少するような関数 $f(m, h)$ である．つまり

$$major(c) = f(m, h)$$

とする．本論文では，インスタンスの個数 m とクラスの平均距離 $h(c)$ の単純な分数を用いて

$$major(c) = \frac{m}{h(c)} = \frac{m^2}{\sum_{1=i}^{m} dis(c_i, m)}$$

と定義しておく．

実際の YAGO から得られたデータでそのまま MEC を計算しようとすると，$MEC(I, n) = \{c_1, c_2, \cdots, c_n\}$ のクラスの大半のインスタンスが非常に少なく，1 個または,2 個のクラスが残りの多くのインスタンスを持つようなクラス集合になってしまう．例えば，図 4 のように，$MEC(I, 6) = \{c_2, c_3, c_4, c_5, c_6, c_7\}$ は c_2 と c_3 にインスタンスのほとんどが偏ることになるので，検索支援において効率の悪い表現になってしまう．

そこで，得られた MEC のクラス集合のうち，メジャー度の高いクラス，すなわち含んでいるインスタンス数が多く，インスタンスまでの距離が遠いクラスをメジャーな主クラス集合とする．一方，メジャー度の低いクラス，すなわち含んでいるインスタンス数が少なく，インスタンスまでの距離が近いクラスをマイナーなその他のクラス集合とする．

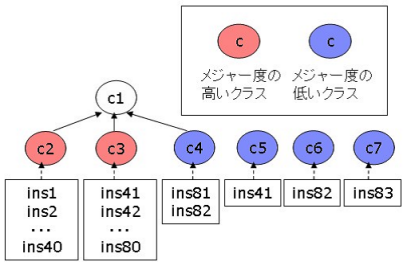


図 4　大半のクラスが少ないインスタンスしか持たない場合

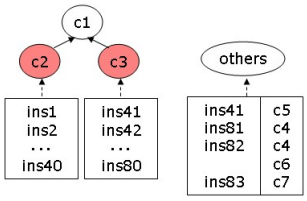


図 5　図 4 の簡略化

MEC のクラス集合のうち，メジャー度が α 以上のクラスをメジャーな主クラス集合として扱い，それ以外のクラスをマイナーなその他のクラス集合として扱う．すなわち，$MEC(I, n)$ としてクラス集合 $C = \{c_1, c_2, \cdots, c_n\}$ が得られたとき，

$$C' = \{c_i \mid major(c_i) > \alpha \ and \ 1 \leqq i \leqq n\}$$
$$C'' = C - C'$$

とする．C' を MEC の主クラス集合，C'' をその MEC の持つその他のクラス集合とする．図 4 の MEC は図 5 のように簡略化されたグラフで表示できる．

4.2　最小包括クラスを用いた検索支援

キーワードベースのエンティティ検索では，ユーザの入力語に曖昧性がある場合がある．これに対して，検索において入力するべき語が不確定であるという場合と混在する Web 検索結果から求めるものを探すという場合の 2 つの対策が考えられる．

前者は Web 検索の入力前にユーザが検索したい対象の語を確定することであり，その詳細については，4.2.2 節で説明する．

後者は入力後の結果をユーザにわかりやすく提示する必要がある．これを結果閲覧の支援として，4.2.3 節の検索結果のクラスタリング手法が考えられる．

4.2.1 検索語の曖昧性の解消

例えば，ユーザははっきりとした名前は思い出せないが，‘Washington’ という名のついた地名を検索したいとする．そのようなとき，その名前を確定したい．つまり，入力語 ‘Washington’ からユーザの求めるエンティティ（インスタンス）を確定する．

一般的に，ユーザはインスタンス ins_u を求めていて，検索語としてそのキーワードである k を入力すると，k をキーワードとするインスタンスの候補集合 I は，ユーザの求めるインスタンス ins_u を含んでいる．何千にもなる候補から目的のインスタンスを探し出すのは困難であるので，ユーザが自分の求めるインスタンスがどのようなカテゴリに含まれるのかを意識することで，候補集合を絞っていく．

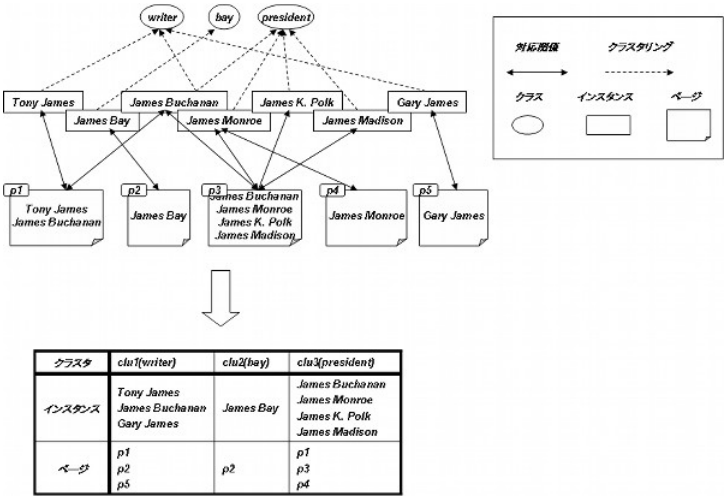
$\exists c_u \in MEC(I,n)$ に対して，$c_u \in type(ins_u) \cup supClassOf*(type(ins_u))$ が成り立つ．ユーザはその c_u を選ぶことで，I より小さい新たな候補集合 $I' = instance(c_u)$ が得られる．I を I' に入れ替えてこれを繰り返すことで，ユーザは候補インスタンス集合を絞り込める．最終的に小さくなった候補集合から求めていたインスタンス ins_u を探し出すことで，曖昧性を解消する．その手順を以下に示す．

1. ユーザはキーワードを入力する．
2. ユーザの入力キーワードを含んでいるインスタンスを YAGO から全て求めて，そのインスタンス集合を候補集合 I とする．
3. a. I から $MEC(I,n)$ を計算して，簡略化する．簡略化された MEC での主クラス集合は，それぞれのクラスをメジャー度に応じて，インスタンス集合をリストなどを用いて提示する．その他のクラスについては，テーブルなどにしてインスタンスとクラスの対応表を提示する．ユーザはこの簡略化された MEC の主クラス集合の数を任意に決めることができる．例えば，スライダーなどで変化させる．ただし，MEC が存在しない数字は選択できない．
   b. ユーザは提示されたリストやテーブルなどから求めるインスタンス ins_u を探して選択する．
   c. b. で候補集合が多くて ins_u が見つからない場合，クラス c を選択することで，$I' = instance(c)$ として，I' を新たな候補集合とする．すなわち $I = I'$ とする
4. 3.b. で ins_u を選択するまで，3 を繰り返す．

上の例で，求めているものが ‘Washington, Kansas’ であるとする．ユーザは ‘location’ や ‘area’ というようなクラスをたどりながら候補集合を絞っていくことで，自分の求めているものを見つけやすくなる．

4.2.2 検索結果のクラスタリング

例えば，ユーザは ‘james’ という名前の米国大統領について調べているとする．しかし，‘james president’ というキーワードによる検索では，その結果は大統領だけでなく，会社の社長など関係ないようなものまで混在しているので，それらを分類したい．



クラスタ	clu1(writer)	clu2(bay)	clu3(president)
インスタンス	Tony James James Buchanan Gary James	James Bay	James Buchanan James Monroe James K. Polk James Madison
ページ	p1 p2 p5	p2	p1 p3 p4

図 6　検索結果のクラスタリングの例

ここで，キーワードの Web 検索結果のページ集合を $P = \{p_1, p_2, \cdots, p_m\}$ とする．p_i は Web ページである．

ユーザが検索キーワードを入力すると，それを含むインスタンスの集合 I とそのページ集合 P が得られる．この 2 つの集合 I, P のメンバの対応関係を求める．つまり，p（P のメンバ）に ins（I のメンバ）が含まれていれば，p は ins に対応するとする．また，$MEC(I,n)$ により，I は n 個のクラスタに分けられる．これに対応して，P も $n(+1)$ 個のクラスタに分割できる．手順を以下に示す．

1. ユーザがキーワードを入力する．
2. 検索エンジンからキーワードの検索結果 $P = \{p_1, p_2, \cdots, p_m\}$ が返される．
3. YAGO から，キーワードを含むインスタンス集合 $I = \{ins_1, ins_2, \cdots, ins_l\}$ が得られる．
4. I から $MEC(I,n) = \{c_1, c_2, \cdots, c_n\}$ を求め，簡略化する．ユーザはこの簡略化された MEC の主クラス集合 $C = \{c'_1, c'_2, \cdots, c'_{n'}\}$ の数を任意に決定することができる．例えば，スライダーなどで変化させる．ただし，MEC が存在しない数字は選択できない．
5. p_i と ins_j の対応関係を求める．$(1 \leqq i \leqq m,\ 1 \leqq j \leqq l)$
6. I を n' 個のクラスタ $clu_1, clu_2, \cdots, clu_{n'}$ に振り分ける．$c_k \in type(ins_j) \cup supClassOf(type(ins_j))$ であれば，インスタンス ins_j はクラスタ c_k に属する．$(1 \leqq k \leqq n')$
7. ページ p_i は対応するインスタンスの属しているクラスタ全てに属するとする．対応するインスタンスが存在しない場合，その他として，クラスタ $clu_{n'+1}$ に入れる．これで，P が $n'(+1)$ 個のクラスタにクラスタリングされる．

上の例で図 6 のように，インスタンス集合が $n = 3$ で，クラス $c_1 = writer$, クラス $c_2 = bay$, クラス $c_3 = president$ と分けられているとする．ページ p_1 はインスタンス ‘Tony James’, ‘James Buchanan’ と対応しているので，クラスタ clu_1，clu_3 に振り分けられる．ページ p_2 がインスタンス ‘James Bay’ と対応しているので，クラスタに clu_2 に振り分けられる．ページ p_3

がインスタンス‘James Buchanan’，‘James K. Polk’，‘James Monroe’,‘James Madison’ と対応しているので，クラスタに clu_1, clu_3 に振り分けられる．

我々は，文書や検索結果を直接クラスタリングするのではなく，知識ベースにあるインスタンスを MEC 用いて分類したうえ，インスタンスと文書の対応関係を利用して行う．つまり，参照型クラスタリング手法を提案している．

5. プロトタイプシステム

４章で説明した手法を実装し，その手法が効果的に検索を支援していることを検証するために，プロトタイプシステムを構築中である．ここでは我々の開発中のプロトタイプシステムの構成について説明する。

システムは，図７に示されているように，候補インスタンス集合の取得，汎化構造処理，MEC 作成，曖昧性解消処理，クラスタリング処理，曖昧性解消出力そしてクラスタリング出力から構成される．

- 候補インスタンス集合の取得
  YAGO から，ユーザが入力する検索キーワードを含むインスタンスを全て，候補インスタンス集合として取得する．
- 汎化構造処理
  YAGO から，候補インスタンス集合のインスタンスと‘type’ 関係であるクラスを取得してクラス集合を得る．クラス集合のクラスと ‘subClassOf’ 関係であるクラスを取得してクラス集合に追加し，これを繰り返す．これらのインスタンス集合とクラス集合とそれらの関係から汎化構造を構成する．
- MEC 作成
  候補インスタンス集合と汎化構造から，MEC を作成する．メジャー度がある数 α 以上のクラスをメジャーな主クラス集合とし，それ以外をその他のマイナーなクラス集合とすることで，MEC の表示を簡略化する．そして重複度，標準偏差を計算して複数得られた結果から一つに絞る．
- 曖昧性解消処理
  簡略化された MEC を提示して，ユーザがインスタンスが見つからずクラスを選んだ場合に，それに対応した新たな候補インスタンス集合を作る．インスタンスを見つけた場合そのインスタンスの検索結果ページ集合を求める．

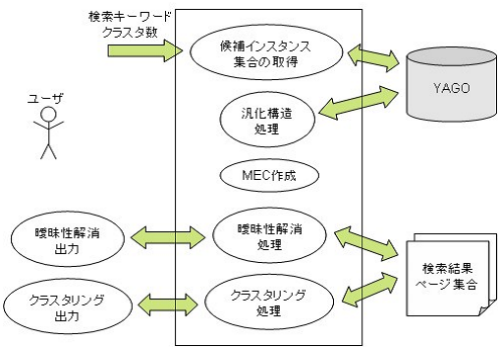


図 7　モジュール図

- 曖昧性解消出力
  簡略化された MEC の主クラス集合のそれぞれのクラスで表現されているインスタンス一覧とその他のクラス集合のクラスの持つインスタンスのテーブルを表示する．インスタンス一覧はそのクラスのメジャー度に応じて大きくする．ユーザにインスタンスかクラスを選ばせる．インスタンスが選ばれた場合，それをキーワードとした検索結果を表示する．クラスが選ばれた場合，また新たな MEC のクラスとインスタンスを表示する．
- クラスタリング処理
  検索キーワードの検索結果ページ集合を求める．簡略化された MEC を用いて，入力されたクラスタ数だけ候補インスタンス集合をクラスタに振り分ける．そして，インスタンスに対応するページを同様に振り分けることで，クラスタリングを行う．
- クラスタリング出力
  検索キーワードの検索結果ページをクラスタリングして表示する．クラスタリングの粒度を細かくしたい場合，スライダーなどの操作によってクラスタ数を変化させる．

次に曖昧性解消におけるシステムの実行例における実行画面の説明する．

図 8 はシステムの実行画面である．①はキーワードの入力ボックスである．図は，①にキーワードとして ‘washington’ という語を入れて隣の start ボタンを押してシステムを実行させた状態である．キーワード ‘washington’ を含んでいる全てのインスタンス I は４つ（分割数３＋ Others）に分類されている．

②のスライダーによって分割数を操作でき，その左の数字は現在の分割数が表示される．例えば，スライダーを３から５に動かすと図 9 のようになる．初期値は３とする..

③のパネルはクラス名を表示し，木構造になっていてクラスの親子関係を表している．一番下には分割数分に分類されたパネルと Others パネルが並び，下にリストを持つ．

④のリストは上のパネルのクラスによって分類されているイ

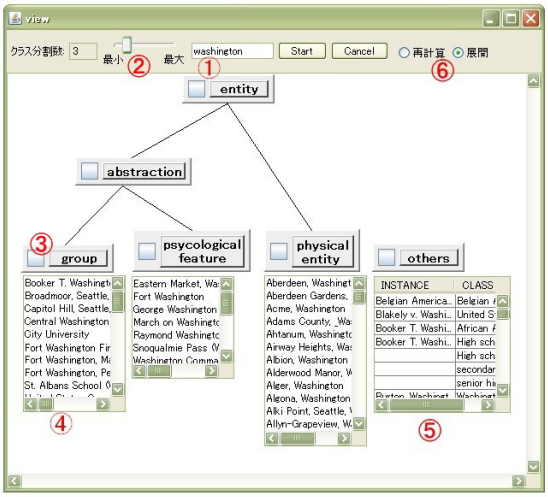


図 8　実行画面図

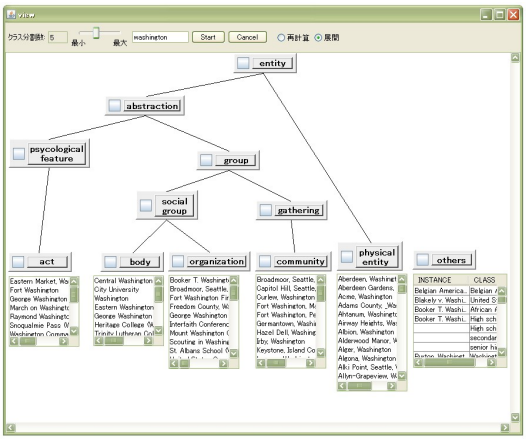


図 9　スライダーによる分割数の変化

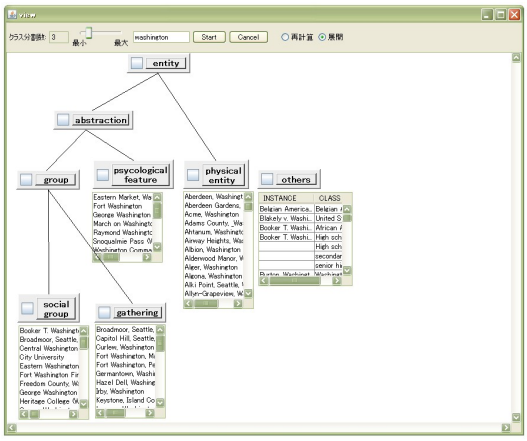


図 10　展　　開

ンスタンスの一覧が表示される．メジャー度の大きさに応じてインスタンスを表示するウィンドウの大きさが決定される．また，このリストのインスタンスはパネルの木構造によって，さらに上の親クラスにも属していることを表す．クラス ‘group’ のインスタンスの場合，上の親クラスである ‘abstraction’，‘entity’ によっても分類されていることを表している．

⑤のリストは MEC の簡略化の際に，‘Others’ クラスに分類されたインスタンスのリストであり，インスタンスとそれに対応するクラスのテーブルが表示される．

③のボタンはそのクラスに分類されているインスタンスの集合を新たな候補集合として再計算し表示するか，そのクラスを展開し，下のクラスを表示するボタンである．⑥はそれを選択するラジオボタンである．例えば，⑥が展開のとき，図 8 の ‘group’ のパネルのボタンが押されると，図 10 のように ‘group’ の下のクラスがとそれらの持つインスタンスのリストを表示する．

このようにして，インスタンスを絞りながら，最終的にリストのインスタンスを選択することで，曖昧性の解消が可能となる．また，マイナー的なものの発見にも効果があると考えられる．

6. 結　　論

本論文では，知識ベース YAGO を利用して，エンティティの汎化構造解析を行い，検索キーワードに対応するエンティティの集合に対して，それを必要最小限のクラスで包含的に表現する方法として，最小包括クラスというものを定義した．そして，そのクラス集合を本研究における検索支援に有効に利用できる計算手法を提案した．そして，それを利用した検索支援として，検索語の曖昧性解消と検索結果ページのクラスタリングの手法を提案し，プロタイプシステムの設計を行った．

今後，プロトタイプシステムを実装して，実験による提案手法の検証を行う予定である．また，MEC を効率よく扱うために更なる定義と検索支援の手法の洗練および上記のプロトタイプシステムのインターフェースの向上が必要であると考える．

謝辞　本研究の一部は，グローバル COE プログラム「知識循環社会のための情報学教育研究拠点」（拠点リーダー：田中克己）による．ここに記して謝意を表します．

文　　献

[1] Tao Cheng, Kevin Chen-Chuan Chang. Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web. Proc. of CIDR2007, pp.108-113, 2007.
[2] Wikipedia: http://en.wikipedia.org/
[3] Fabian M. Suchanek and Gjergji Kasneci and Gerhard Weikum. Yago: A Core of Semantic Knowledge, Proc. of WWW2007, 2007.
[4] WordNet: http://wordnet.princeton.edu/
[5] Auer, S.; Bizer, C.; Lehmann, J.; Kobilarov, G.; Cyganiak, R.; Ives, Z.. DBpedia: A Nucleus for a Web of Open Data. Proc. of ISWC 2007 + ASWC 2007, LNCS4825, 2007.
[6] David Milne, Ian H.Witten and David M.Nichols. A Knowledge-Based Search Engine Powered by Wikipedia. Proc. of CIKM07, pp.445-454, 2007.
[7] Gjergji Kasneci, Fabian M. Suchanek, Georgiana Ifrim, Maya Ramanath and Gerhard Weikum. NAGA: Searching and Ranking Knowledge. Proc. of ICDE2008, 2008.
[8] Quang Minh Vu, Tomonari Masada, Atsuhiro Takasu, and Jun Adachi. Name Disambiguation in Web Search Using Knowledge Base. IPSJ SIG Notes, Vol.2006, No.78(20060713) pp. 185-192, 2006.
[9] S. Tejada, C.A.Knoblock, and S.Minton. Learning Object Identification Rules for Information Integration. Information Systems, Vol 26, No.8, pp 607-633, 2001.
[10] Reema Al-Kamha, David W. Embley. Grouping Search-Engine Returned Citations for Person-Name Queries. Proc. of WIDM04, pp.96-103, 2004
[11] Xiaojun Wan, Janfeng Gao, Mu Li and Binggong Ding. Person Resolution in Persion Search Results: WebHawk. Proc. of CIKM05, pp.163-170, 2005.
[12] R. Bekkerman and A. McCallum. Disambiguating Web paperances of people in a social network. Proc. of COLING-ACL98, pp. 79-85, 1998.
[13] 木村塁，戸田浩之，田中克己. 検索結果スニペットのクラスタリングによる同姓同名人物の特定. DEWS2006 2C-i11, 2006.
[14] Robertson, S.E.: On Term Selection for Query Expansion, Journal of Documentation. Vol.46, No.4, pp.359-364, 1990.
[15] Google サジェスト: http://www.google.com/webhp?complete=1
[16] goo 検索: http://search.goo.ne.jp/
[17] 若木裕美，正田備也，高須淳宏，安達淳. 検索語の曖昧性を解消するキーワードの提示手法, DBSJ Letters Vol.4. No.2, pp.41-44, 2005.
[18] Katsumi Tanaka, Masatoshi Yoshikawa: Towards abstracting complex database objects: Generalization, reduction and unification of set-type objects (extended abstract). LNCS Vol.326, pp.252-266 1988