学位論文 博士（工学）

膨大な文書を対象とした
情報集約データベースに関する研究

2012年3月

慶應義塾大学大学院理工学研究科

富田準二

論文要旨

Web や検索エンジンの進歩によって誰でもが簡単に，所望のページを取得できるようになってきている．しかしながら，例えば，会社や製品の評判や競合他社の動向などは，単一の文書としてまとめて記述されてはいないため，複数の文書を集め，その内容をまとめる作業（情報集約）を行わなければならない．現状，このような情報集約を行うための汎用的な枠組みは確立されていないため，情報集約サービスを実現するためには，個別のアプリケーションプログラムを最初から開発する必要がある．

本研究では，情報集約タスクを実行するための汎用的な枠組みとして，情報集約データベース（IADB: Information Aggregation DataBase）を提案する．IADB では，集約の対象となる情報の断片を，対象物とそれに付随する属性の集合（情報要素タプル）で表現する．例えば，評判情報であれば，“製品 A の画面は美しい”という情報の断片を，<製品 A，画面，美しい，好評>（<対象物，評価属性，評価表現，評価極性>）という情報要素タプルで表現する．IADB は，このような情報要素タプルからなる仮想的な情報要素リレーションを大規模な文書集合から自動的に生成し，そこへの検索と集計を行うことで，様々な情報集約タスクを実行できるようにする．本研究では，特に，IADB を構築するうえでの技術課題として，(a) 文書から対象物を抽出するための辞書の自動構築手法，(b) 事前に抽出した情報要素の属性と，対象物を表す入力キーワードとを用いた情報要素リレーションの動的生成手法，(c) 情報集約に特化した独自の問合せ言語，のそれぞれについて検討し，設計・実現する．

IADB を評判情報の集約を行う実サービスに適用し，情報要素リレーションへの簡易な問合せによって，様々な有用な情報集約結果を取得できることを示す．また，各技術課題に対して提案手法は，(a) に関しては，特に多義語の語義の網羅的な収集に有効であること，(b) に関しては，入力キーワードが未知語であったとしても，実時間で情報要素リレーションを生成できること，(c) に関しては，表記ゆれなどに対応しながら階層的な内訳をもつ集約結果を簡易な記述で取得できることを示す．更に，IADB が，他の情報集約タスクにも適用できる汎用的な枠組みであることを述べる．このように，IADB を用いることで，新商品の評判のような今まですぐには取得できなかった情報を多面的かつ即座に提供するオンラインサービスを，少ないコストで実現できる．

Title

A Study on an Information Aggregation Database for a Large Number of Documents

Abstract

Web and search engines enable us to obtain required documents easily. However, aggregating the fragments of information in a large number of documents is needed for obtaining the information that is not written in a single document such as the repetition of a company or a product, and the strategy of competitors. Since there is no concrete framework for aggregating the fragments of information, developers have to implement a specific program for such an aggregation task.

This research proposes an information aggregation database (IADB) for executing such tasks. In the IADB, an information fragment can be represented as a tuple consisting of a target object and its attributes. For example, for sentiment information, a fragment "The display of Product A is clear." is represented as a tuple "<Product A, display, clear, positive> (<target object, sentiment property, sentiment expression, sentiment orientation>)". The IADB enables to execute such aggregation tasks by automatically generating a virtual relation of the tuples from a large number of documents, and searching and summarizing the relation. This research deals with three technical issues; (a) automatic construction of a dictionary for extracting target objects in documents, (b) online relation generation based on combining pre-extracted attributes and a given keyword, and (c) a query language especially designed for such tasks.

The trial of applying the IADB to a real sentiment analysis service shows that simple queries to the relation can generate effective aggregation results for the service. Evaluations also show (a) is effective for extracting all the meanings of multi-meaning words, (b) allows realtime generation of the relation for an unknown word, and (c) makes a hierarchical result having subtotal of aggregation and handles the same word in different forms easily. Furthermore, this research also shows the IADB is general enough to apply to other tasks. Therefore, the IADB enables developers to realize online services with small amount of effort, which produce multiple views of aggregated information such as the repetition of a new product.

目 次

図 目 次

表 目 次

第1章　序章

Web の進歩によって誰でもが簡単に情報を発信し，世界中のどこからでも発信された情報に瞬時にアクセスできるようになった．また，検索エンジンをはじめとするアクセス技術の進歩によって，ユーザの所望のページを容易に取得できるようになってきている．しかしながら，ユーザの要求がある単一のページに記述された内容では満たされない場合，所望の情報を得るためには多くの労力が必要となる．本研究では，このような複数のページ中に存在する情報を集約するタスクに着目し，このようなタスクを容易に実行できるフレームワークを実現する．

1.1　膨大な文書を対象とした情報集約の必要性

検索エンジンをはじめとする Web へのアクセス技術は，大幅な進歩を遂げた．ユーザの情報要求が，どこかの特定のサイトへの到達を目的とするナビゲーショナルクエリであり，的確なキーワードが指定される場合，Google などの検索エンジンは，かなりの確率で検索結果の上位に所望のサイトを表示できるようになってきている．また，ニュース，ブログ，SNS（Social Network Service）やツイッターなどのサイトや，RSS リーダ（Resource description framework Site Summary reader) の仕組みを用いることによって，ユーザの興味に即した公開されたばかりの情報を集めることができるようになった．更に，キーワードで表すことが難しい質問に対しても，教えて goo や Yahoo 知恵袋などの質問に対する回答をネット上の誰かが行うといったサービスも一般的になってきている．これらの仕組みを用いると，ネット上の誰かが何らかの知識を持ち，それを個別のページに記述できるような場合には，情報のアクセスは比較的容易になってきているといえる．

一方で，ユーザが所望する情報の中には，1 つの文書だけから得られるものばかりではなく，複数の文書に含まれている情報を集約することによって，はじめて得られるものも数多くある．例えば，研究を開始する際にサーベイを行うことを考えると，個別の論文へのアクセス手段は提供されているものの，誰かがサーベイ論文を作成し公開していない限りは，莫大な労力をかけて関連する論文を集め，これらをまとめる必要がある．また，自社製品の評判を知りたい場合，評判情報の発信者は不特定多数のネット上のユーザであるため，製品名をキーワードとして検索した後に，検索結果のページを 1 つずつ確認して何らかの集計を行う必要がある．この他にも，例えば，“2020 年にスマートフォンの販売台数は x 倍になる” などの将来を予測した記事を収集し，これらをまとめることで，指定したキーワードに関する将来像を俯瞰するといった将来情報の集約に関する研究も行われている [29].

上記に述べたものはいずれも，ネット上にすでに存在するページには，まだ，記述されていない何らかの知識を，複数のページをまとめることによって，生成しているといえる．

このように，膨大なネット上の文書の中から，ユーザの所望の情報を最適な形式に集約することは，新しい知識や知見を発見するためのタスクであり，非常に重要なものである．本研究では，このようなタスクを情報集約タスクと呼び，情報集約タスクの自動化を行うための枠組みの構築を課題とする．ここで，情報集約は，テキストマイニングと似た概念であるが，テキストマイニングという用語は，様々な文脈において多様な意味で用いられているため，その定義が曖昧である．そのため，本研究では，“複数の文書に記述された情報をまとめる” という点に特に焦点を絞り，情報集約という用語を用いる．

1.2　情報集約タスクの例

ある企業が自社の製品をリニューアルした際に，Web 上からその評判を調査するタスクを考える．製品の評判は，レビューサイトやブログなど様々なサイトに記述されていることが想定される．この場合，企業の分析者は，例えば次の作業を行う．

(1) 製品名をキーワードとして Web 検索を行う．

(2) 検索結果の中からレビューらしいものを探す．

(3) 各レビュー中の評判に関する用語を集計する．

(4) 各用語が好評か不評か，どのような属性（色や形など）について書かれているのかを分類する．

(5) 著者の性別や年代ごとに，用語を集計する．

(6) リニューアルの前後で書き込み件数などに変化があったかどうかを調べる．

(7) (3)〜(6) に関してクロス集計を行う．

このように，分析者は，検索を行うことで関連文書を収集し，その後，所望の形式に文書中の情報の断片を集計する操作を行っている．また，これらの集計のための作業は，その時々によって異なり，その組合せも様々である．

1.3　情報集約フレームワークの必要性

1.2 節で述べたような情報集約タスクを自動的に行うためには，文書に書かれている内容をコンピュータが扱える表現に変換し，このような表現に対して，必要な集計操作を組み合わせて実行できる必要がある．しかしながら，現状，このようなタスク全般に利用できるようなフレームワークは存在しないため，個別のアプリケーションプログラムをアドホックな方法で開発するしかない．

一方，形式化されたデータに対しては，RDB（Relational Data Base）が利用でき，RDB では，様々な用途にデータの加工ができるので，アプリケーションプログラムは少ない工数で開発できる．このように，大規模文書データ中の情報を対象とした RDB のような汎用的な枠組みを構築すれば，各情報集約タスクを実行するシステムを，少ない工数で構築できると考えられる．

1.4　本研究の目的

本研究では，情報集約タスクを実行するための汎用的な枠組みに関して検討を行う．このような枠組みは，次の要件を満たす必要があると考えている．

要件 (1) Web 上に公開されている自然言語で記述された大量の文書情報を対象とできること

要件 (2) 文書に含まれる情報の断片を対象として，問合せに応じて複数の操作を組み合せて実行し，情報集約結果を即座に生成できること

要件 (3) 様々な情報集約タスクに対して，少ない開発コストで適用できること

要件 (1) に関して，文書情報を特に対象としているのは，情報発信者側には，追加の負担を掛けることなしに，情報発信者とは別の分析者が，情報集約タスクを実行できるようにするためである．例えば，評判などの情報を考えた場合，記事の著者は，自分の記事を発信し，共有することが目的であって，特に，その記事が情報集約といった別の用途で利活用されることを望んでいるわけではない．このような場合，情報発信者側が集約のために労力を掛け，何らかのメタデータを付与することは考え難い．このように，情報発信者側に何らかの負担をかけると，集約される情報の種類が限られてしまう．一方，文書情報を対象とすることができれば，情報発信者に集約のメリットが少ないような情報に対しても情報集約タスクが実行できるようになる．また，情報量が少ないと事実とは異なる結果を抽出してしまう恐れがあるので，対象となるドメインの文書データを網羅的に処理できることが重要である．

要件 (2) に関して，1.2 節で述べたように，情報集約タスクにおける情報の断片に対する操作は，その時々によって必要なものが異なり，また，その組合せも様々である．そのため，ある情報集約タスクにおける操作の組合せを柔軟に変更できる必要がある．したがって，RDB などと同様に，問合せ言語によって，各種の集約結果を取得できることが重要である．また，情報集約サービスは，現状の検索サービスと同様に，一般ユーザ向けのオンラインサービスとして提供するニーズもあると考えている．そのため，このようなニーズを考えると問合せを実行した際に，即座に集約結果を生成できる必要がある．

要件 (3) に関して，ある情報集約タスク用に個別のアプリケーションプログラムを最初から構築するのでは，各情報集約タスクに対応するために多くの工数がかかる．その結果，このような情報集約サービスが普及するのは難しくなると考えている．したがって，各情報集約タスクに依存する処理と依存しない処理を切り分け，依存しない処理を RDB のような汎用的な機能によって提供できることが重要である．

本研究では，上記の要件を満たすものとして，情報集約データベース（IADB: Information Aggregation DataBase）を提案，設計・実現する．IADB では，集約の対象となる情報の断片を，対象物とそれに付随する属性の集合とで表現する．ここでは，この表現を情報要素と呼ぶ．例えば，評判情報であれば，“製品 A の画面は美しい” という情報の断片を，

<製品 A，画面，美しい，好評>（<対象物，評価属性，評価表現，評価極性>）

という情報要素で表現する．IADB は，このような情報要素のタプル（情報要素タプル）からなる仮想的な情報要素リレーションを大規模な文書データから自動的に生成し，そこ

への検索と集計を行うことで，様々な情報集約タスクを実行できるようにする．これらの機能によって，要件 (1) と要件 (2) を満たすようにする．一方，各情報集約タスクに容易に適用できるように，タスク間で共通する機能とタスクに依存する機能とを分離する．そして，タスクに依存する機能を，外部関数で容易に組み込めるアーキテクチャとすることによって，要件 (3) を満たすようにする．

本研究では，IADB を構築するうえでの技術課題を明らかにし，その解決方法を示す．また，評判情報を集約する実サービスに IADB を適用し，要件 (1)〜(3) が実際にどのように満たされたのかを検証する．

1.5　論文の構成

本論文の次章以降の構成は次のとおりである．

第 2 章 関連研究
各研究分野の要素技術を，(i) テキスト構造化，(ii) テキスト操作・分析，(iii) 可視化・要約，の観点で概観するとともに，情報集約のためのフレームワークの必要性と関連研究の問題点について述べる．

第 3 章 情報集約モデルと情報集約データベース
1.4 節で述べた情報要素の妥当性と情報集約タスクを実行するためのモデルの詳細について述べる．次に，このモデルに基づく IADB の基本アーキテクチャを提案し，その実現のためには，次の技術課題があることを述べる．

技術課題 1: 文書中の情報の断片からの情報要素タプルの自動生成

技術課題 2: 集約処理を実行するための問合せ言語

第 4 章 固有表現辞書の自動構築
技術課題 1 の解決には，文書中の対象物を高精度に抽出できる必要があり，そのためには，固有表現を網羅的に集めた辞書が有効であることを述べる．次に，固有表現辞書への用語の自動登録を実現するために，未知語が与えられたときに，その表記が対象となる固有表現のクラスに属するかどうかを自動判定する手法を提案する．特に，ある表記が多義性をもつ場合に，提案手法は使用頻度の少ない語義の網羅的な収集に有効であることを示す．

第 5 章 動的なリレーション生成
固有表現辞書の自動構築手法が実現できたとしても，新たな固有表現は日々生まれるので，文書中の全ての対象物を完全に抽出することはできない．一方，多くの情報集約タスクでは対象物を表すキーワードがユーザから入力される．そこで，入力キーワードを利用することで，文書中から対象物を抽出し，情報要素タプルを生成する手法を提案する．特に，提案手法では，事前処理で情報要素タプルの一部を生成しておくことで，入力キーワードが未知語であったとしても，実時間で情報要素リレーションを生成できることを示す．

第 6 章 情報集約言語
技術課題 2 に関して，情報集約のための問合せ言語がもつべき要件として，SQL（Structured Query Language）では実現が困難な，(a) 階層的な内訳をもつ集約結果の生成，(b) 表記ゆれなどに対応した柔軟な集計，を挙げる．次に，これらの要件を満たす問合せ言語を提案する．特に，提案言語では，類義語のグループ化といった柔軟な基準での集計を行いながら，階層的な内訳をもつ集約結果を簡易な記述で生成できることを示す．

第 7 章 情報集約データベースの実現と評価
IADB のアーキテクチャの詳細と，IADB を用いて情報集約サービスを構築する事例を示す．特に，評判情報の集約サービスをポータルサイト goo* 上の実サービスとして提供し，この結果をもとに，IADB が，1.4 節で述べた 3 つの要件を満たすものであることを述べる．

第 8 章 結論
まとめとして，IADB を実現するうえでの各技術課題が，本研究においてどのように解決されたのかを述べる．最後に，文書情報以外の情報集約への拡張などについて今後の展望を述べる．

*http://www.goo.ne.jp/

第2章　関連研究

文書データを対象とした情報集約は，複数の技術の融合であるため，様々な研究分野で議論がされている．本章では，まず，関連する研究分野の動向を述べる．次に，個々の要素技術を，(i) テキスト構造化，(ii) テキスト操作・分析，(iii) 可視化・要約，の 3 つに分類し概観する．更に，個々の要素技術を統合して利用できるフレームワークの必要性について述べ，関連研究の問題点を明らかにする．

2.1　研究分野の動向

文書データを対象とした情報集約技術は，情報検索，データマイニング，自然言語処理などの分野で活発な議論が行われている．また，文書データを特に対象とはしていないが，Web 上のデータの連携利用に関する議論がセマンティック Web の分野で行われている．ここでは，各研究分野におけるアプローチの概要と関連する国際会議を挙げる．本章で示した国際会議などとそれらの URL の一覧を，表 2.1 に示す．

2.1.1　情報検索

情報検索の元々の定義は，文献 [55] に，“Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.” とあるように情報の構造化や分析などを含む非常に幅広い概念であり，本研究における情報集約そのものである．しかしながら，90 年代半ばまでは，文書を検索式との関連性 (relevance) に応じてランクづけする手法 (文書検索) が最もホットなトピックであり TREC (Text Retrieval Conference) を中心に様々な手法が試された．一方，ACM (Association for Computing Machinery) の主催するこの分野の代表的な国際会議である ACM SIGIR (ACM Special Interest Group on Information Retrieval) では，分類，情報抽出，質問応答 (QA：Quetion Answering)，TDT (Topic Detection and Tracking) などの様々な情報集約処理に関連するセッションが行われている．また，アジアを中心とした NTCIR (情報検索システム評価用テストコレクション構築プロジェクト) では，純粋な文書検索だけでなく，特許分析や要約などのタスクも行われている．このように，単純な文書検索から，情報を構造化，分析する手法まで幅広いトピックが情報検索分野の対象である．この他，CIKM (ACM Conference on Information and Knowledge Management) や，デジタルライブラリ関連の JCDL (Joint Conference on Digital Libraries) などで幅広く議論されている．また，文献 [3] には情報検索分野の課題が数多く述べられている．

表 2.1 情報集約に関連する国際会議など

会議略称	正式名称	URL
TREC	Text Retrieval Conference	http://trec.nist.gov/
SIGIR	ACM Special Interest Group on Information Retrieval	http://www.sigir.org/
NTCIR	情報検索システム評価用テストコレクション構築プロジェクト	http://research.nii.ac.jp/ntcir/index-ja.html
CIKM	ACM Conference on Information and Knowledge Management	http://www.cikm2011.org/
JCDL	Joint Conference on Digital Libraries	http://www.jcdl.org/
SIGKDD	ACM Special Interest Group on Knowledge Discovery and Data Mining	http://www.sigkdd.org/
ECML PKDD	The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases	http://www.ecmlpkdd2011.org/
PAKDD	Pacific-Asia Conference on Knowledge Discovery and Data Mining	http://pakdd2012.pakdd.org/
ICDM	IEEE International Conference on Data Mining	http://www.cs.uvm.edu/ icdm/
ACL	The Association for Computational Linguistics	http://www.aclweb.org/
EACL	The European Chapter of the ACL	http://www.eacl.org/
EMNLP	Conference on Empirical Methods in Natural Language Processing	http://conferences.inf.ed.ac.uk/emnlp2011/
COLING	International Conference on Computational Linguistics	http://www.coling-2010.org/
WWW	The World Wide Web Conference	http://www.iw3c2.org/
ISWC	International Semantic Web Conference	http://iswc2011.semanticweb.org/
ESWC	Extended Semantic Web Conference	http://www.eswc2011.org/
ASWC	Asian Semantic Web Conference	http://www.aswc2009.org/

2.1.2　データマイニング

データマイニングの代表的な手法には，決定木，ニューラルネットワーク，回帰分析，クラスタリング，マーケットバスケット分析などがある [6]．これらの手法は，従来，主に構造化された数値データやカテゴリデータなどを対象としていた．しかしながら，この分野の代表的な国際会議である SIGKDD（ACM Special Interest Group on Knowledge Discovery and Data Mining）では，上記に述べたような基本的な手法に加えて，テキストデータを対象としたマイニングが取り扱われるようになってきている．データマイニングの研究分野では，マイニング手法とともに，いかにきれいなデータをマイニング手法の入力とするのかといった，データの前処理が重要な課題とされている [28]．人手で作成されたテキストは形式や用語が統一されていないため，このようなデータの前処理手法はますます重要になるものと思われる．この他，ヨーロッパを中心とした ECML PKDD（The European Conference on

Machine Learning and Principles and Practice of Knowledge Discovery in Databases），アジアを中心としたPAKDD（Pacific-Asia Conference on Knowledge Discovery and Data Mining），IEEE （The Institute of Electrical and Electronics Engineers）の主催するICDM （IEEE International Conference on Data Mining）などで幅広い議論が行われている．

2.1.3　自然言語処理

テキストは人間が読める言語で記述されているので，当然，自然言語処理は最も重要な技術の１つである．自然言語処理の主な対象領域は，従来，対話処理や翻訳が多かった．しかしながら，この分野の代表的な国際会議であるACL（The Association for Computational Linguistics）では，情報検索関連の応用も多く想定されるようになってきている．また，構文解析，意味解析，単語の曖昧性解消，辞書の自動生成なども現在なおホットトピックである．対話処理や翻訳は非常に難しい課題であるが，この課題を解決するために長年研究が行われてきた手法の多くは，そのまま情報集約技術に応用することができる．自然言語処理技術の今後の発展と応用が，情報集約技術を飛躍的に向上させるものであると考えられる．この他，COLING（International Conference on Computational Linguistics），EACL（The European Chapter of the ACL），EMNLP（Conference on Empirical Methods in Natural Language Processing）などで幅広い議論が行われている．

2.1.4　セマンティック Web

セマンティックWebは，従来のWebが人間が読む文書情報を対象としていたのに対して，コンピュータ処理できるデータを公開することで，各種知識を連携利用させるという考え方である．近年，この分野で，Linked Data[8] が注目されている．Linked Dataは，セマンティックWebの分野で研究されてきた技術の上に成り立っている．ただ，セマンティックWebの従来研究では，主にオントロジによってデータを統制的に制御することに重点がおかれてきた．一方，Linked Dataでは，統制的な側面をひとまずおいておき，個別情報（インスタンス）を中心として考え，データセットをWeb上で共有する仕組みを作ることに重点がおかれている [64][65]．このようにデータを共有することで，事実的な知識の検索や，複数の情報源の情報を利用した推論などの各種サービスが実現できる可能性がある．

Linked Dataは，欧米を中心に，New York Times，BBC（The British Broadcasting Corporation）などのメディア [50]，W3C（The World Wide Web Consortium ）のLinking Open Drug Dataに代表される医薬品関連 [23]，各国政府や図書館の情報 [57]，地理情報 [61] などの公開が急速に進んできている．また，Wikipediaの表形式の情報をオープンデータ化するDBPedia*などの取組みが活発に行われるようになってきている．このように，いくつかの分野では非常に有効なアプローチであるが，データが公開されるためには，情報発信者側へのメリットが欠かせない．そのため，一般のユーザの記述した評判情報の集約など，本研究が想定している自然言語で記述された文書情報の集約タスクとはその主なター

*http://dbpedia.org/

ゲット分野が異なる．しかしながら，情報の形式的な表現方法など，フレームワークを構築するうえで参考になるものも多い．セマンティック Web に関しては，ISWC（International Semantic Web Conference），ESWC（Extended Semantic Web Conference），ASWC（Asian Semantic Web Conference）などで幅広い議論が行われている．

2.1.5　各研究分野の動向のまとめ

以上のように，情報集約技術は，自然言語処理をベースとし，データマイニング手法を適用し，情報検索の本来の目的を達成する複合技術であるといえる．また，フレームワークを構築するための情報の形式的な表現方法などには，セマンティック Web の分野での研究成果が活用できる．この他にも，Web 上のデータを中心に扱う国際会議 WWW（The World Wide Web Conference）や Web マイニングのサーベイ文献 [36] などが関係が深い．国内では，人工知能学会誌 ‘特集:テキストマイニング’[74]，毎年発行される電子情報技術産業協会の ‘ヒューマンインターフェース技術に関する調査報告書’[73]，文献 [54] などが関連が深い．

2.2　情報集約を実現する技術

情報集約を実現するためには，様々な要素技術が必要である．本節では，(1）テキスト構造化，(2）テキスト操作・分析，(3）可視化・要約，に分けて要素技術を概観する．

2.2.1　テキスト構造化

単なる文字コードの列であるテキストは，まず，コンピュータ内で意味や内容を取り扱うことができる形式へと変換する必要がある．ここでは，この処理をテキスト構造化と呼び，特に語句抽出，語句間の関係抽出，文書の内容表現生成について述べる．

(1) **語句抽出**
最も基本的なテキスト構造化処理は，語句（term）抽出である．日本語に対しては，形態素解析処理によって，入力テキストを形態素（言語学的に意味をもつ最小の単位）に分割し品詞の決定，原型への変換を行う [43]．次に名詞などの特定品詞の形態素を抽出し，これを語句として利用することが多い．英語においては，stemming 処理によって語幹抽出を行い，単純に不要語を除いたものを語句として使用する場合が多い [15]．ただし，英語においても品詞付け処理（part-of-speech tagging）が，語句抽出に利用されることもある．形態素解析を行うフリーソフトには chasen[†] など，いくつかが公開されている．英語の stemming アルゴリズムには Porter のアルゴリズム，425 語からなる不要語リスト（stoplists）がしばしば利用され，いずれも文献 [15] で紹介されている．

語句は，単純な単語でなくても良く，名詞句や固有表現を単位とすることもできる．ここで，固有表現とは，‘日付’，‘組織名’，‘場所’，‘製品名’ といった特定のクラスに

[†] http://chasen-legacy.sourceforge.jp/

属する文字列である．名詞句の抽出方法には，品詞列の出現パターンを指定する方法 [26]，固有表現の抽出手法には，ルールの学習による手法 [27] などがある．特に固有表現抽出を用いると，例えば，“ある事件が起った日はいつか?” といった‘日付’が回答となる質問に答えることや，‘組織名’や‘場所名’に紐づく情報の集計を行うことができる．固有表現抽出は，従来，数個のクラスを対象としたものであるが，‘製品名’といった広い概念のクラスではなく，‘車名’や‘曲名’といった詳細なクラスを同定できれば，更に有効な情報集約を実現できると考えられる．しかしながら，細かい粒度の固有表現に対応するためには，学習データの準備，ルールの整備など多くの課題があり，半教師あり学習や辞書の自動構築など，多くの研究が現在も行われている [52, 18, 31, 51, 13].

(2) **語句間の関係抽出**

語句間の関係を抽出する最も基本的な技術は，係り受け解析である [43]．係り受け解析では，ある単語 (や文節) に対して関係する単語 (や文節) を抽出することができる．例えば，ある製品に対して‘画面–きれい’とか‘Web–検索’といった‘主語–述語’や‘目的語–述語’の関係を取得できる．完全な文全体の構文解析は曖昧性が大きく難しい課題である．ただ，ある 2 単語の関係だけに着目すれば，かなりの精度で係り受け解析を行うことができ，これらを利用した手法は今後増えていくものと思われる．現在利用できるフリーの係り受け解析ツールとして，KNP[‡]が公開されている．

係り受け解析では，主に，語句間の構文的な繋がりを抽出することを目的としているが，複数の固有表現間や，固有表現と評判情報といった意味的な関係性の抽出を目指した研究も数多く行われている [21, 1, 9, 68]．例えば，このような関係の例として，会社名と社長や，国名と首都などの抽出がある．これらの技術を用いると，ある対象物に対して，関係する属性を付与できるので，文書中の情報の構造化に有効である．

(3) **文書全体に対する内容表現生成**

文書を類似検索したり，分類したりする場合には，その内容をコンピュータで扱うことができる表現に写像する必要がある．ここでは，このような表現を文書の内容表現と呼ぶ．内容表現として最もよく利用されるのは，タームベクトル（term vector）である．タームベクトルでは，文書から抽出された語句を，その出現頻度などを用いて重み付けし，語句を次元とするベクトルによって内容を表現する．語句の重要度計算には，様々なものがあるが TF*IDF 法が最もよく利用される [15]．TF*IDF 法では，対象とする文書に多数回出現し（TF: term frequency），その語句を含む文書数が少ない（IDF: inverted document frequency）語句に対して高い重要度を与える．また，最近の研究では，確率モデルに基づく Okapi BM25 という手法が TF*IDF 法よりも検索精度が高いことが示されている [53]．TF*IDF 法や Okapi BM25 法は，主に検索を目的とするものであるが，このような重み付け手法は必ずしも全てのテキスト操作に対して普遍的なものではなく，分類の場合には相互情報量 [44] などが利用されることもある．また，要約生成の基礎データに利用したり，より精度の高い検索を行ったりするためには，文書全体の内容を表す 1 つのタームベクトルを生

[‡]http://nlp.ist.i.kyoto-u.ac.jp/index.php

成するのではなく，パッセージや文といったより短い単位に対してこのようなタームベクトルを生成することもある [20, 56].

全ての語句を次元とするタームベクトルでは，語句同士の直交性を仮定している．すなわち，似た語句が出現しても似た内容の文書とは見なされないことを意味する．この問題の解決を目指すものとして，LSI（Latent Semantic Indexing）と呼ばれる文書単語行列を固有値分解することによって直交する次元に分解する手法 [12] や，概念ベース [30] や意味の数学モデル [33] のような語句自体をあらかじめベクトルで表現する手法がある．これらの手法は，似た語句を含む文書を検索できるので再現率[§]の向上につながるが，逆に似た語句の細かい意味の違いを考慮することができず適合率[¶]の低下につながる可能性もある．

このような，ベクトルによる内容表現は非常にシンプルであり検索，分類を中心とした幅広いシステムで利用されているが，テキストを語句の集合（bag-of-words）で表しているため，文書内での語句の出現位置や語句同士の関係は全く考慮されていないという問題がある．ベクトル以外の内容表現として，語句間の関係に注目して単語をノード，単語間の関係や関連をリンクとしたグラフによって文書を表現する Conceptual Graphs[62] 手法がある．Conceptual Graphs は，各文を意味解析し単語をノード，単語間の関係（agent, object など）をリンクとしたグラフによって文書の内容を表現する．文献 [42, 41] で実際に検索やマイニングへの適用が試みられているが，意味解析などの深い言語処理を必要とするため幅広い分野への適用が難しい．一方，筆者は，単語の重要度をノードの重み，単語間の関連度をリンクの重みとした主題グラフ (Subject Graphs)[70] によって，文書の内容を表現する手法を提案した．主題グラフは，深い言語処理を必要としないために適用範囲は広いが，Conceptual Graphs ほどの高度なテキスト操作を実現することができない．より洗練された内容表現は，高度なテキスト操作を実現できるが，その適用範囲が絞られるといったトレードオフの関係にある．

2.2.2　テキスト操作・分析

テキスト構造化によって，テキストをコンピュータ上で扱える表現に変換することができれば，様々な数学的な手法によって有用なテキスト操作を実現することができる．この場合のテキスト操作には，大きく分けて，(1) 抽出された語句や語句間の関係を単位として扱い集計を行うもの，(2) 各文書を単位として扱い検索や分類を行うもの，がある．

(1) **語句や語句間の関係の集計操作**
最も単純な集計操作は，語句の出現頻度をある属性値をもつものや時系列で集計するものなどである．この処理は非常に単純に見えるが，使用される語句に対する何らかの正規化処理を行わなければ所望の結果が得られない場合がある．例えば，‘PC’ と ‘パソコン’ といった表記ゆれの問題や，‘ネットワーク’ と ‘インターネット’ といった同義語，類義語の問題を解決しなければならない．

[§]正解集合の中で検索されたものの割合．もれの少なさを表す．
[¶]検索された集合の中で正解の割合．外れの少なさを表す．

語句間の関係性を利用したシステムとして，日本 IBM の TAKMI（Text Analysis and Knowledge Mining）[‖] がある．TAKMI は，‘メモリ–増設する’ といった係り受け関係を抽出し，どの程度の頻度で文書に出現するのかを集計することができる．実際にコールセンタへの問合せに適用しその有効性を検証している [45]．また，この他にも関係を利用したものとして単語間の相関ルールの抽出を行うものがある．Feldman らが提案した FACT システムがその一例である [14]．FACT は，テキスト，キーワード同士の関係，質問を入力として与えると，質問を満たすような（[条件部]->[結論部]）といった関連をテキストの中から抽出する．

(2) **文書の内容表現に対する操作**
タームベクトルを用いると，類似文書検索は，これらのベクトルの内積や cosine で実現される [66]．文書分類の場合も同様にタームベクトルを特徴ベクトルとして捉え，パターン認識のアルゴリズムを適用することで実現される．分類アルゴリズムには，Naive Bayes や SVM（Support Vector Machine）のようなルールを用いない手法と，決定木のようなルールを用いる手法がある [44]．前者は精度が高いが得られた結果が人間にはわかり難く，後者は人間に分かりやすいルールが得られるが分類精度が低いという問題がある．山根らは人間に分かりやすいということが非常に重要であるとし，ルールを用いながら分類精度を向上させる手法を提案している [76]．

上記に述べた手法は，その用途に応じて主に個別に研究されてきたが，ある条件で検索した結果を分類し，分類カテゴリごとに語句や語句間の関係を集計するといったように，組み合わせて利用できることが重要である．

2.2.3　可視化・要約技術

テキスト操作・分析の結果を最終的に判断するのは人間であるため，可視化技術は情報集約の主要なテーマの 1 つである．可視化手法では，文書をある表現に構造化し，その表現に対する何らかの集計を行ったものを分かりやすく提示することを目的としてる．可視化手法には，次に述べるように，語句や語句間の関連の可視化，文書集合の概観を表示するものなどがある．また，自動要約も出力が文章形式である可視化の一種とみなす．

(1) **語句・関連の出現頻度表示**
テキストデータに対する可視化手法で最も基本的なものは，語句や語句間の関連の集計表示である．出現回数を棒グラフで表したり，著者や作成日などの文書属性と語句の相関を取った散布図で表示したりするなどである [45]．また，係り受け関係や関係抽出の頻度などを表示することで，用語間の関係の強さを直感的に表す手法も提案されている．

(2) **文書集合の概観表示**
ある属性をもつ文書集合や，自動分類された文書集合が全体としてどのような傾向 (内容) なのかを可視化することは，大量の文書の内容を直感的に把握するために重要である．三末らは，単語をノード，単語間の関連をリンクとしたネットワークに

[‖] http://www.trl.ibm.com/projects/s7710/tm/takmi/takmi.htm

よって文書集合の内容を可視化する手法を提案している [39]．この他の研究として，DualNavi システムでは，単語のネットワークを可視化し，表示された単語のいくつかを選択すると，その単語をもとに再び検索を行い，新たな単語のネットワークを可視化するインタラクティブ検索手法 [63] を提案している．また，大澤らは，高頻度語と，それらの単語と共出現する低頻度語を可視化する手法（KeyGraph）を提案している [47]．KeyGraph を実際にユーザに見せる実験を行い，これらの低頻度語は，ユーザにある種の重要な気づきを与えることができることを示している [46]．このように可視化はあくまで次のインタラクションのためのステップであり，このような可視化画面を見ただけでは，詳細な意味を把握することは困難である．

内容自体の可視化ではなく，文書間の関連性をなんらかの形で可視化することも文書集合全体の傾向把握やブラウジングには有効である．Cutting らは，検索結果の文書をクラスタリングし，関連したものを集めて提示する Scatter/Gather を提案している [11]．更に，文書間の関連性自体を直接可視化するものには，WEBSOM（Self-Organizing Maps for Internet Exploration）** がある．WEBSOM では，自己組織化マップを用いて文書間の距離が近いものを近くの格子に配置する．これにより，似た内容の文書が近くに配置されるので，全体の関係性の把握に役立つ．この他，文書集合の可視化に関しては，文献 [5] の Chapter10 が詳しい．

(3) **自動要約**

自動要約技術は，単純に単一の文書から決められた要約を生成するだけはでなく，“ユーザに適応した動的な要約手法” や “複数文書を対象にした要約手法” が研究の対象となってきている．前者は，特に検索質問に応じた要約を生成する手法（query biased summary）が有名であり，Tombros らによってその有効性が示されている [67]．後者は，次のステップを含む [49]．

(a) 関連するテキストの自動収集

(b) 関連する複数テキストからの情報の抽出

  (b-1) 重要箇所の抽出

  (b-2) テキスト間の共通点の検出

  (b-3) テキスト間の相違点の検出

(c) テキスト間の文体の違いなどを考慮した要約文章の生成

この技術を利用したものとして，新聞記事から関連記事を集めた要約生成や，技術論文のサーベイ生成などがある．このような複数文書を対象としたユーザに適応した自動要約は，情報集約技術の究極の目標であるといえる．しかしながら，現在の技術レベルでは，完全な自動化を達成することは難しく，要約処理自体の精度向上だけでなく，要約プロセスの支援を行い，少しずつ自動化のレベルを上げユーザの負担を軽減するアプローチが有効である．

上記に呼べたように，可視化を行うためには，テキストの構造化を行う必要があり，この際に，テキストが持っていた意味や内容の一部が落とされる．そのため，テキストに含

**http://websom.hut.fi/websom/

まれる情報を単純な表現で表すと，集計は行いやすいが可視化結果を見ても内容の把握が困難となる．逆に，複雑な表現を用いるとテキスト構造化や集計自体が難しくなってしまい，任意の文書情報に適用できなくなる．このように，テキストの構造化や集計のしやすさと表現の詳細度はトレードオフの関係にある．

2.3　情報集約フレームワーク

2.2 節で述べたように，情報集約に関する各要素技術については，様々な分野で詳細に検討がされてきている．しかしながら，第 1 章の 1.2 節の例にあるように，大規模な文書を対象とした情報集約を行うためには，様々なテキスト操作を柔軟に組み合わせる必要がある．また，Hearst[19] は，Broad[10] の研究を例に取り，“重要な情報を抽出するためには，様々なテキスト操作を柔軟に組み合わせて実行することが必要である” と述べている．形式化されたデータに対しては，従来からリレーショナルデータモデルがあり，様々な用途に応じて，柔軟にデータ操作の組合せを行い，所望の結果を抽出することができる．文書データに対して，このような汎用的な枠組みを構築できれば，様々な情報集約タスクを効率的に実行できるようになると考えられる．

実際に，このような目的のため，文書データを対象としたフレームワークの提案が少数ながらも行われてきている．筆者は，テキストを表現するモデルとして 2.2.1 節で述べた主題グラフを採用し，このグラフを操作するグラフ表現代数によって，柔軟にテキスト操作を組み合わせるフレームワークを提案した [69]．提案手法を，“製品を購入する際の比較” を行う情報集約タスクに適用し，ある程度有効な集約結果を得た．しかしながら，各文書を単一の表現で表す主題グラフでは，テキスト構造化の段階で，多くの情報が落とされてしまい最終的な可視化結果を見ただけでは，有用な情報を発見することが難しかった．このため，各文書をもう一度見る必要が生じ，ユーザの作業効率の大幅な向上には至らなかった．

一方，大島らは，商用の検索エンジンから取得した検索結果に対して，文書解析処理を施し，この結果を仮想的なテーブルと見立てて，既存の RDB 上のデータと統合利用する手法を提案している [48]．このようにリレーションに情報を割り当てる手法では，SQL などの問合せ言語によって，効果的に集約を実行できると期待できる．しかしながら，この手法では，ユーザが問合せを行ってから全ての解析処理を行うため即時性の問題がある．また，問合せ言語に SQL を用いているが，文書中から自動抽出された情報という形式化されたデータではないものを集約するという目的に対して，SQL は必ずしも最適なものではない．

2.4　従来研究の課題

本章では，情報集約の要素技術を概観し，これらを統合するためのフレームワークの必要性について述べた．このようなフレームワークを実現するうえで，まず，重要なことは，どのように情報をコンピュータ上にモデル化するのかということである．表現力の高いモデルは，それだけ，テキストの構造化に高度な技術が必要とされ，テキスト操作が複雑なものになるため，大規模な文書データへの適用や高速化が難しい．一方，表現力の低いモ

デルでは，必要なテキスト操作が定義できず，また，集約結果を可視化しても内容を把握することが難しくなる．このように，適切な情報表現のためのモデルを持ち，このモデルに対する操作を柔軟に組み合わせて実行できるフレームワークを構築することは，大規模な文書データに対する効果的な情報集約を実現するうえで重要な課題である．

第3章　情報集約モデルと情報集約データベース

第 2 章では，文書情報の集約を行うためのフレームワークを構築するためには，まず，情報をコンピュータ上にどのように表現するのかを決定することが重要であることを述べた．本章では，対象物と対象物に結びつく属性の集合とで，文書データ中の各情報の断片を表現する情報集約モデルを提案する．情報集約モデルでは，各情報の断片をタプルとした情報要素リレーションによって，文書データ中の全ての集約の対象となる情報を表現する．これにより，情報要素リレーションへの検索と集計によって情報集約タスクを実行できるようにする．また，提案モデルに基づく情報集約データベースのアーキテクチャを示し，その実現課題について述べる．

3.1　情報の表現方法

文書中の情報の表現方法について考察する．まず，個々の文書全体の内容を単位として 1 つの表現を生成したとする．この表現は，文書を単位としたテキスト操作に向いている．例えば，文書の分類や，似た文書の検索などでは，このような文書を単位とした表現が実際に利用されている．しかしながら，通常，文書には様々な内容が記述され，その内容間の関係など細かな情報を表現することは難しい．そのため，このような文書を単位とした表現は，ある文書へのアクセスを目的としている場合には適しているが，より詳細な文書中の情報の断片を集約するのには適さない．

各文書には，いくつもの情報が記述されているが，多くの場合，各情報の断片は，ある特定の対象物に結びつく．例えば，“製品 A の画面はきれい” や “企業 A は，2011 年に新規事業に参入する予定” といった文章に書かれている情報は，製品名や企業名に結びついたものであるといえる．また，多くの情報集約タスクでは，どのような観点で集約を行いたいかをあらかじめある程度想定することができる．例えば，第 1 章の 1.2 節で述べた評判情報の集約タスクでは，集約の観点として，各評判に関する用語や好不評，著者の性別や年代，文書の記述された日時などが挙げられる．

本研究では，以上の考察から情報の断片を，対象物と集計の観点を表す属性（属性名と属性値）の集合とで表現する．ここでは，このような情報の単位を情報要素と呼ぶ．例えば，評判情報を表現する 1 つの例は，

<製品 A，美しい，好評>(<対象物，評価表現，評価極性>)

である．ここで，‘( )’ 内は，情報要素のスキーマに相当し，属性名（又は，対象物）に対応する．この表現によって，評価表現や評価極性を観点とした，評判情報の集約ができるようになる．また，イベントを表現する 1 つの例は，

<企業 A，2011 年，新規事業，参入>(<対象物，日時，イベントの目的語，イベントの動作>)

である．この表現によって，日時やイベントの種類（目的語や動作）による集約ができるようになる．このように，いずれの場合でも，対象物に対して，集約の観点となる属性の集合を付与した表現となっている．

3.1.1　Linked Data との関係

Linked Data では，各情報の単位を主語，述語，目的語のトリプルによって表現している [8]．ここで，主語と述語は URI であり，目的語は，URI か文字列とすることが規定されている．つまり，この表現では，まず，主語となる対象物を明確に特定し，その対象物と関係するものと，その関係性がどのようなものであるかを明確に記述することで，各情報の単位を表現している．

情報要素は，対象物を主語，属性名を述語，属性値を目的語とすれば Linked Data のトリプルと等価な表現である．しかしながら，対象物や属性名が URI であることは，特に規定していない．対象物は，本来，現実世界の実体を特定できることが望ましい．そのため，理想的には，URI のような一意に特定できる参照先や ID を用いるべきである．しかしながら，自然言語で記述された文書中のある用語が表わす対象物の実体を特定することは，非常に困難であるため，対象物を URI とはしていない．一方，属性名は URI とすることが原理的には可能である．ただし，今回の属性名は集約の観点に対応するものであり，各タスクに対して独自のものである．このため，共通的な語彙（オントロジ）を使うメリットが少ないため，現状は，独自のボキャブラリ（スキーマ）によって定義している．

3.1.2　情報要素の規定方法

情報要素は，文脈から切り離されて集約されることになるため，情報要素が単体で表す内容と，文脈中の該当する箇所の表す内容とが一致しないことがあることに注意しなければならない．例えば，“店舗 A に行った．チキンの入ったあのカレーはとてもおいしい” という文章から情報要素を生成することを考える．‘おいしい’ の主語は，‘カレー’ であるため，この構文情報から，仮に，

<カレー，おいしい>（<対象物，評価表現>）

という情報要素を生成したとする．この場合，元の文脈では，‘カレー’ は，‘ 店舗 A のカレー’ もしくは，‘店舗 A のチキンの入ったカレー’ に限定された内容を表している．これに対して，生成された情報要素を単体で見ると，‘カレーはおいしい’ という任意のカレーについての評価を表しているようにみえる．このように，元の文章と生成された情報要素とで表す内容が大きく異なる．

このようなことが起こるのは，対象物を，特定できていないことに原因がある．例えば，対象物として，‘店舗 A のカレー’ というものが取得できたとすると，

<店舗 A のカレー，おいしい>（<対象物，評価表現>）

は，元の文章の内容に近いものである．しかし，この表現には，次の 2 つの問題がある．

(1) ある用語に対してその用語が文脈中でどのような範囲の概念に限定されているのかを正確に特定することは，困難である．

(2) 特定の方法は，文脈により様々な記述方法が考えられ，仮に抽出できたとしても，集計が困難となる．

例えば，上記の例の場合，‘カレー’ を限定する語として，‘チキンの入った’ を付けてしまうと，‘店舗 A のカレー’ と’ 店舗 A のチキンの入ったカレー’ の間で集計を行うことが難しくなる．

上記の考察から，本研究では，現実的な解として，対象物は，固有表現だけを用いることとする．ここで，本研究における固有表現は，組織名（Organization）や人名（Person）といった特定の実体や概念を表す名前（Name）だけを対象とする．本来，固有表現は，名前以外にも数値表現や時間表現を含むが，これらは，対象物にはふさわしくないため，ここでは取り扱わない．対象物が固有表現の場合，文脈情報がなくてもある程度，対象物の指す実体や概念を特定できる．この結果，文脈から切り離した情報要素の内容と，元の文脈中での対応する箇所の内容とが近いものとなる．上記の例では，‘店舗 A’ が，固有表現なので，これを対象物として，情報要素は，

<店舗 A，カレー，おいしい>　（対象物，評価属性，評価表現）

となる．ここで，‘チキンの入った’ の扱いは難しいが，これは，評価属性の表記ゆれと考える．すなわち，対象物の表記がゆれていると情報要素自体の特定性の問題となるが，属性として扱うことで，属性の表記の問題に置き換えられる．この結果，

<店舗 A，カレー，おいしい>　（対象物，評価属性，評価表現）
<店舗 A，チキンの入ったカレー，おいしい>　（対象物，評価属性，評価表現）

の両方について，“店舗 A のカレーはおいしい” は，事実として成立すると考える．

同様に，将来情報の集約タスクにおいて，“中国の経済成長スピードは，衰えを知らない．2012 年は，X%程度の成長となるだろう” という文章からは，対象物として固有表現の ‘中国’ を用いて，次の情報要素を生成する．

<中国，2012 年，X%程度の成長>　（対象物，日時，イベント）

この情報要素は元の文章と近い内容となっている．このように，対象物として固有表現を用い，それに付随する属性を抽出することによって，情報要素が単体で表す内容と元の文脈中での内容が近いものとなる．当然，固有表現を用いたとしても多義語や同姓同名などの問題もあり，内容が必ずしも一致するわけではないが，一般語と比べて特定性が高いため，内容の一致の度合いが高くなると考えている．

一方，文献 [34] では，評判情報に関して，詳細に分析を行い，次に示す形式で各評判情報を表現する方法を提案している．

<対象物，評価属性の連続，評価表現，シチュエーション，評価者>

この考え方を用いると，“店舗 A に行った．寒い日に食べると，チキンの入ったあのカレーはとてもおいしい” は，次のように表現される．

対象物: 店舗 A

評価属性の連続: [チキンの入った，カレー]

評価表現: とてもおいしい

シチュエーション: 寒い日に食べると

評価者: この文章の著者

ここで，シチュエーションや評価者も情報要素の 1 つの属性とみなせる．ただし，情報要素の各属性は構造を持たないため，複数の評価属性の連続やシチュエーションの細かい指定などには対応できない．属性に構造を持った表現を採用することによって，より正確に情報の断片の内容を表現することができると考えられる．しかしながら，第 2 章の 2.4 節で述べたように情報の表現が複雑になると，自動生成が困難になり，また，集約のための処理も複雑なものとなる．このように，本研究の情報要素は，次の 2 つの特徴を持っている．

(1) 対象物を固有表現と規定することで，特定性をある程度確保できる．

(2) 各属性には構造がないため，取扱いが容易である．

このため，記述能力は限定的であるが，大規模な文書データの自動処理に適用できるものであると考えている．

3.2　情報集約モデル

3.1 節で述べた情報要素を情報の単位とした情報集約モデルを提案する．まず，用語を次のように定義する．

情報要素属性： 情報要素を構成する属性．属性名と属性値をもつ．複数の情報要素属性によって，各情報要素を表現する．例えば，評判情報を表す情報要素を
<製品 A，操作，直感的，好評>（<対象物，評価属性，評価表現，評価極性>）
のように 4 つの情報要素属性で表現する．

文書属性： 文書に対して 1 対 1 に割り振られる属性．例）URL，作成日，文書種別（レビューなど) など．

文書集合属性： 文書集合に対して割り振られる属性．例えば，ある著者の書いたブログ記事全てを文書集合にとった場合の著者の性別や年代など．

文書は 1 つの文書集合にしか属することができないと制限すると，文書属性や文書集合属性は，各情報要素に対して一意に特定できる．そのため，これらの属性を各情報要素タプルに join することで，図 3.1 のような各情報要素をタプルとする単一の情報要素リレーションによって，大規模文書データに存在する全ての情報の断片を表現できる．このよう

ID	対象物 (IS)	評価属性 (IP)	評価表現 (IE)	評価極性 (IO)	URL (DI)	日付 (DD)	文書種別 (DT)	性別 (BS)	年代 (BG)	Blog siteID (BID)
1	製品A	画面	きれい	好評	http://host/tomita/111.html	20100316	ブログ	男性	20	host1/tomita
2	製品A	操作	簡単	好評	http://host/tomita/111.html	20100316	ブログ	男性	20	host1/tomita
3	製品B	電池	短い	不評	http://host/tomita/111.html	20100316	ブログ	男性	20	host1/tomita
4	製品C	画面	みにくい	不評	http://host/tomita/112.html	20100317	ブログ	女性	20	host1/tomita
5	製品A	–	美しい	好評	http://host2/suzuki/011.html	20100327	レビュー	女性	30	host2/suzuki
…										

図 3.1 情報要素リレーション

に情報を表現すると，たとえば，第 1 章の 1.2 節で挙げた例の (1)〜(6) の作業は全て，次のように情報要素リレーションに対する検索，グループ化と個数の集計とみなすことができる．

(1)，(2) 対象物及び文書種別によるタプルの検索

(3)，(4) 評価表現及び評価極性といった情報要素属性によるタプルのグループ化と個数の集計

(5) 性別，年代といった文書集合属性によるタプルのグループ化と個数の集計

(6) 日付といった文書属性によるタプルのグループ化と個数の集計

また，(7) のクロス集計は，いったんグループ化した各グループを再度グループ化し，個数を集計するといった多段のグループ化処理とみなせる．例えば，好不評の性別ごとの違いは，性別でグループ化した後に，各グループを好不評でグループ化し，集計すればよい．

以上のように，本モデル上では，情報集約タスクは，

(i) 情報要素をタプルとするリレーション（情報要素リレーション）の生成

(ii) 情報要素リレーションからのタプルの検索

(iii) 多段の情報要素タプルのグループ化と集計

に置き換えられる．このように，単一の情報要素リレーションのビューをもつことによって，各情報要素がもつ条件や集計結果の生成方法などを容易に変更し，集約処理を実行できるようになる．

3.3　情報集約データベース

3.2 節で述べた情報集約モデルを実現するためのデータベースのアーキテクチャについて述べる．図 3.2 は，このようなデータベースがもつ機能を模式的に表したものである．本研究では，このデータベースを情報集約データベース（IADB: Information Aggregation DataBase）と呼ぶ．IADB で情報集約タスクを実行する処理フローは次のとおりである．

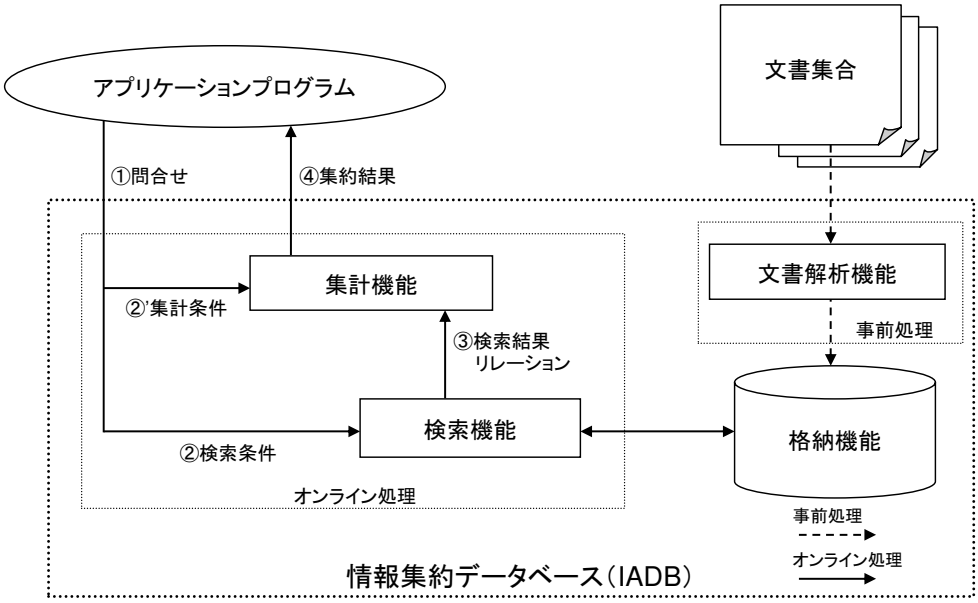

図 3.2 情報集約データベース（IADB）の模式図

(1) 事前処理として，文書解析機能は，集約の対象となる文書集合をあらかじめ解析し，何らかの形で格納しておく．

(2) アプリケーションプログラムは，ユーザの情報集約要求に応じて，問合せを IADB に発行する．ここで，問合せは，情報要素リレーションに対する検索条件と集計条件からなる．

(3) IADB は，問合せを受信すると，問合せの中の検索条件を取り出し，検索機能に渡す．

(4) 検索機能は，格納されたデータを用いて，仮想的な情報要素リレーションを生成し，検索条件を適用することで検索結果リレーションを生成する．

(5) 集計機能は，問合せ中の集計条件を検索結果リレーションに適用することで集計結果を生成し，これを集約結果として返却する．

(6) アプリケーションプログラムは，集約結果を可視化してユーザに提示する．

ここで，実際に文書がどのように格納されているのかにかかわらず，アプリケーションプログラムからは，全ての情報の断片が，仮想的な情報要素リレーションとして見える必要がある．これにより，各タスクに応じた情報集約要求は，情報集約リレーションへの検索と集計といった問合せの形で記述できるようになる．

3.4　情報集約データベースの実現上の課題

3.3 節で述べた IADB を実現するためには，次の 2 つの技術課題がある．

表 3.1 情報集約データベースの実現へのアプローチ

技術課題	技術課題へのアプローチ	詳細
技術課題 1	文書から対象物を抽出するための固有表現辞書の自動構築手法	第 4 章
	対象物を表す入力キーワードを用いた動的な情報要素リレーションの生成手法	第 5 章
技術課題 2	クロス集計と表記ゆれに対応した情報集約言語	第 6 章

技術課題 1: 文書中の情報の断片からの情報要素タプルの自動生成

技術課題 2: 集約処理を実行するための問合せ言語

これらの技術課題とその解決へのアプローチを表 3.1 に示す．次に各アプローチの概要を述べる．

技術課題 1 に関して，情報要素タプルの生成に利用できる技術として，固有表現抽出や評判，時間情報の抽出などがある．ここで，評価表現 [34] や時間表現などは一般語であるため事前に抽出できることが多いが，対象物を表すキーワード（例えば製品名）は未知語となることが多く，文書から抽出することは依然として困難である [13]．そのため，対象物の抽出を容易にするために，固有表現の辞書を自動生成することが有効である．第 4 章では，このような固有表現の辞書の自動構築手法の課題と解決策を述べる．

また，固有表現の辞書を自動生成する手法を確立し，辞書の語彙がいくら増えたとしても，固有表現は，日々生まれるため未知語はなくならず，対象物を完全に自動抽出することは不可能である．このため，対象物を含む完全な形で情報要素タプルを事前に生成することはできず，事前生成したタプルを通常の RDB で管理する方法は利用できない．一方，情報集約タスクは多くの場合，分析の対象となるキーワードでの検索によって開始されるため，入力キーワードとの単純なマッチングによって，情報集約タスク実行時に文書中から分析の対象となる用語を抽出することは容易である．そのため，入力されたキーワードを利用して情報報要素タプルを生成するアプローチが有効である．しかしながら，当然，情報要素タプルを生成する処理全てを実行時に行っていたのでは，実行時間が長くかかりすぎる．そこで，第 5 章では，特に，事前抽出が可能な情報要素の一部だけをあらかじめ処理し，キーワードと結びつける処理だけをオンラインで実行することで，未知語に対応しながら，高速に情報要素リレーションを生成する手法を提案する．また，提案手法を評判情報の集約タスクにおける情報要素リレーションの生成処理に適用し，その高速化の効果を検証する．

技術課題 2 に関して，情報要素リレーションが生成されれば，SQL を用いてある程度情報集約処理を記述することができる．しかしながら，アプリケーションプログラムでは，集約結果をユーザに対して分かりやすく表示する必要があるが，形式化されたデータの検索を主な目的とする SQL では，必ずしもこのような集約結果の可視化に向いた集計データを生成できるわけではない．特に，次の 2 点が SQL では対応できない重要な要件であると考えている．

(a) 複数の観点でのクロス集計結果の可視化が行いやすいように，階層的な内訳をもつ

集約結果を生成できること

(b) 自然言語で記述された文書から抽出された属性値を対象とできるように，表記ゆれなどに対応した柔軟な集計を実行できること

第 6 章では，これらの要件の詳細について述べるとともに，情報集約言語を設計・実現する．

第4章　固有表現辞書の自動構築

文書中の対象物を表す固有表現を抽出し，その固有表現がどのようなクラスに属するのかを判定できると，より高度な情報集約を実現することができる．しかしながら，文書中の固有表現の抽出は事前知識がないと難しいため，固有表現辞書を構築するアプローチが有効である．本章では，このような辞書の自動構築を目指し，ある用語を与えたときに，その表記が対象クラスに属するかどうかを自動判定する手法を提案する．特に，ある用語が複数の語義をもつ場合に，全ての語義を網羅的に抽出することを目的とする．まず，表記ごとに全ての文脈情報を合成してからクラスの判別モデルの学習・推定を行う従来手法について，多義語が精度に与える影響を検証する．この結果から，推定対象が多義語の場合に，対象とする語義以外の文脈から得られた特徴量が悪影響を及ぼし，精度の低下が大きいことを示す．次に，従来手法とは異なり，推定対象の表記が出現する個々の文脈ごとに推定を行い，推定結果であるスコアを合成する手法を提案する．提案手法では，対象とするクラス以外の語義で，その表記が用いられている文脈の影響を軽減できるため，使用頻度の少ない語義に対するクラス判定の精度を向上できることを示す．

4.1　対象物の抽出と辞書構築

評判情報の抽出を行うシステムを考えた場合，文書中にある ‘love letter’ という文字列が映画名であると分かると，‘love letter–BGM が最高’ は結びつくが，‘love letter–おいしい’ は結びつかないことが容易に判断できる．このように，文書中に記述された対象物を表す固有表現を抽出し，その詳細なクラスを判定することは，高精度の情報集約を行ううえで不可欠である．ここで，本研究では，固有表現の名前（Name）だけを取り扱い，数値表現や時間表現は対象物とは成りえないため除外して考える．

固有表現の抽出に関しては，従来，MUC[16] や IREX[59] などの会議を中心に研究が行われ，ルールや学習に基づく手法の有効性が示されている．しかしながら，その多くは，広い概念のクラスを対象としたものであり，細かい粒度の固有表現に対応するためには，学習データの準備，ルールの整備などの問題がある．一方，関根らは，情報抽出などの用途に利用することを想定し，固有表現を網羅的に分類した拡張固有表現階層を提案している [58]．表 4.1 に，拡張固有表現階層* の一部を示す．拡張固有表現階層は，名前を中心に，固有表現を可能な限り網羅的に分類することを想定して作成されている．この階層に基づく固有表現辞書があれば，固有表現の抽出の精度を飛躍的に高めることができる．しかしながら，固有表現は日々生まれるため，このような分類体系があったとしても，辞書

*http://sites.google.com/site/extendednamedentityhierarchy/top%E4%BB%A5%E4%B8%8B%E3%81%AE%E9%9A%8E%E5%B1%A4%E3%81%AE%E5%85%A8%E3%83%AA%E3%82%B9%E3%83%88

表 4.1 関根の拡張固有表現階層（抜粋）

拡張固有表現階層（ENE）のクラス名			例	クラス名英語表記
人名			岡本文弥，カーン，長門美保	Person
組織名（Organization）	組織名-その他		総務課，孔門の十哲，向田ファミリー	Organizaton–Other
	競技組織名（Sports-Organization）	競技組織名–その他	野良黒山の会，桐山部屋，馬家軍	Sports–Organization–Other
		プロ球技組織名	読売ジャイアンツ，ＡＣミラン，鹿島アントラーズ	Pro–Sports–Organization
		競技リーグ名	ＮＢＡ，セリエＡ，セントラル・リーグ	Sports–League
		…	…	…
製品名（Product）	芸術作品名（Art）	映画名	七人の侍，モダン･タイムス，ゴジラ，男はつらいよ	Movie
		音楽名	動物の謝肉祭，おけさ節，魔弾の射手	Music
…	…	…	…	…

に用語を手動で登録し続けることは容易ではない．そのため，固有表現辞書の語彙を自動的に獲得する手法が必要とされている．

固有表現辞書の語彙を獲得する処理は，

(1) 用語の収集

(2) 各用語のクラス判定

という 2 つのステップに分けて考えられる．用語の収集では，辞書に登録する用語を網羅的に集めることが必要とされ，各用語のクラス判定では，各用語が所属するクラスを網羅的に判定することが必要とされる．本研究では，第 2 ステップの各用語のクラス判定を課題とする．第 1 ステップの用語の収集は，検索クエリのログや対象とする文書集合に含まれる全ての未知語を用いることなどによって，かなりの割合を網羅的に収集できると考えている．また，初期の辞書は人手である程度作成できることを想定する．したがって，本研究の課題は，“ターゲットとするある対象クラス集合と各クラスに属する用語の表記のセットが，教師データとして与えられたときに，未知語の表記を与えると，その表記が各クラスに所属するかどうかを判定すること” となる．例えば，自動車の名前の集合が教師データとして与えられたときに，入力された未知語が自動車かどうかを判定するものである．

本研究では，特に，入力された表記が多義をもつ場合に，その表記が所属する全てのクラスを判定することを課題とする．これは，例えば，ブログ記事から ‘音楽名’ を抽出するために辞書を利用することを想定し，ある多義がある語（例：love letter) が ‘音楽名’ に成

りうる場合，‘音楽名’としての用例が実際にどれぐらいあるのかにかかわらず，辞書には登録しておく必要があるからである†.

4.2　クラス判定タスクの定義と多義性の問題

4.2.1　クラス判定タスクの定義

まず，用語を定義する．本研究では，拡張固有表現のクラスだけを扱うこととし，1 つの表記が複数の拡張固有表現のクラスに属することを，“ある表記は多義をもつ” という．ここで，クラスとは，拡張固有表現階層の最下層を表すこととする．また，ある表記がある特定のクラスに属する用語としての意味のことを ‘語義’ と呼ぶ．例えば，‘love letter’ は ‘音楽名’ というクラスに属し，‘love letter’ という表記が ‘音楽名’ として使用された場合の意味を ‘love letter’ の ‘音楽名’ としての語義という言い方をする．また，本研究では，文脈は各表記の出現する文とし，そこから得られる特徴量を文脈情報と呼ぶ．

本研究で想定しているクラス判定タスクは次のとおりである．

学習処理： 教師データとして，<表記，クラス>ペアの集合と，タグなしコーパスを与える．タグなしコーパスから文脈情報を取得し，教師データに含まれる全てのクラスについて判別モデルを生成する．

推定処理： 入力として，未知語の表記，各クラスの判別モデル，タグなしコーパスを与える．学習処理と同様にタグなしコーパスから文脈情報を取得し，各判別モデルを用いて，入力された表記が，各クラスに属するかどうかを判定する．

学習時に，ある表記が多義をもつ場合は，語義の個数分だけ<表記，クラス> ペアを与えることとする．推定時に，ある表記が多義をもつ場合は，その表記が，推定時に入力されたタグなしコーパス中で所属していた全てのクラスを正解とする．このように，本クラス判定タスクは，多義語の曖昧性解消を目的としているのではなく，ある表記がもつ語義を網羅的に収集することを目的としている．

4.2.2　クラス判定に関する関連研究

辞書の自動構築に関する研究の中には，クラス階層（又は上位概念）が人手で与えられる想定と，クラス階層自体の構築を目的としているものがある．クラス階層の自動構築に関して，文献 [52] の手法では，タグなしコーパス中の語をクラスタリングし，各クラスタに対して，is-a パターンによって各クラスタの名前（上位概念）を付与している．しかしながら，この手法では，抽出される上位概念の一貫性を保つことが難しく，コーパスを変えると異なるクラス階層が出力されることが予想されるため，情報抽出などの用途には利用しにくい．

クラス階層を与えて，文字列パターン [18] や並列構造 [31] を利用してコーパスから網羅的に，同じクラスに属する用語を収集する手法があるが，特定のパターンにマッチす

†登録された表記が実際の文脈においてどのクラスの語義かを判定するタスクは，曖昧性解消の問題であり，本研究のスコープ外としている．

る用語しか抽出できないため用語の網羅性の観点で問題がある．網羅性を上げるために，少量のシードとクラスを与え，文字列パターンをブートストラップによって学習するもの [51, 13] があるが，この場合でも，学習されたパターン中に出現した用語だけしか抽出できない．また，用語抽出では，用語の切れ目を見つける必要があるため，抽出のルールには，通常，文字列パターンといった限られた形式の表現を採用する必要がある．これに対して，想定しているクラス判定タスクでは，推定対象の用語を与えているので，その切れ目を見つける必要がないため，文（や場合よっては文書など）に含まれる全形態素といった広い範囲の特徴量を利用した学習手法を採用できる．このため，検索ログの上位ワードなど，辞書に載せるべき用語がすでにある場合に，原理的には，コーパス中に出現したこれら全ての用語についてのクラス判定ができる．

文献 [35, 75, 22] では，検索エンジンに未知語の表記を投げ，その表記が出現する文脈情報を全て合成することで特徴ベクトルを生成し，これを関連語の抽出やクエリのタイプ推定に利用している．この考え方を用いると入力された任意の表記のクラスを判定できると考えられ，用語の網羅性の点で優れている．しかしながら，これらのいずれの手法でも，1 つの表記に対して，全ての文脈情報を合成した 1 つの特徴ベクトルを生成している．そのため，推定対象が多義語の場合に，頻度の低い語義の特徴が失われ，全ての語義を網羅的に判定することは難しいという問題がある．また，実際に多義語の影響がどの程度であるのかの検証は行われていない．4.2.3 節では，この手法をベースラインと考え，本研究の想定する 4.2.1 節のクラス判定タスクに適用する場合の多義語の問題点を詳しく述べる．

4.2.3　表記特徴量法と多義性の問題

本研究では，文献 [35, 75, 22] の考え方をクラス判定に適用した次の手法をベースラインと考え，表記特徴量法と呼ぶ．

- 表記特徴量法（学習処理）

  (1) 教師データとして<表記 f，クラス c>ペアの集合と，タグなしコーパスを入力する．

  (2) 各<表記 f, クラス c>ペアについて次の処理を行う．

    (a) コーパス中で表記 f の出現する全ての文脈情報を取得し，これらを合成して表記 f に対する特徴ベクトル V_f を生成する．このベクトルを表記特徴ベクトルと呼ぶ．

    (b) <表記 f, クラス c, 表記特徴ベクトル V_f>を合わせて保存する．

  (3) 判別モデル生成対象の各クラス c について次の処理を行う．

    (a) <表記, クラス, 表記特徴ベクトル>集合の中からクラス c に属する全ての表記特徴ベクトルを取得し，これらを正事例集合とする．

    (b) 同様に，クラス c に属さない全ての表記特徴ベクトルを取得し，これらを負事例集合とする．

    (c) 正事例集合と負事例集合を SVM（Support Vector Machine）学習器に与えてクラス c の判別モデルを生成する（one-vs-rest 法）．

- 表記特徴量法（推定処理）

  (1) 入力として表記 f，全クラスの判別モデル，タグなしコーパスを与える．

  (2) コーパス中で表記 f の出現する全ての文脈情報を取得し，これを合成して表記特徴ベクトル V_f を生成する．

  (3) 表記特徴ベクトル V_f を各クラス c の判別モデルに入力し，表記 f のクラス c への所属スコア S_{fc} を得る．

  (4) 所属スコア S_{fc} が閾値を超える全てのクラスに属するものとして，表記 f を辞書に追加する．

想定しているクラス判定タスクの教師データは，表記とクラスだけであり，タグ付きコーパスではないため，表記が多義をもつ場合に，どの文脈がどの語義に対応するのかが分からない．表記特徴量法（学習処理）では，多義語を特に意識することなく全ての文脈情報を合成して扱っている．そのため，あるクラスの判別モデルを学習する際に，他のクラスの語義で用いられている文脈の特徴量が混入してしまうという問題がある．例えば，“文脈 1：love letter が上映される” と，“文脈 2：love letter の歌詞が良い” という 2 つの文脈があったとする．‘love letter’ は，文脈 1 では ‘映画名’，文脈 2 では ‘音楽名’ の語義で用いられている．この場合，表記特徴量法では，‘音楽名’ の判別モデルを生成する際に，‘映画名’ の文脈から得られた特徴量（‘上映’ など）が混入してしまう．

一方，表記特徴量法（推定処理）では，入力された表記が複数の語義をもっていたとしても，これらの文脈情報は全て合成されてしまう．このため，例えば，推定対象の ‘さくら’ という表記が，‘植物名’ と ‘音楽名’ の 2 つの語義をもつ場合，‘さくら’ が ‘植物名’ に属するかどうかを判定する際に，‘音楽名’ の文脈が影響を及ぼす．特に，学習器が one-vs-rest の場合，‘音楽名’ に関する特徴量（‘歌詞’ など）は負事例として学習されているので，‘さくら’ の ‘植物名’ に属するかどうかを表す所属スコアが必要以上に低くなってしまうという問題がある．

このように，表記特徴量法では，

問題 (1) 学習時に多義語を完全に除外することは不可能であり，ある程度の割合で，対象とする語義以外の文脈情報が混入した判別モデルが生成される．

問題 (2) 推定対象が多義語である場合，対象となるクラスとは関係のない語義の文脈情報が利用され，これが負事例の特徴量とマッチするため，必要以上にそのクラスへの所属スコアが低くなる．

という 2 つの問題がある．

4.3　多義性の問題の影響度調査

本節では，問題 (1)，問題 (2) が実際にどの程度，精度に影響しているのかを検証し，特に問題 (2) が精度に大きな影響を与えていることを示す．これらの検証を行うために，まず，理想的な状態として，タグ付きコーパスを用いて，ある用語の出現する各文脈がどの語義と対応するのかを特定したうえで，語義ごとに文脈情報を合成して特徴ベクトルを生

成し，学習及び推定を行う手法を示す．この手法を語義特徴量法と呼ぶ．ここで，語義特徴量法は，学習時にタグ付きコーパスを必要とし，また，推定時には推定する対象である語義があらかじめ分かっていることを想定しているので，クラス判定タスクの定義上，現実にはありえない理想的な状態であり，本検証のために導入したものである．この理想的な状態の語義特徴量法と比べて，ベースラインの表記特徴量法が，どの程度の精度の低下があるのかを測定することで，問題 (1) と問題 (2) がどの程度の影響を与えているのかを検証する．

4.3.1　語義特徴量法

タグ付きコーパスを用いた場合，表記の出現ごとに語義を特定できるため，表記ごとに特徴ベクトルを生成するのではなく，語義ごとに異なる特徴ベクトルを生成できる．この場合，表記特徴量法の学習処理のステップ (2)-(a) にあたる処理は，次のように変更となる．

- 語義特徴量法（学習処理）

(2)-(a)’ コーパス中で表記 f がクラス c の語義で出現する全ての文脈情報を取得し，これらを合成して表記 f のクラス c の語義に対する特徴ベクトル V_{fc} を生成する．このベクトルを語義特徴ベクトルと呼ぶ．

その他の処理は，表記特徴ベクトルを語義特徴ベクトルに置き換えるだけである．4.2.3 節の例では，‘love letter’ の ‘音楽名’ としての語義に対応した語義特徴ベクトルは，(‘歌詞’, ‘良い’）となり，‘映画名’ の文脈から得られる ‘上映’ は含まれない．

一方，推定時に，未知語に対して語義が事前に分かっている状態は，当然ありえない．しかしながら，対象となるクラスの語義の文脈情報だけを用いて推定が行える理想的な状態を再現するために，次の手順によってある表記 f の対象クラス c への所属スコア S_{fc} を求める．

- 語義特徴量法（推定処理）

(1) 入力として表記 f，全クラスの判別モデル，タグ付きコーパスを与える．

(2) 表記 f の語義 m ごとに，次の処理を行う．

(a) コーパス中で表記 f が m の語義で出現する全ての文脈情報を取得し，これを合成して表記 f の語義 m に対する語義特徴ベクトル V_{fm} を生成する．

(b) 語義特徴ベクトル V_{fm} を各クラス c の判別モデルに入力し，表記 f の語義 m についてのクラス c への所属スコア s_{fcm} を得る．

(3) 表記 f ごとに，得られた所属スコア s_{fcm} の最大値を求め，表記 f のクラス c への所属スコア S_{fc} とする．

$$S_{fc} = \max_{m}(s_{fcm}) \tag{4.1}$$

(4) 所属スコア S_{fc} が閾値を超える全てのクラスに属するものとして，表記 f を辞書に追加する．

本研究では，表記特徴量法，語義特徴量法のいずれの場合でも，文献 [60] を参考に，特徴量は次のものを用いた．

(1) 対象表記の出現する文に含まれる全形態素（名詞/動詞/形容詞）の表記

(2) (1) を，対象表記の左側と右側で分けて集計したもの

(3) 対象表記の（左側/右側）に連接する 1 形態素の (表記/品詞/字種)

(4) 対象表記の（左側/右側）に連接する 2 形態素の (表記/品詞/字種)

(5) 対象表記文字列に含まれる全形態素の (表記/品詞/字種)

ここで，字種は文献 [60] と合わせて 9 種類 (句読点，ひらがな，数字など）とした．また，学習器には，LIBLINEAR[‡]を用いた．

いずれの手法でも特徴量を計算する際には，頻度を考慮せずに，重みは全て ‘1’ とした．これは，予備実験の結果から，頻度を重みに用いると精度が低下する傾向にあったためである．この理由は，表記ごとの出現数に大きな偏りがあるため，出現数の大きな表記の影響が過度に大きかったこと，及び，学習器には SVM の線形カーネルを用い，各特徴量の影響度は SVM によって自動調整されるため，重みを付与する効果が小さかったこと，にあると考えている．

4.3.2　評価方法

問題 (1) の学習時の多義性の影響を検証するために，多義語の文脈情報を全て合成して学習した場合と，語義ごとに分けて学習した場合との比較を行う．具体的には，次の試験パターンで検証を行う．

- 学習処理
  - (a) 語義特徴量法：教師データとして，学習セットに含まれる全ての <表記，クラス>ペアを用いる．
  - (b) 表記特徴量法 (全語義)：教師データとして，学習セットに含まれる全ての <表記，クラス>ペアを用いる．
  - (c) 表記特徴量法 (最大頻度)：教師データとして，各表記についてコーパス中の出現頻度が最大の<表記，クラス>ペアだけを用いる．
  - (d) 表記特徴量法 (多義語なし)： 教師データとして，多義語を除く全ての <表記，クラス>ペアを用いる．
- 推定処理
  全ての手法で，表記特徴量法 (推定処理）を用いる．

[‡]http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

(b) は，教師データの各表記に対して，その表記がコーパス中で所属する全てのクラスが指定できる想定であり，表記特徴量法における理想的な教師データの与え方である．しかしながら，実際に，ある表記をあるクラスの教師データとして採用する際に，その表記が所属しうる他の全てのクラスを網羅的に指定することは現実的には難しい．これに対して，(c) は，クラスごとに，頻度の高い表記を集め，集められた各表記が他のクラスに所属しうるかの詳細な判断は行わないことを想定したコストの少ない教師データの与え方である．また，(d) は，(b) と同様に実際には指定することが難しい，多義語が与える影響を検証するための特別な教師データの与え方である．

ここで，(a) と (b) については，ある表記が多義をもつ場合，その表記に対しての事例（<表記，クラス>ペア）は複数となるため，負事例の与え方として，次の 2 つを用いる．

(NI) 多義語は正解クラス c 以外にも属すると考え，クラス c の負事例にも採用する．

(NN) 多義語は正解クラス c だけに属すると考え，クラス c の負事例には採用しない．

(NI) は，多義語が正負の両方に含まれるため特徴量が相殺され，問題 (1) の影響が小さくなると考えられる．一方，(NN) は，クラスごとに純粋に，表記が所属するかどうかで，正負の事例の選択を行っているため，表記が各クラスに属するかどうかを判定するという目的に即したものとなる．また，問題 (1) の影響を，推定時に多義語が与える影響と分けて検証を行うため，検証データとして全表記集合と，多義語を除く表記集合の 2 通りで測定を行う．ここで，全ての手法で，推定側は表記特徴量法を用いるが，多義語を除く表記集合では，語義特徴量法のものと全く同じ特徴ベクトルとなる．

以上のように，

(a-NI) 語義特徴量法（全語義，負事例にも含める）

(a-NN) 語義特徴量法（全語義，負事例に含めない）

(b-NI) 表記特徴量法（全語義，負事例にも含める）

(b-NN) 表記特徴量法（全語義，負事例に含めない）

(c) 表記特徴量法（最大頻度)

(d) 表記特徴量法（多義語なし）

の 6 手法で学習を行い，2 つの検証データ（全表記集合，多義語を除く）で精度の測定を行う．

一方，問題 (2) の推定時の多義性の影響を検証するために，多義語の文脈情報を全て集約して推定を行った場合と，語義ごとに分けて推定を行った場合との比較を行う．特に，評価セットとして全表記集合を用いた場合と，多義語だけを用いた場合で比較を行うことによって，多義性の影響を明らかにする．具体的には，次の試験パターンで検証を行う．

- 学習処理
  全ての手法で，学習時の多義性の検証で用いた試験パターンの (b-NN) 表記特徴量法（全語義，負事例に含めない）を用いる．

- 推定処理

  (l) 語義特徴量法（推定処理） 全表記集合

  (m) 表記特徴量法（推定処理） 全表記集合

  (n) 語義特徴量法（推定処理） 多義語だけ

  (o) 表記特徴量法（推定処理） 多義語だけ

評価セットのクラス体系として，拡張固有表現階層-7.1.0-を用いた[§]．タグ付きコーパスとして，拡張固有表現タグ付きコーパス（毎日新聞）[17] を用いた[¶]．このコーパスには，8,584 記事に対して，拡張固有表現階層に従いタグ付けが行われている．

評価セットの作成は次の手順で行った．

(1) 毎日新聞コーパスでタグ付けされた全ての拡張固有表現の中で，‘名前（Name）’配下のほぼ全てのクラス（141 クラス）に所属する全ての用語を抽出し<表記，クラス>ペアを作成した．

(2) これらのペアを 3 等分（A，B，C）した．ここで，3 等分する際には次の制約を用いた．

制約 a) 同じ表記は，A，B，C のいずれかに属するようにし，同じ表記が，評価セット間をまたがらないようにする[‖]．

制約 b) 各クラスについて，A，B，C に割り当てられた<表記，クラス>ペア数が少なくなりすぎないようにする．

(3) A，B，C のいずれかを検証データ，残り 2 つを教師データとした 3 つの評価セット（E1，E2，E3）を作成した．

教師データには，全 141 クラスに所属する全ての用語を用いた．ただし，検証データには，低頻度のクラスを用いると誤差が大きくなりすぎるため，最低でも各評価セットについて，5 語の多義語を含むクラスだけを用いた．この条件を満たすものは，31 クラスであった．対象クラス及び用語数と多義語の例を表 4.8 に示す．

4.3.3　評価結果

精度の測定では，まず，各手法でクラスごとに推定を行い，各表記の所属スコアを求める．次に，クラスごとに所属スコアの降順でソートを行い，スコアの上位から各表記のいずれかの語義が対象とするクラスに属する場合に正解，いずれの語義も属さない場合に不正解として，再現率と適合率を求める．最後に，再現率が 0.1 刻みになるように補完を行い，全クラスについて同一の再現率となる適合率の平均を求め，更に，0.0〜1.0 の 11 点の平均を求める（11 点平均補完適合率 [32]）．

[§]http://sites.google.com/site/extendednamedentityhierarchy/top%E4%BB%A5%E4%B8%8B%E3%81%AE%E9%9A%8E%E5%B1%A4%E3%81%AE%E5%85%A8%E3%83%AA%E3%82%B9%E3%83%88

[¶]CD–毎日新聞データ集 1995 年版を使用．

[‖]この制約によって教師データと検証データに同じ表記が混入しないようにしている．

表 4.2 学習手法を変更したときの 11 点平均補完適合率の比較（全表記集合）

	語義特徴量法		表記特徴量法			
	(a-NI)	(a-NN)	(b-NI) 全語義	(b-NN) 全語義	(c) 最大頻度	(d) 多義語なし
E1	53.83	55.88	53.56	55.46	52.37	49.92
E2	54.95	56.76	54.53	56.61	53.44	51.87
E3	56.34	58.44	55.72	57.68	54.88	52.42
平均	55.04	57.03	54.60	**56.58**	53.56	51.40

表 4.3 学習手法を変更したときの 11 点平均補完適合率の比較（多義語を除く）

	語義特徴量法		表記特徴量法			
	(a-NI)	(a-NN)	(b-NI) 全語義	(b-NN) 全語義	(c) 最大頻度	(d) 多義語なし
E1	54.91	54.27	54.40	53.62	53.96	54.09
E2	55.18	54.54	54.70	53.73	54.24	55.05
E3	57.37	56.85	56.55	55.35	56.30	56.46
平均	55.82	55.22	55.22	54.23	54.83	55.20

問題 (1) の検証結果として，表 4.2，表 4.3 に，4.3.2 節で述べた 6 つの学習手法について，検証データとして全表記集合を用いた場合と，多義語を除く表記集合を用いた場合の 11 点平均補完適合率を，それぞれ示す．まず，語義特徴量法と表記特徴量法とを比べた場合**，いずれの場合でも，語義特徴量法が表記特徴量法の精度を若干上回ったが，その差は，0.44〜0.99 パーセントポイントと小さかった．この理由は，多義語は全体に対して平均で約 18%とそれほど多くなく，一部の多義語によって，他のクラスの特徴量が含まれたとしても，そのクラスに属する別の表記が多数あるため，これらのふさわしくない特徴量は学習時に支配的にはならかったためと考えている．例えば，前述した例では，‘love letter’ の ‘映画名’ に関する特徴量である ‘上映’ は，‘音楽名’ では支配的にはならなかった．

次に，表記特徴量法における教師データの与え方による精度を比較する．表 4.2 の多義語を含む全表記集合を用いた場合の精度は，

$$\text{(b-NN)} > \text{(b-NI)} > \text{(c)} > \text{(d)}$$

となったのに対して，表 4.3 の多義語を除く検証データを用いた場合は，

$$\text{(b-NI)} \simeq \text{(d)} > \text{(c)} > \text{(b-NN)}$$

と大きく異なる結果となった．ここで，(b-NI) では，多義語は正負の両事例に含まれるため，多義語のもつ特徴量の影響は相殺される．このように，推定対象が多義語ではない場合，問題 (1) の多義語の影響がより起こりにくい (b-NI) が (d) と同様に精度が高かった．一方，検証データに多義語を含む場合，逆に，多義語の特徴量の影響がより残りやすい手法の順で，精度が高かった．この傾向は，語義特徴量法での比較 (全表記集合：(a-NI)<(a-NN),

**表 4.2，表 4.3 における，それぞれ (a-NI) と (b-NI)，(a-NN) と (b-NN) の比較．

表 4.4 推定手法を変更したときの 11 点平均補完適合率の比較

	全表記集合		多義語だけ	
	(l) 語義特徴量法	(m) 表記特徴量法	(n) 語義特徴量法	(o) 表記特徴量法
E1	57.03	55.46	66.90	57.29
E2	57.62	56.61	65.72	57.47
E3	59.29	57.68	67.06	55.37
平均	57.98	56.58	66.56	56.71

多義語を除く表記集合 (a-NI)>(a-NN)）でも同様の傾向が得られた．この理由は，今回の評価セットでは，教師データと検証データで，多義語がもつクラスの分布が似ているものが多かったためである．例えば，推定対象が ‘Flora’ と ‘Food_Other’ をもつ多義語（‘ブルーベリー’ など）の場合，教師データにも全く同じ，‘Flora’ と ‘Food_Other’ の両方をもつ多義語（‘リンゴ’ など）が多数あった．‘Flora’ が推定対象クラスの場合，(b-NN) では ‘Food_Other’ をもつ不正解事例（‘アイスクリーム’ など）を誤判定する可能性があるが，‘ブルーベリー’ を正解とできる．一方，(b-NI) では，‘アイスクリーム’ の誤判定を防げるが，‘ブルーベリー’ を不正解と判定する可能性がある．このように，教師データと検証データが，同じようなクラス分布をもつ場合には，学習時に多義語の影響を抑える効果は更に小さかった．

このように，今回想定したクラス判定タスクにおいては，問題 (1) の影響は大きくないため，タグ付きコーパスを用いて完全に語義を分けて学習を行う必要はなく，タグなしコーパスでも同等の精度の学習が行える．また，教師データを与える際には，多義語を除く必要はなく，多義語を含めて表記とその表記が所属しうるクラスをなるべく多く指定し，また，多義語の場合は，同じ表記を負事例に含めないで学習することが，精度向上につながる．以下の実験では全て，表記特徴量法で最も精度の高かった (b-NN)（全語義，負事例に含めない）を利用する．

問題 (2) に対する検証結果として，表 4.4 に，推定処理として，全表記集合を対象とした場合と，多義語だけを対象とした場合の表記特徴量法，語義特徴量法の各検証セットにおける 11 点平均補完適合率を示す．表 4.4 を見ると，全表記集合を対象とした場合の (m) 表記特徴量法の (l) 語義特徴量法に対する精度の低下は約 1.4 パーセントポイントとわずかであるが，対象を多義語だけに限定した場合の (o) の (n) に対する精度の低下は，約 9.85 パーセントポイントとかなり大きいことが分かる．表 4.4 から 3 つの評価セットで全て同様の傾向が表れている．4.4 節で詳しく述べるが，表記特徴量法では，多義のある表記が頻度の高い語義と頻度の低い語義をもつ場合に，頻度の低い語義を正しく判定できないという問題があり精度が低下していた．

以上に述べたように，問題 (1) については，学習器の汎化能力によって多義語のもつ特徴量の影響は軽減されるが，多義語が推定対象の場合は，問題 (2) のために，精度の低下が大きい．そのため，語義特徴量法（推定処理）のように，推定時に対象となるクラス以外の文脈情報を除外して扱うことが重要であるといえる．

4.4　表記出現特徴量法

4.3 節の実験において，問題 (2) が多義語の推定精度に大きな影響を及ぼしていることを示した．語義特徴量法のように，推定時に，対象となるクラスとは関係のない語義の文脈情報を取り除くことが理想的であるが，今回のクラス判定タスクは，表記の語義を推定するタスクなので，関係のない語義をあらかじめ除外して特徴ベクトルを生成することは原理的に不可能である．そこで，“表記に多義があったとしても，表記の出現する個々の文脈においては，通常，いずれか 1 つの語義しかもたない” ことに着目する．例えば，‘love letter’ に多義があったとしても，“love letter の歌詞は良い” という文脈においては，‘音楽名’ としての語義だけしかもたない．そして，この性質を利用し，表記ごとに文脈情報を全て合成して扱うのではなく，ある表記の各出現に対応する個々の文脈から特徴ベクトルを生成して推定を行い，出現ごとに得られたスコアを合成することによって，表記の所属スコアを計算する手法を提案する．ここで，対象とするクラス以外の語義の文脈には低いスコアが付与されることが予想されるため，推定結果を合成する際に上位のスコアだけを利用することで，対象クラス以外の語義の文脈情報の影響を軽減させる．本提案手法を，表記出現特徴量法と呼ぶ．

4.4.1　手法概要

学習処理は，表記特徴量法（学習処理）と全く同じ方法を用いる．これは，4.3 節において，学習時における問題 (1) の影響はあまりないことが確認でき，また，出現ごとに正しく語義を指定することは，タグ付きコーパスを用意することと同義であり，大きなコストがかかるからである．推定処理を次に示す．

- 表記出現特徴量法 (推定処理)

  (1) 入力として表記 f，全クラスの判別モデル，タグなしコーパスを与える．

  (2) コーパス中で表記 f が出現する個々の文脈情報を取得し，各 1 回の出現 o に対して 1 つの特徴ベクトル V_{fo} を生成する．このベクトルを表記出現特徴ベクトルと呼ぶ．

  (3) 表記出現特徴ベクトル V_{fo} を各クラス c の判別モデルに入力し，表記 f の出現 o についてのクラス c への所属スコア s_{fco} を得る．

  (4) 表記 f ごとに，得られた所属スコア s_{fco} を合成することで，表記 f のクラス c への所属スコア S_{fc} を求める．

  (5) 所属スコア S_{fc} が閾値を超える全てのクラスに属するものとして，辞書に表記 f を追加する．

ステップ (4) において，スコアを合成する方法は，いくつかのものが考えれる．全ての出現に対して推定されたスコアの合計値や平均値を計算する方法では，多義語の場合に，対象クラスではない語義の文脈を利用することとなってしまう．また，各出現を見れば人間はその語義を判定することができる場合も多いので，最大値は，有効な方法であると考えられるが，1 つの値だけを用いるため外れ値の影響を受けやすい．一方，多義語の場合，正

解クラスの語義で出現する文脈に対しては，不正解クラスの文脈と比べて高いスコアを付与できると予想されるため，正解の文脈から得られたスコアは，スコアの上位に偏って分布すると考えられる．そこで，推定スコアの上位だけを用いた次の 2 つの手法を提案する．

q 合計法： 推定されたスコアを降順にソートし，上位からの割合が q となる順位までのスコア s_{fco} の合計を求め，これを所属スコア S_{fc} とする．

$$S_{fc} = \sum_{o \in Q_f} s_{fco} \tag{4.2}$$

q 平均法： 推定されたスコアを降順にソートし，上位からの割合が q となる順位までのスコア s_{fco} の平均を求め，これを所属スコア S_{fc} とする．

$$S_{fc} = \frac{\sum_{o \in Q_f}(s_{fco})}{|Q_f|} \tag{4.3}$$

ここで，Q_f は，スコアの上位からの割合が q に対応する表記 f の全出現を表す集合である．

このように，これらの手法では，表記 f があるクラス c に属するかどうかを判定する際に，スコアの上位からの割合が q のものだけを用いることによって，c に関する語義で用いられていると考えられる文脈を優先的に選択して所属スコアの計算ができる．

4.4.2　評価結果

表記出現特徴量法と，そのバリエーションである q 合計法，q 平均法の効果を検証するために，次の手法を比較する．

(a) 語義特徴量法（推定処理）：4.3 節と同様に，推定対象の語義を特定可能な現実にはない理想的な状態に相当する．

(b) 表記特徴量法（推定処理）：4.3 節と同様に，この手法がベースラインとなる．

(c) 表記出現特徴量法：(c-1)q 合計法，(c-2)q 平均法

学習処理は，4.3.2 節の (b-NN) 表記特徴量法（全語義，負事例に含めない）を用いた．その他の実験方法は 4.3.2 節と同様とした．

図 4.1，図 4.2 に，q 合計法，q 平均法について，q の値を変化させたときの 11 点平均補完適合率の変化を示す．ここで，図 4.1 は全表記集合，図 4.2 は多義語だけに検証データを限定したときの評価結果である．比較のため，語義特徴量法，表記特徴量法の 11 点平均補完適合率（※ q によらず一定値）をあわせて示す．q 平均法は，全表記集合，多義語だけのいずれの場合でも q の値を増加させたときに精度が次第に高くなり，表記特徴量法の精度を超えるが，更に，増加させると精度が下がっている．このように，スコアが上位のある程度の割合の文脈情報だけで推定を行う手法が有効であるといえる．

また，q が 0 の場合は最大値をとる手法となるが，この場合は表記特徴量法よりも精度が低い．これは最大値の場合，1 つの表記に対してたった 1 つの出現だけからスコアを推定することになるので，ノイズなどに弱いためと考えられる．q 合計法では，多義語だけ

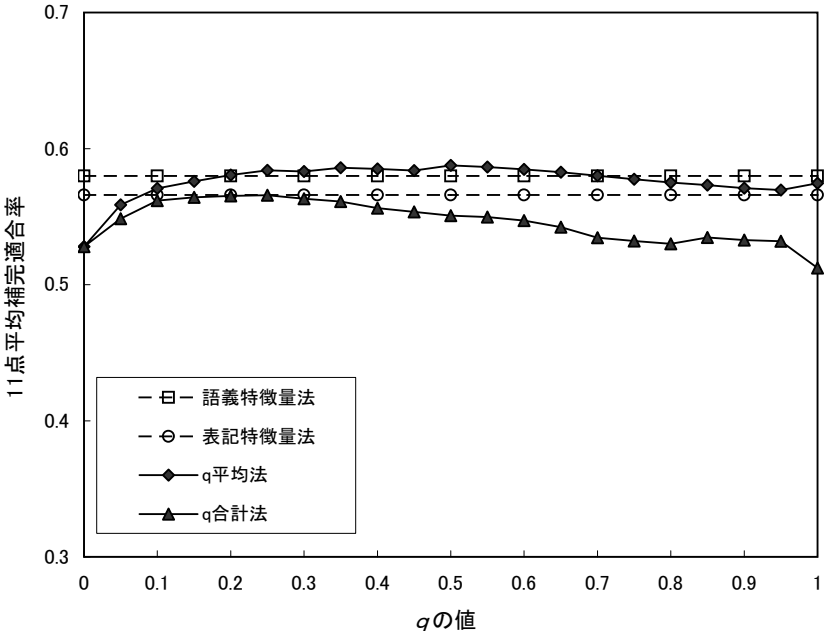

図 4.1 q を変化させたときの精度 (全表記集合)

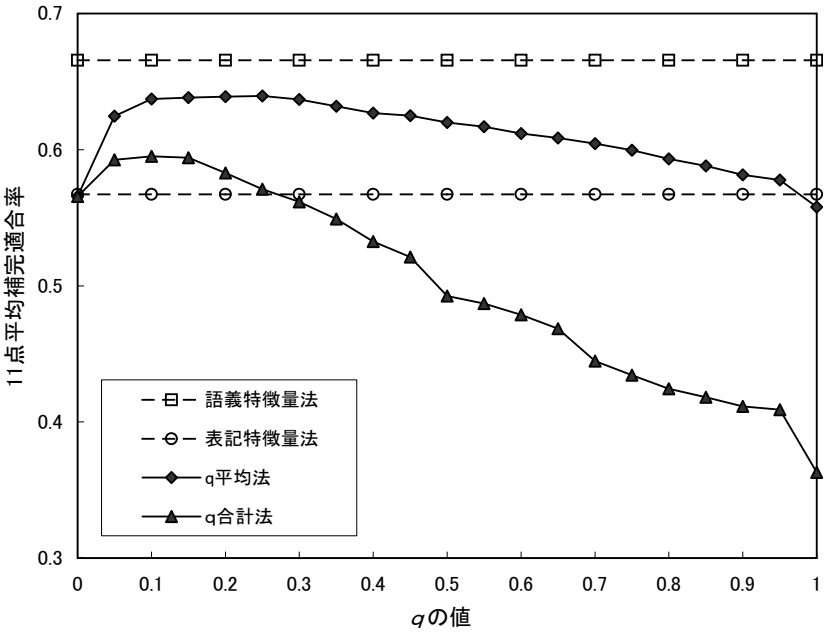

図 4.2 q を変化させたときの精度 (多義語だけ)

に限定した場合は精度の向上が見られたものの，全表記集合については最大の精度となるところでも表記特徴量法とほぼ同等であった．これは，q 合計法は頻度の影響を強く受けるため，不正解の文脈情報を多数含む場合に，これらのスコアが必要以上に低くなってし

表 4.5 提案手法における q の値と精度の比較

	11 点平均補完適合率				q の値		
	全表記集合		多義語だけ		全表記集合		多義語だけ
	最大	推定	最大	推定	最大	推定	最大
(c-1) q 合計法	56.58	56.36	59.77	58.04	0.2500	0.2066	0.0700
(c-2) q 平均法	58.77	**58.43**	63.97	**62.32**	0.5000	0.4466	0.1900
(a) 語義特徴量法	-	57.98	-	66.56		-	
(b) 表記特徴量法	-	56.58	-	56.71		-	

まったことが原因である．

次に，q の値の最適化に関して検証を行うために，各評価セットについて q を次のように割り当てる．

(1) 各検証の対象とする評価セットについて，その評価セット以外の 2 つのセットで，それぞれ q を変えながら全表記集合での精度を測定し，最大となる q を得る．

(2) これら 2 つの q の平均値を，検証の対象とする評価セットの q とする．

この手順では，自セットの検証データを用いずに q を定めている．このようにして求めた q の推定値と，その際の精度について，表 4.5 に語義特徴量法，表記特徴量法との比較を示す．また，このようにして求めた q の値を用いて表記のスコアを計算したときの，再現率と適合率の関係を図 4.3，図 4.4 に示す．ここで，図 4.3 は全表記集合，図 4.4 は多義語だけに検証データを限定したときの評価結果である．更に，全ての対象クラスごとの表記特徴量法，q 平均法，語義特徴量法の 11 点平均補完適合率を表 4.9 に示す．

表 4.5 を見ると，q 平均法の推定した q を用いた場合の多義語の精度（62.32%）は，最大値となる q を用いた場合と比べて，約 1.65 パーセントポイントの低下があった．しかしながら，表記特徴量法と比べて約 5.61 パーセントポイントの精度向上という高い水準にあった．また，全表記集合についても，最大値と比べて若干の精度低下があったものの，理想的と考えていた語義特徴量法とほぼ同じ精度であった．このように，q の値が多少変化しても大きな精度低下はないため，q の値は全探索などの手法によって妥当な値を得ることができると考えている．

全表記集合を推定の対象とした図 4.3 を見ると，精度の差は小さく大きな傾向の違いは見られなかった．一方，多義語だけに推定対象を限定した図 4.4 を見ると，再現率の低い部分（スコアの高いものが集まっている部分）に関しては，表記特徴量法と表記出現特徴量法（q 平均法や q 合計法）の精度の差は小さい．しかしながら，再現率が高くなるにしたがって，特に q 平均法が，表記特徴量法と比べて精度が高くなっている．そこで，この理由を検証するために，ある表記 f が，推定対象クラス c の語義で出現する割合を，表記出現率 (f, c) とし，次のように定める．

$$\text{表記出現率 } (f,c) = \frac{\text{推定対象クラス } c \text{ の語義での表記 } f \text{ の出現数}}{\text{表記 } f \text{ の総出現数}}$$

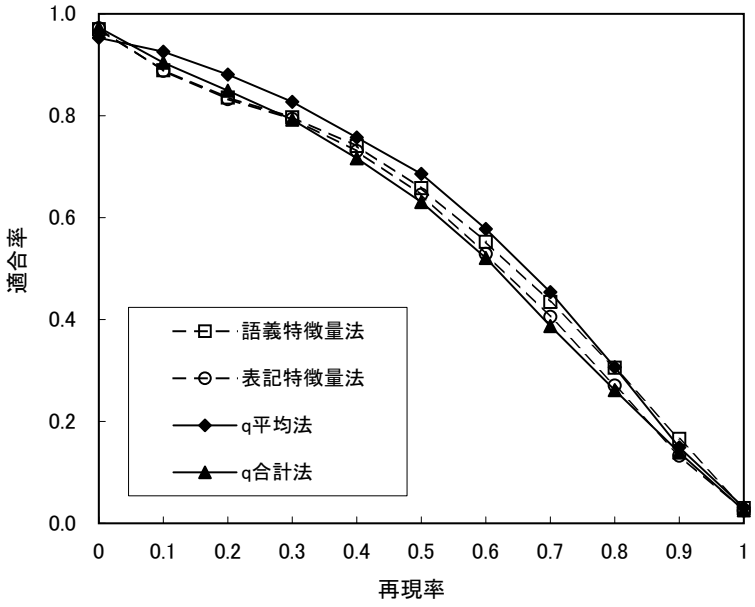

図 4.3 再現率と適合率の比較（全表記集合）

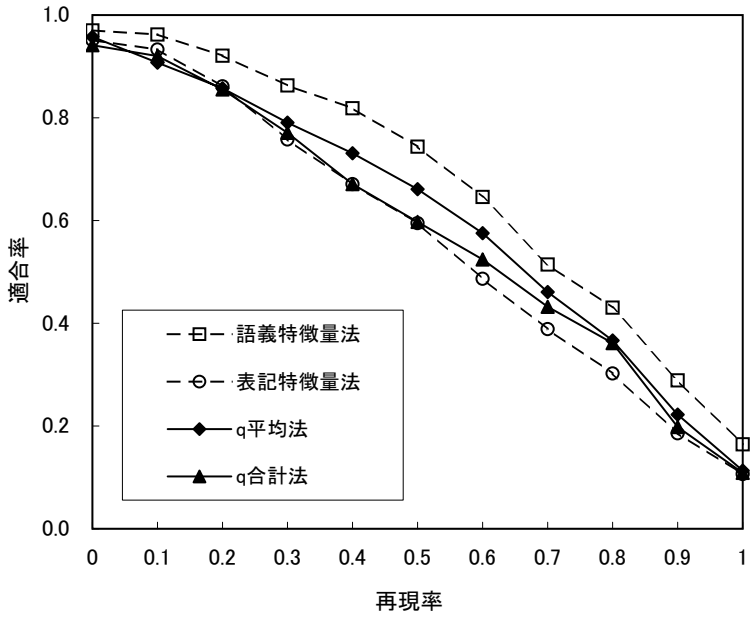

図 4.4 再現率と適合率の比較（多義語だけ）

表記 f が多義をもたない場合は，表記出現率は，つねに 1 であり，また，負事例の場合は，表記出現率はつねに 0 となる．表記出現率が大きくなるにしたがって，表記 f のクラス c の語義が，相対的な頻度の小さい副次的な語義から，頻度の大きい主要な語義を表すようになる．

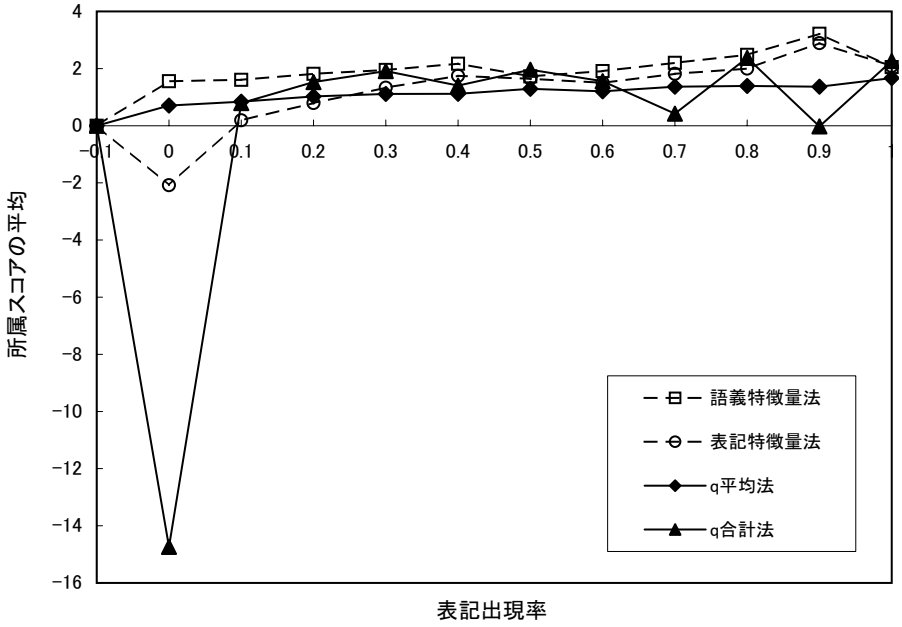

図 4.5 表記出現率と所属スコアの関係

全クラスの全表記について，表記出現率と各手法による所属スコアの関係を求めた (図 4.5)．表記出現率 x が正のものについては，0.1 刻みとなる a について，$a \leq x < (a{+}0.1)$ となる x の所属スコアの平均を x 軸の値が a の位置にプロットした．ただし，不正解の場合の表記出現率は，‘0.0’ と区別ができるように，-0.1 の位置にプロットした．ここで，y 軸は，手法ごとに所属スコアの値域に差があったので，不正解の場合の所属スコアの平均を基準値と考え，各所属スコアからこの値を減算した値とした．本グラフにおいては，-0.1 は不正解で，0.0 以上は正解であるため，理想的には，-0.1 と比べて 0.0 が跳ね上がることが望ましい．

図 4.5 を見ると，表記特徴量法は，0.0 付近で，非常に小さい値（不正解よりも低い値）を付与していることが分かる．このため，表記特徴量法では，表記出現率 (f, c) が低い副次的な語義 c に対しては，正解と不正解を判別することができない．また，q 合計法に関しても同様に，0.0 付近で表記特徴量法以上に小さい値となった．これは，対象以外のクラスの語義の頻度が高く，それらを合計してしまったため，この影響を強く受けたためである．これに対して，q 平均法では，表記出現率が 0.0 の位置からある程度高いスコアを付与している．このように，q 平均法では，副次的な語義に対しても不正解事例と判別できるだけのスコアを付与していることが分かる．仮に，推定時に使用するタグなしコーパスの量を増やした場合，いずれの手法でも用語の網羅性を向上させることができる．しかしながら，表記出現率（語義の相対頻度）は変わらないため，表記特徴量法では，相対頻度の低い語義は依然として抽出することができない．一方，q 平均法では，コーパスの増加にともない使用されることが少ない副次的な語義であっても，低頻度で出現することが予想されるため，これらの副次的な語義も含めて全ての語義を網羅的に獲得できると考えている．

表 4.6 適合率上位の再現率（クラス平均）

	全表記集合			多義語だけ		
適合率	q 平均法	表記特徴量法	語義特徴量法	q 平均法	表記特徴量法	語義特徴量法
100	11.36	8.34	8.48	32.12	26.60	29.96
90	**28.84**	25.70	26.56	39.37	35.22	40.94
80	40.59	37.42	38.66	47.10	42.76	53.16

表 4.7 適合率上位の再現率（表記平均）

	全表記集合			多義語だけ		
適合率	q 平均法	表記特徴量法	語義特徴量法	q 平均法	表記特徴量法	語義特徴量法
100	37.64	27.80	23.86	15.55	6.43	4.88
90	**55.72**	52.23	54.11	58.81	55.92	56.02
80	65.24	62.07	68.52	67.89	65.53	66.27

理想的な状態である語義特徴量法と比べると表記出現特徴量法のスコアは全体的に若干低い．この理由は，語義特徴量法では複数の文脈情報を合成しているため，多くの特徴量を総合的に利用して推定を行っているのに対して，表記出現特徴量法では各出現から特徴ベクトルを生成しているため，少ない特徴量で推定を行っていることにある．表記出現特徴量法の精度が語義特徴量法と比べて 3 つの評価セットのいずれでも 2 パーセントポイント以上低かったのは，‘Compound’ だけであり，1 パーセントポイント以上低かったのは，‘Movie’，‘Music’，‘Station’ の 3 つであった．各対象クラスの上位 20 件の中でエラーとして抽出してしまった表記が最も多かったクラスは，それぞれ，‘Compound’ では ‘Food_Other（頑丈元気，能力第一，イチジクなど）’，‘Movie’ では俳優や監督を表す ‘Person（フランク・キャプラ，イヴ・モンタン，マーなど）’，‘Music’ では音楽アルバムを表す ‘Product_Other（ヒストリー，ホワッツ・インサイド，ミラクル・オブ・ソウルなど）’，‘Station’ では ‘City（琴平，摩耶，小山市など）’ であった．

このような似た概念を判別するためには，各出現から得られたスコアの平均を用いるだけでは不十分であり，複数の文脈から得られた情報をうまく活用する必要があると考えている．今回は，文脈として個々の表記が出現する文に限定して議論を進めた．しかしながら，段落や同一の文書内などの少し広い範囲では，同じ語義を共有する [24] と考え，この範囲での出現に対して，合成された同一の特徴ベクトルを付与する方法も考えられる．このように，対象クラス以外のクラスの語義の影響を少なくしつつ，なるべく多くの文脈情報を活用できる手法は今後の課題である．

本手法を，実際に，固有表現辞書への用語の自動追加に適用する場合，非常に高い精度でクラス判定ができる範囲だけで利用することが考えられる．そこで，適合率が 100%，90%，80%となる再現率を求めた (表 4.6, 表 4.7)[††]．表 4.6 は，各クラスについて適合率が p%以上となる最大の再現率を求め，これを全クラスで平均（マクロ平均）したもので

[††]図 4.4 の再現率–適合率グラフから読み取れる値（同一再現率における適合率の平均）とは異なる．

ある（クラス平均と呼ぶ）．一方，表 4.7 は，各クラスについて，適合率が，p%以上となる最大の正解数を求め，これを全クラスで合計したうえで，全正解表記数で除算（マイクロ平均）したものである (表記平均と呼ぶ）．このように，各クラスについて，最適な閾値を設定することによって，90%の精度の辞書を，クラス平均では再現率 28.84 %，表記平均では再現率 55.72%で生成できた．表記平均がクラス平均と比べて再現率が高いのは，語彙数の多い Person などの精度が高かったことが主な原因である．ここで，閾値の設定が問題となるが，各クラスについての所属スコアのバラつきが 3 つの評価セット間で少なかったことから，検証セットを用いて妥当な値を設定することができると考えている．このように，各クラスに対して正しい閾値を設定することによって，自動追加がある程度できる．また，これより高い再現率が必要な場合は，ランキング結果を上位から順に見て，人手で辞書に登録するかどうかを判定するプロセスが必要となる．

4.5　固有表現辞書の自動構築に関するまとめ

本章では，文書中の対象物を表す固有表現を抽出し，そのクラスを判定することの重要性について述べ，その実現のためには，用語及び語義の網羅性をもつ固有表現辞書を自動構築することが有効であることを述べた．そして，このような辞書の自動構築を目指し，表記とクラスのペアが教師データとして与えられたときに，タグなしコーパスから文脈情報を収集し，これを合成することによって判別モデルを学習し，推定を行う手法における多義性の影響を検証した．この結果から，学習処理では，教師データの中に多義語が混入することによって，他のクラスの特徴量が含まれてしまう可能性があるが，多義語が通常の出現分布の範囲内であるならば，学習器の汎化能力によって，これらの特徴量の影響が軽減される．そのため，精度への影響は小さく，タグなしコーパスでもタグ付きコーパスと同等の精度で学習が行えることを示した．また，推定処理では，推定の対象が多義語の場合，頻度の高い語義の特徴量が支配的になってしまい，頻度の低い語義に対する特徴量の影響が失われ，クラス判定が難しくなることを示した．

推定時の多義性の問題に対処する手法として，表記が出現する個々の文脈に対して特徴ベクトルを生成して推定を行い，その推定結果であるスコアを合成する手法を提案した．特に，スコアの低いものは，対象クラス以外の文脈と考え，上位からの割合が q となるスコアの平均を用いることによって，最大値や上位の合計値を用いる方法と比べて精度が高くなることを示した．多義語だけを対象とした評価セットで，表記ごとに特徴ベクトルを生成する従来手法と比べて 11 点平均補完適合率で，約 5.61 パーセントポイントの精度の向上を確認した．本手法は，特に相対頻度の少ない副次的な語義に対して有効であり，この結果から，タグなしコーパスの量を増やすだけで，従来手法では難しかった多義語のもつ複数の語義を網羅的に獲得できると考えている．

このような固有表現辞書があると，辞書に存在する用語を用いた単純なパターンマッチによって，文書中に存在する固有表現を容易に抽出することができる．この結果，対象物の抽出精度は飛躍的に高められると期待できる．ただし，各用語が多義性をもつ場合，このような辞書だけでは，対象物のクラスを一意に定めることはできない．今回，提案した手法の所属スコアは，表記出現率に対して単調増加する傾向にあったため，今後，提案手法を文書中の用語の曖昧性解消の事前確率として活用する方法を検討していきたい．

表 4.8 対象クラスと用語例

クラス名	用語数		多義語率	多義語の例
	全用語	多義語		
Book	623	33	0.0529	アンネの日記，ペスト，環境基本計画
City	2146	159	0.0740	小倉，観音寺，伊勢
Company	1725	224	0.1298	松下，鹿島，アウディ
Compound	126	25	0.1984	アルコール，ニコチン，メタン
Conference	536	38	0.0708	円卓会議，国連総会，世界女性会議
Country	324	36	0.1111	韓国，スリランカ，日
Event_Other	545	164	0.3009	北方領土，PKO，ドーピング問題
Fish	82	34	0.4146	アジ，さけ，メバル
Flora	375	190	0.5066	イチゴ，きゅうり，にんじん
Food_Other	483	255	0.5279	きゅうり，さけ，カキ
Game	901	16	0.0177	甲子園，バルセロナ，日本リーグ
Goe_Other	725	39	0.0537	日本芸術院会館，国会議事堂，戸田城
Government	906	52	0.0573	円卓会議，建設省，警視庁
International_Organization	225	23	0.1022	世界女性会議，国連総会，NAFTA
Mollusc_Arthropod	41	20	0.4878	エビ，カキ，タニシ
Movie	308	29	0.0941	アンネの日記，シカゴ，戒厳令
Music	251	25	0.0996	四季，白鳥の湖，フィガロの結婚
Organization_Other	778	46	0.0591	観光協会，カンボジア仏教会，全国連
Person	11743	238	0.0202	さくら，小笠原，松下
Political_Organization_Other	301	34	0.1129	カレン民族同盟，平成会，ハマス
Position_Vocation	2971	36	0.0121	キャプテン，捜査一課長，三役
Pro_Sports_Organization	221	52	0.2352	ダイエー，バルセロナ，京都
Product_Other	950	192	0.2021	PKO，ニフティサーブ，失業問題
Province	273	46	0.1684	京都，三重，秋田
Public_Institution	506	16	0.0316	国会議事堂，大法廷，伊丹
Religion	86	20	0.2325	ハマス，カレン民族同盟，仏教青年会
School	1059	277	0.2615	京都大，天理，小倉
Show	212	27	0.1273	ベルサイユのばら，マクベス，白鳥の湖
Sports_Facility	225	23	0.1022	甲子園，西武，ナゴヤ
Sports_Organization_Other	752	460	0.6117	バルセロナ，ロケッツ，ダイエー
Station	281	37	0.1316	中野，渋谷，仙台
平均	989.67	92.45	0.1809	

表 4.9 対象クラスの 11 点平均補完適合率

	全表記集合			多義語だけ		
クラス名	q 平均法	表記特徴量法	語義特徴量法	q 平均法	表記特徴量法	語義特徴量法
Book	70.99	71.17	72.44	59.98	38.05	65.97
City	80.36	76.90	78.12	64.73	54.77	73.97
Company	70.72	68.83	69.81	79.84	76.49	80.35
Compound	56.44	57.03	58.84	72.70	73.20	84.93
Conference	67.25	64.02	65.75	61.78	40.31	59.53
Country	40.35	40.14	40.90	56.96	51.67	59.10
Event_Other	60.05	55.55	55.20	97.17	95.13	91.99
Fish	40.01	38.76	40.07	64.38	63.79	62.47
Flora	58.42	58.24	58.94	80.35	82.95	80.33
Food_Other	56.24	54.45	55.20	89.53	87.49	85.46
Game	76.83	74.91	74.81	25.91	20.44	34.97
Goe_Other	47.03	43.31	43.92	32.59	22.76	35.85
Government	62.00	56.96	57.59	56.71	54.28	59.96
International_Organization	46.84	40.12	41.44	84.74	65.14	82.75
Mollusc_Arthropod	28.58	26.75	27.06	55.97	53.99	48.36
Movie	82.29	82.25	84.89	78.54	73.10	84.21
Music	53.95	54.21	56.04	64.67	58.81	61.57
Organization_Other	42.44	39.32	40.31	49.19	33.57	50.20
Person	93.06	92.44	92.88	75.45	63.28	77.40
Political_Organization_Other	55.43	48.51	50.06	71.68	68.59	70.04
Position_Vocation	77.34	75.37	75.33	58.38	41.95	59.13
Pro_Sports_Organization	43.24	46.26	48.55	42.38	48.54	57.90
Product_Other	46.97	42.94	43.15	89.45	82.68	81.54
Province	58.73	55.49	56.23	23.99	38.13	42.38
Public_Institution	61.43	59.44	59.91	37.27	30.52	45.29
Religion	11.25	14.06	13.86	36.14	52.58	34.32
School	74.46	72.33	73.73	78.47	80.88	80.93
Show	54.35	54.49	58.06	63.03	54.22	79.23
Sports_Facility	50.91	52.96	57.00	43.55	28.21	65.90
Sports_Organization_Other	60.80	55.15	61.19	89.78	84.22	86.81
Station	82.42	81.70	86.05	46.75	38.23	80.42
平均	58.43	56.58	57.98	62.32	56.71	66.56

第5章　動的なリレーション生成

第4章のような固有表現辞書の自動構築技術が高いレベルで実現できたとしても，対象物となる固有表現は日々生成されるため，文書中から全ての対象物を完全に抽出することはできない．一方，情報集約タスクでは，ある対象物となるキーワードを入力とし，そのキーワードに関する情報を集約するということがしばしば行われる．この場合，入力されたキーワードを利用すれば，文書から単純な文字列マッチによって対象物を抽出できるが，大量の文書に対して全ての解析処理を，キーワードの入力後に実行するとオンラインでの処理時間が掛かり過ぎてしまう．本章では，事前処理で情報要素タプルの一部の属性を抽出しておき，キーワードが入力されると，入力キーワードとこれらの事前抽出された属性とを結びつける処理だけをオンラインで実行することで，高速に情報要素リレーションを生成する手法を提案する．提案手法を実際に評判情報のリレーション生成処理に適用することで，その有効性を示す．

5.1　情報要素リレーションの生成における課題

第3章の3.4節で述べたように，対象物を表す固有表現は，未知語となることが多く，文書中から抽出することが難しい．一方，情報集約タスクは多くの場合，分析の対象となるキーワードの入力によって開始される．例えば，ある製品名に関連する評判を取得したい場合，通常，製品名はユーザによって入力される．このようなタイプの情報集約タスクでは，入力キーワードの文字列や形態素列を文書中から探索すれば良いので，情報集約タスク実行時に文書中の対象物を抽出することは容易である．文献 [48] では，ユーザがキーワードを入力した時点で，商用の検索エンジンを用いて文書を取得し，仮想的なリレーションを生成する手法を提案している．この手法では，未知語にも対応することができるが，問合せ時の文書解析に多くの時間がかかり，オンラインの情報集約タスクへの適用は困難である．

5.2　課題解決へのアプローチ

評判情報を集約するタスクにおいて，“製品Aを購入した．画面がとてもきれい” という文章から評判情報を抽出する例を考える．まず，‘きれい’ という評価表現は辞書やルールとのマッチングによって抽出でき，評価表現からの係り受け関係を用いて，‘画面’ という評価属性も抽出できる [34]．このように，これらの抽出は，対象物を表すキーワード（製品A）とは独立な処理であるため，対象が未知語であるかどうかにかかわらず，オフラインで実行できる．文献 [40, 71] では，同様の考察から，対象物，評価属性，評価表現の3項関係の抽出を，対象物に依存しない ‘評価属性と評価表現の関係’ の抽出と，依存する ‘対

象物と評価属性の関係' の抽出とに分離できることを示し，'評価属性と評価表現の関係' をオフラインで事前抽出することで，オンラインの処理時間を短縮する手法を提案している．

同様に，あるキーワードに関連する将来を予測した情報を集約するタスク [29] においても，相対的な日時表現（例：10 年後）が示す時刻の特定や，各日時表現と紐付けられるイベントを表す特徴ベクトル (周辺テキストの単語集合) の生成などの処理は，対象物となるキーワードとは独立な処理である．

このように，情報要素タプルを生成する処理は，多くの場合，対象物を表すキーワードに依存する処理と，依存しない処理とに分離することができる．そこで，依存しない処理を事前にオフラインで実行し，何らかの形でインデックスしておき，これらと入力キーワードとを結びつける処理だけを，オンラインで実行するという枠組みによって，高速なリレーション生成処理を実現する．

5.3　情報要素リレーションの動的生成手法

事前処理で対象物を表すキーワードが抽出できない場合でも，入力キーワードに関連する情報要素タプルの集合を高速に生成するために，次の (S1)，(S2) の処理を行う（図 5.1）．この処理を動的タプル生成と呼ぶ．

(S1) **事前タプル生成：**事前処理として，文書集合が入力されると，各文書を解析し，対象となるキーワードとは独立に抽出できる情報要素属性，及び，処理 (S2) で必要な付加情報を抽出する．このように対象となるキーワードに依存する情報要素属性が欠落した情報要素タプルを，部分情報要素タプルと呼ぶ．図 5.1 の例では，入力文書中の対象物を表す '製品 A'，'製品 B'，'製品 C' は，未知語であるため抽出できず，<評価属性，評価表現，評価表現の出現位置，部分スコア，文書 ID>からなる部分情報要素タプルを生成している．ここで，部分スコアは，例えば，各評価表現自体の重みや，評価表現と評価属性の係り受け関係などのルールによって決定される部分情報要素タプルが成立するかどうかの確信度を表す．また，部分情報要素タプルは，文書中に出現する情報要素タプルの候補数分だけ生成する．例えば，評判情報の場合は，文書中の全評価表現の各出現に対して，1 つの部分情報要素タプルを生成する．抽出された全ての部分情報要素タプルを，抽出元の文書と関連付けて保存し，(S2) の処理で利用する．

(S2) **動的タプル補完：**オンライン処理として，キーワードが入力されると，(1) 保存された部分情報要素タプルが入力キーワードと結合されて情報要素タプルとなるかどうかの確信度を表すタプルスコアを計算する．この際，入力キーワードを含む文書から抽出された部分情報要素タプルだけを，スコア計算の対象とする．また，入力キーワードの文書中の出現位置などが利用できるように，保存された抽出元の文書も取得し利用する．(2) タプルスコアの上位，又は，閾値で足切りを行い，入力キーワード (図 5.1 の製品 A) を対象物としてセットした情報要素タプル (以下，完全情報要素タプル) を生成する．タプルスコアは，例えば，次式によって計算する．

$$tuple\_score = \alpha \frac{1}{\mid p_e - p_s \mid + \beta} + s_{part} \tag{5.1}$$

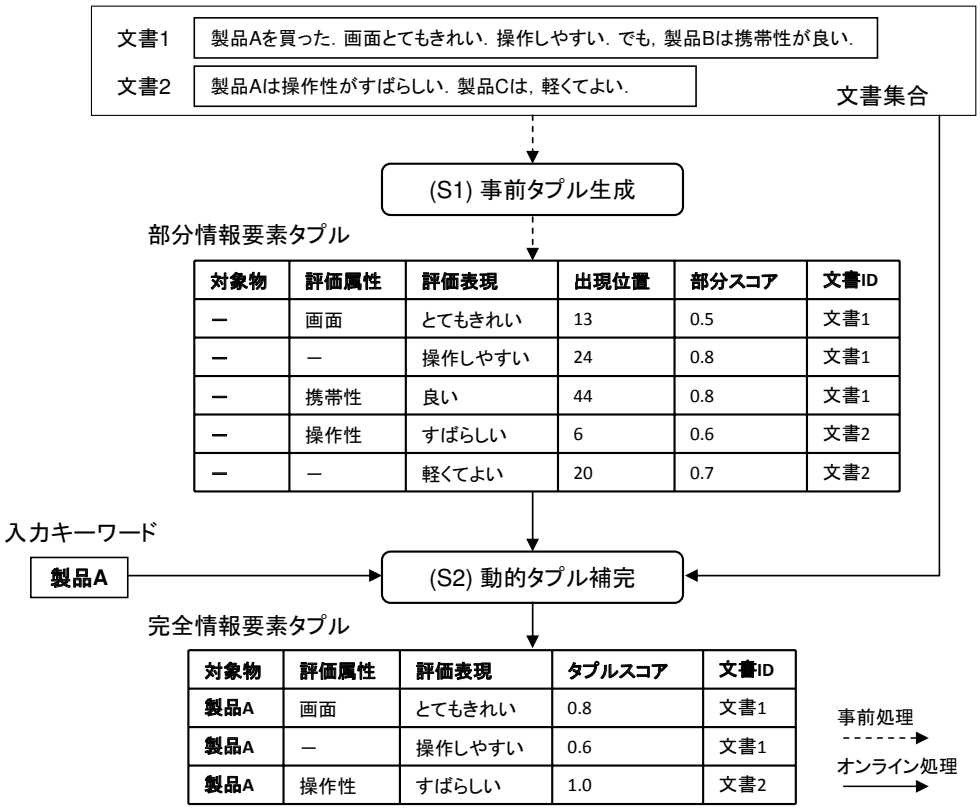

図 5.1 動的タプル生成

ここで，p_e，s_{part} は，それぞれ部分情報要素タプル内の評価表現の出現位置と部分スコア，p_s は，入力キーワードの対象文書中での出現位置，α は，相対位置と部分スコアのどちらを重視するかの係数であり，β は，相対位置をスムージングするための係数である．

(S1) における文書の解析処理や，(S2) におけるタプルスコアの計算は，タスクに応じて異なっている．そこで，それぞれ次の関数を外部関数として定義できるようにする．

- **文書解析関数（analyzeDocument）：** (S1) で呼び出され，入力文書を解析し，情報要素タプル集合を返却する関数

  入力： 文書 (タイトル，本文，所属する文書集合 ID，クロール時間などを含む.)

  出力： 情報要素タプル集合*

- **タプル評価関数（calcTupleScore）：** (S2) で呼び出され，“入力キーワードと部分情報要素タプルが結合されて，完全情報要素タプルとなるかどうかの確信度” を表すタプルスコアを返却する関数

*未知語の場合は部分情報要素タプル，後述するように，既知語の場合は対象物がセットされた完全情報要素タプルが返却値に加えられる．

入力： 入力キーワード，本文，部分情報要素タプル
出力： タプルスコア

このように，部分情報要素タプルをオフラインで生成し，これと入力キーワードとを結びつける処理だけをオンラインで実行することで，未知語に対応しながら，高速にタプルを生成できる．

上記に加えて，対象物を表すキーワードが事前に抽出できる場合（既知語の場合）は，完全情報要素タプルをオフラインの文書解析関数の中で抽出し格納しておくことができる．この場合，オンライン処理では，入力されたキーワードと，格納された対象物を表す用語との一致によって完全情報要素タプルを取得する．この構成では，対象物と部分情報要素タプルとを結合する処理はオフラインで実行できるので，時間が掛かるけれども精度の高い処理を採用することもできる．このように，未知語に対しては部分情報要素タプルから動的に生成したタプルを，既知語に対しては事前抽出された完全情報要素タプルを用いることによって，既知語に対する精度を維持したまま，未知語に対応した情報要素リレーションを高速に生成することができる．

5.4　全文検索エンジンを用いたインデックス手法

大規模な文書集合の中から，問合せに応じて対象文書を効率的に絞り込み，また，情報要素リレーションの動的生成を行うために，全文検索エンジンを用いる．全文検索エンジンは，次の機能をもつことを想定している．

- 大量の文書の中から，キーワードによって高速に対象となる文書を絞り込むことができる．
- タイトルや本文などの任意の文字列情報を，それぞれ別のカラムに格納し検索ができ，検索時に，格納された文字列情報をそのまま取得できる．

全文検索エンジンへの情報要素タプルの格納方法と，格納された情報のオンラインの検索処理について次に示す．

5.4.1　情報要素タプルの格納方法

オフライン処理として，5.3 節の (S1) は，外部関数の analyzeDocument を呼び出すことで情報要素タプルを生成し，これを全文検索エンジンに格納する．ここで，通常の検索システムでは，各文書を論理的な 1 つのレコードとして格納しているため，レコードの単位を文書とすれば，通常の検索システムや，情報要素のスキーマが異なる複数の情報集約タスク間で，サイズの大きい本文に対する転置インデックスを共有できる．しかしながら，情報要素は各文書に複数出現するため，各情報要素属性を，それぞれ文書単位のレコードのカラムに格納すると情報要素属性間の関係が失われてしまう．例えば，評価表現や，評価属性といった情報要素属性を保存するカラムをそれぞれ文書単位のレコードに単純に追加すると，それぞれのカラムには，複数の評価属性や評価表現の値が格納されるため，どの評価表現と評価属性が対応していたのかが分からなくなってしまう．そこで，次の方法を用いる．

ID	URL (DI)	本文 (DM)	日時 (DD)	Blog site id (BI)	性別 (BS)	年代 (BG)	完全情報要素属性 (IComp)	部分情報要素属性 (IPart)
1	http://host/tanaka/111.html	製品Aを買った. 画面がとてもきれい. …	20100316	host/tanaka	男性	20	s=製品X p=操作 e=簡単 o=好評 cc=0.6 b	n=13 p=画面 e=とてもきれい o=好評 cp=0.5 b n=18 p=電池 e=短い o=不評 cp=0.4 b
2	http://host/tanaka/112.html	…	20100317	host/tanaka	男性	20	s=製品X p=画面 e=クリア o=好評 cc=0.6 b	n=1 p=画面 e=耐え難い o=不評 cp=0.8 b
3	http://host2/suzuki/011.html	…	20100327	host2/suzuki	女性	30		n=10 e=美しい o=好評 cp=0.7 b

図 5.2 情報要素の格納方法

(1) 情報要素の各属性名に対応するプレフィックスを各属性値に付与する.

(2) 各文書に出現する全ての情報要素の全ての情報要素属性を 1 つのカラムに格納する.

(3) 格納順は情報要素の出現順序とし，各情報要素間にセパレータを置く.

また，第 3 章の 3.2 節で述べたように，文書が 1 つの文書集合にしか属さないとすると，文書に対して文書集合が一意に決まるので，レコード数を変えずに，文書集合属性を全て文書に join して格納できる．この結果，例えば，評判情報の場合には，図 5.2 のように，文書属性用カラム (DI，DD)，文書集合属性用カラム (BI，BS，BG)，完全情報要素属性集合のリスト用カラム (IComp)，部分情報要素属性集合のリスト用カラム (IPart) を本文 (DM) とともに，各文書を論理的な 1 つのレコードとして格納する．ここで，各情報要素属性名に対応するプレフィックスは，s：対象物，p：評価属性，e：評価表現，o：評価極性，b：要素間のセパレータを表す．また，cp：部分スコア，n：出現位置はタプル評価関数だけで使用される．cc：タプルスコアは，完全情報要素タプルの確信度を表し，オンライン処理でタプルを選択する際に利用される.

5.4.2　オンラインの検索処理

5.4.1 節で述べた方法で格納された候補となる情報要素タプル集合の中から動的にリレーションを生成する例を次に示す．“製品 A に関する ‘好評’ の評価” を取得したい場合，ユーザは次のように検索条件を指定する.

```
（IS=’製品 A’）and（IO=’好評’）
```

この式では，対象物（IS）が，‘製品 A’ であり，評価極性（IO）が ‘好評’ の情報要素タプルを検索結果として出力する．詳しい検索条件の指定方法は，第 6 章で述べる.

このような問合せがあると，情報要素リレーションに対する検索条件を，全文検索エンジン用の文書検索式に自動変換する．この際，対象物を表す属性名（IS）が指定された場合は，部分情報要素タプルを取得する文書を絞り込むために，本文（DM）への検索を行う．また，情報要素属性に対する検索条件にはプレフィックス (s=，o=など) を用いることによって，特定の属性（例：評価極性が好評）をもつ情報要素タプルを含む文書だけに絞込みを行う．例えば，上記の問合せ式の検索条件から，次の文書検索式を生成する（カラム名は，図 5.2 を参照）.

```
(IComp='s=製品 A' and IComp='o=好評') or (DM=' 製品 A' and IPart='o=好評')
```

この検索式によって，完全情報要素属性集合のリストのいずれかの要素として，‘s=製品 A’ と ‘o=好評’ をともに含むか，又は，本文に ‘製品 A’ を含み，かつ，部分情報要素属性集合のリストのいずれかの要素として ‘o=好評’ を含む文書が検索される．ここで，各文書を論理的な 1 つの単位と扱っているため，

```
(IComp='s=製品 A' and IComp='o=好評')
```

は，情報要素タプルの絞込みではなく，文書に対する絞込みである．そのため，ある情報要素タプルの対象物が ‘製品 A’ であり，別の情報要素タプルの評価極性が ‘好評’ である 2 つの情報要素タプルを含む文書も検索されてしまう．

そこで，検索された文書に対して，次の処理を行い情報要素リレーションを生成する．

(1) 全文検索エンジンから，本文，文書属性，文書集合属性，(完全/部分）情報要素属性集合のリストを取得する．

(2) 各情報要素属性集合に対して，文書属性，文書集合属性を付与し（完全/部分) 情報要素タプルを生成する．

(3) 各部分情報要素タプルに対しては，5.3 節の (S2) の処理を実行し，calcTupleScore を呼び出し，完全情報要素タプルを生成する．

(4) (2) と (3) で生成した完全情報要素タプルの集合に対して，入力された検索条件を適用し，検索結果リレーションとする．

このように，入力された検索条件に合致する情報要素リレーションを動的に生成する．

5.5　動的なリレーション生成の評価

評判分析処理に本手法を適用することによって動的なリレーション生成の効果を検証する．具体的には，(1) 全てをオフラインで処理する方法との比較として，未知語に対応することの効果を明らかにする．(2) 文献 [48] のようなオンラインで全ての処理を行う手法と比べて，どの程度の高速化が実現できるのかを検証する．(3) 今回の実装によって，どの程度の応答速度を実現できたのかの実測値を示す．

5.5.1　評判分析処理

文献 [4, 25] の評判情報を抽出する処理を情報要素タプル生成処理に当てはめると次のようになる．

(a) **基本言語解析:** 入力文書に対して，文区切り，形態素解析，係り受け解析を行い，文境界，品詞，文節，表層格，係り先を付与する．

(b) **評価表現/評価属性抽出:** (a) の結果を入力として，辞書やパターンとの照合などによって，評価表現と，その極性（好評/不評）を抽出する．また，評価表現との係り受け関係などを用いて，各評価表現に対する評価属性を抽出し，部分情報要素タプル (<評価属性，評価表現，評価極性，部分スコア，出現位置>）を生成する[†]．ここで，部分スコアは，5.3 節の処理 (S1) の例にあげたルールによる方法で決定する．

(c) **固有表現抽出:** (a) の結果を入力として，対象物の候補となる固有表現を抽出する．

(d) **対象物抽出:** (a)〜(c) の結果を入力として，各部分情報要素タプルに対して，各固有表現が対象物として結びつくかを判定し，完全情報要素タプル（<対象物，評価属性，評価表現，評価極性，タプルスコア>）を生成する．

この処理を，analyzeDocument，calcTupleScore で書き換えると次のようになる．

analyzeDocument: (a)〜(d) の全ての処理を実行する．ここで，(d) の処理で，対象物が抽出された場合には，完全情報要素タプルを返却値に加える．抽出されない場合には，(b) で生成した部分情報要素タプルを返却値に加える．

calcTupleScore: 入力キーワードを対象物の候補として，(d) の入力となる情報を生成する処理（本文から入力キーワードの出現位置を求めるなど）を実行し，(d) の処理を行う．

以降の節では，この処理フローで動的なリレーション生成を行った場合の効果を検証する．

5.5.2　未知語に対応することの効果

評判分析を行なう実サービスに，実際に一般のユーザが入力したキーワードを用いて，どの程度，未知語の影響があるのかを検証する．測定用のキーワード集合は次の手順で作成した．

(1) 2009 年 11 月〜2010 年 10 月に goo ブログ評判分析機能[‡]に投入されたキーワードの上位 1 万件から約 100 件ごとに全体で 100 語を選択した．[§]

(2) 2010 年の 10 月までにクロールした約 2 億件のブログ記事に対して，評判情報を 1 つでも含む記事を検索し，各キーワードについて最大 100 件の記事を取得した[¶]．ここで，キーワードのうち 2 件はヒット件数（ヒットした記事の件数）が 0 件であったので，分析対象のキーワード数は 98 件，分析対象の文書数は 7,270 件となり，これを評価セットとした．

[†]評価属性は空の場合もある．
[‡]http://blog-hyoban.goo.ne.jp/
[§]ここで，記号及びスペースを含むキーワードは除外した．
[¶]実際には，事前に全ブログ記事に対して評判情報を抽出しておき，評判情報を持つもので絞込みを行い文書を取得した．

評価セットの全キーワードの中で，各キーワードに対する全取得文書の中から 5.5.1 節の (c) の処理を用いて，そのキーワードが固有表現として一度も抽出されなかったもの（未知語相当）は，約 12%であった．また，7,270 文書の中で，対象となるキーワードを固有表現として一度も抽出できなかった文書は，約 34%であった[‖]．このように，オフライン処理だけでシステムを構成すると，約 12%のキーワードに対しては全く評判情報を抽出することができず，約 34%の文書は評判情報の抽出に寄与しない．この結果から，任意のキーワードが投入される情報集約サービスにおいて，未知語に対応することが重要であるといえる．

5.5.3　オンラインで全ての処理を行う手法との比較

5.5.1 節の (a)，(b) の処理は，ユーザの入力となる対象物とは無関係な処理であり，(c) の処理は未知語に対して効果がない．このため，未知語の場合に，オンラインで必須となる処理は，(d) 対象物抽出と (d) の入力を生成する処理だけであり，これが，calcTupleScore の処理として実行される．したがって，本構成においては，これらの処理時間が全体に比べて小さければ，オンラインで全ての処理を行う手法と比べた場合の高速化の効果が大きい．また，5.4.2 節で述べたように，事前抽出した情報要素属性を絞込み条件に用いることによって，キーワード入力時に解析対象とする文書数を減らすことができる．そこで，

検証項目 (i) (a)〜(d) の各処理時間

検証項目 (ii) 絞込み条件によるヒット件数の変化

を測定する．以降の測定では，次のマシンを用いた．

CPU: Intel(R) Xeon(R) CPU 2.27GHz 16core

メモリ: 48GB

OS: Red Hat Linux 5.4

検証項目 (i) に関して，比較のため (d) の処理として，次の 2 つを実装した．

(d-1) **距離スコア法**
5.3 節の (5.1) 式を用いてタプルスコアを計算する手法

(d-2) **関係抽出法**
タグ付きコーパスによる機械学習を用いて，係り受け関係や省略情報などを考慮し，対象物と評価表現の 2 つ組の確からしさを求め，タプルスコアとする手法 [4]

ここで，距離スコア法における (5.1) 式の α と β は，距離スコア法による情報要素タプルの順位付けが，関係抽出法による順位付けに最も近くなるように調整した．具体的には，関係抽出法の順位付けを正解とみなして，距離スコア法の順位付けの pairwise accuracy[38] を計算し，その最大値となる α と β を求めた．関係抽出法は，距離スコア法よりも精度が

[‖]固有表現は文脈によって抽出の成功/失敗があるため，文書を単位とした場合の抽出できなかったものの割合は，キーワードを単位とした場合の割合と比べて高い値となる．

表 5.1 処理時間内訳

	処理時間 (ms)	処理時間内訳 (%)	
		距離スコア法	関係抽出法
(a) 基本言語解析	8.07	21.333	19.515
(b) 評価表現/評価属性抽出	25.98	68.762	62.820
(c) 固有表現抽出	3.60	9.515	8.704
(d-1) 距離スコア法	0.18	0.478	-
(d-2) 関係抽出法	3.70	-	8.959
合計 (d-1) 使用	37.84	-	-
合計 (d-2) 使用	41.36	-	-

高いことが想定されるので，このように求めることで，人手による精度評価を行わなくても，妥当な値が得られると考えたためである．このように求めた α=20，β=30 を以降の実験で用いた．

5.5.2 節の評価セットの各文書に対して，(a)〜(d) の各処理時間を測定した（表 5.1）．表 5.1 に示すように，多くの解析時間は，(b) の評価表現/評価属性抽出処理に費やされている (いずれの手法でも全体の 6 割以上)．キーワード入力時に必須となる (d) 対象物抽出処理の全体における割合は，距離スコア法 (d-1) で約 0.5%，関係抽出法 (d-2) でも約 9%と小さい値となった．ここで，(d-1) の入力情報としてオンライン時に取得が必要なものは，入力キーワードの出現位置だけであり，他の解析処理と比べて無視できるほど小さかった．一方，(d-2) では，係り受け関係などの特徴量も必要となり，(a) の基本言語解析を仮にオンラインで再度実行した場合は，更に，約 19.5%が必要となる．以上のように，動的タプル生成手法を用いると，オンラインで全ての処理を行う場合と比べ，単純な距離スコア法を採用した場合には，約 0.5%，比較的高度な関係抽出法を用い，特殊なインデックスを使用しない場合でも，約 28.5%に処理時間を短縮できた．

本アーキテクチャでは，対象のキーワードが事前抽出できる場合には，完全情報要素タプルを抽出しておき，これを利用する．そのため，オフラインの処理では，時間は掛かるが精度の高い関係抽出法などの手法を用い，オンラインの処理だけは，高速な距離スコア法を用いることもできる．この場合，既知語に関しては関係抽出法によりタプルが得られるので，精度の低下は未知語だけで起こる．そこで，このような構成における精度低下を検証するために，評価セット中の未知語相当の 12 個の各キーワードを対象物とし，そのキーワードを含む文書から抽出した全評価表現に対して部分情報要素タプルを生成した．次に，キーワードごとに，(d-1)，(d-2) を用いて，タプルスコアを計算し，タプルスコアの降順に生成した情報要素タプルを並べた．12 個のキーワードのうち情報要素タプルが，20 件以上抽出された 6 個のキーワード（平均抽出タプル数 333.7 件) について上位 10 件を目視確認した**．これら 6 個のキーワードの上位 10 件の適合率の平均は，関係抽出法で 58.3%，距離スコア法で 50.0%であった．このように，本アーキテクチャでは，精度と速度のトレードオフを考え，用途に応じて最適な手法を選択すればよい．

**正負判定は，http://www.syncha.org/op_tagged_corpus/index.html を用いた．

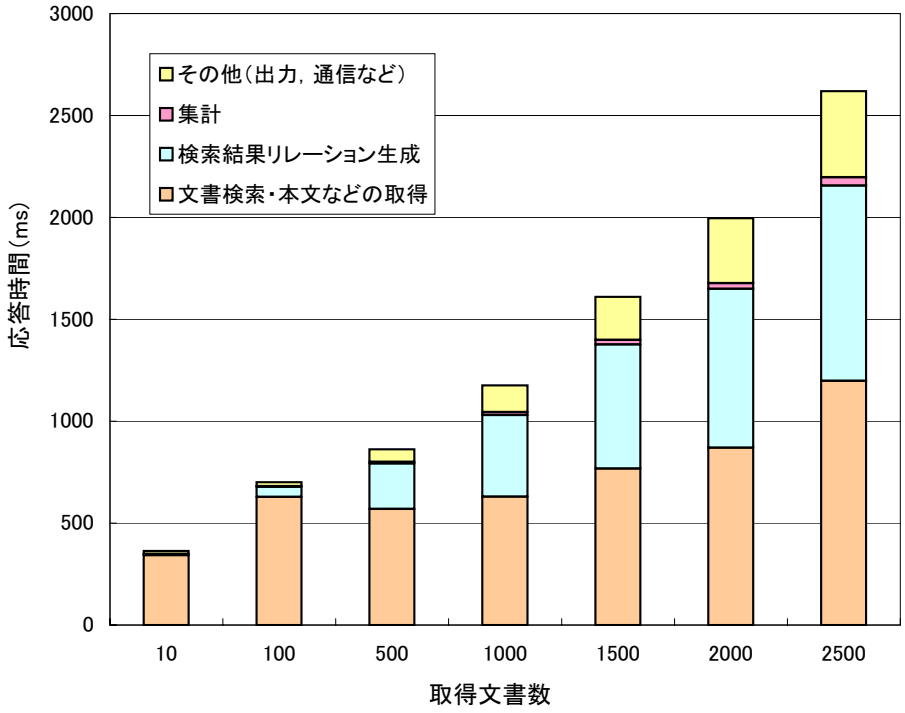

図 5.3 取得文書数を変化させたときの応答時間

検証項目 (ii) に関して，5.5.2 節のブログ記事を対象に，評価セットの 100 キーワードで検索を行い，次のヒット件数を測定した．

(1) キーワードだけの場合のヒット件数

(2) 評判情報を含むもので絞込みを行った場合のヒット件数

(3) 評価極性が好評である評判を含むもので絞込みを行った場合のヒット件数

平均ヒット件数は (1) を 1 とした場合に，(2) で 0.69，(3) で 0.43 であった．このように 5.4 節のインデックス手法を用いるとオンラインで解析の対象となる文書数を効率的に削減することができる．

5.5.4　応答時間

本システムでは，全文検索の転置インデックス [32] を用いているため，文書量の増加に対する応答速度の低下は理論的にはない．しかしながら，取得文書数のオーダで処理時間が掛かるので，どの程度の文書数の解析を実時間で実行できるのかを測定した．評価セットのうち上位 20 件のキーワード (最小ヒット件数 9,264 件) を用いて，5.5.2 節のブログ記事に検索を行い，取得文書数を変化させた時の応答時間とその内訳を測定した (図 5.3)[††]．

[††]各キーワードについて 5 回集計を行い，その平均を計算した．

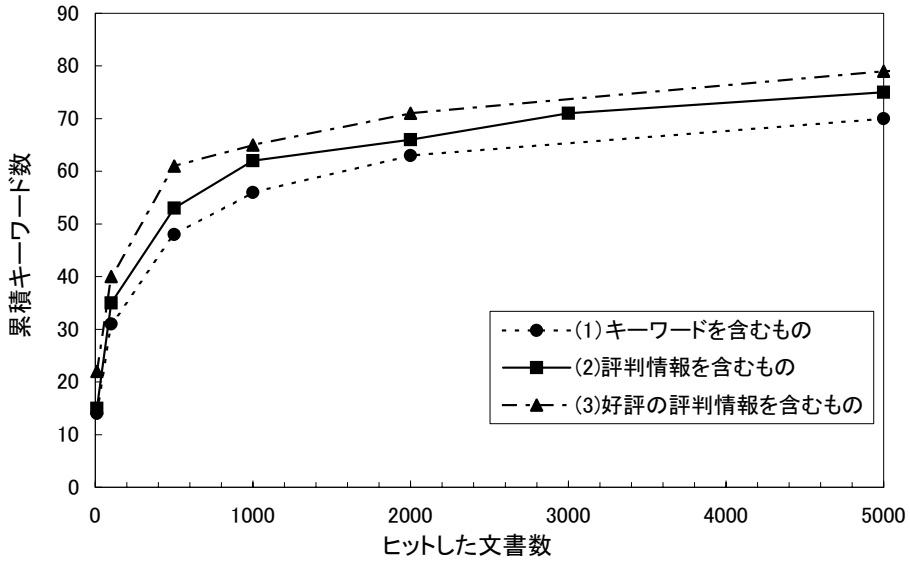

図 5.4 ヒットした文書数と累積キーワード数の関係

ここで，対象物抽出には距離スコア法（d-1）を用いた．図 5.3 より，応答時間の限界を仮に 1 秒とすると，本実装では，約 800 件程度の文書の解析ができる．

一方，評価セットの 100 件のキーワードで検索を行い，ヒット件数 x とヒット件数が x 以下のキーワード数（累積キーワード数）の関係を求めた（図 5.4）．図 5.4 を見ると，実際に入力されたキーワードの出現分布は偏っていて大部分のキーワードのヒット件数はそれほど多くなく，評判情報を含むもので絞込みを行った場合にヒット件数が 800 件以下であるキーワードの割合は，約 60%であった．当然，ブログ記事の総数や実装に依存するが，約 2 億件の記事を対象とした場合に，実際の問合せの約 60%ついて，全数での評判用の情報要素リレーションの生成を実時間（1 秒以内）で実現できた．

5.5.5　動的タプル生成が有効に働くタスクの特徴

動的タプル生成は，情報集約タスクが次の特徴を持っている際に，有効に働く．

(a) 対象物を表すキーワードに依存する処理と依存しない処理を分離でき，対象物を表す入力キーワードと部分情報要素タプルとを結びつけるための処理コストが小さい．

(b) 情報要素タプルや特定の情報要素属性を含む文書の割合が全文書数と比べて少なく，個々の文書中の情報要素タプルの個数が少ない．

(c) 対象物を表すキーワードの出現頻度が小さいものが多いか，サンプリング調査の結果でも許容される．例えば，何らかの比率（好評対不評）や，代表的な記述の集計（評価表現や評価属性の集計）はサンプリング調査でも十分であると考えられる．

評判分析処理では，このような特徴を持っていたため動的タプル生成の効果が大きかった．しかしながら，未知語に対してのより高い精度が必要とされる場合，オンラインの (S2) 動的タプル補完で必要となる特徴量を次のように増やす必要がある．

(1) 分析の対象に依存しないもの
　　例）評価表現や評価属性に紐付く全ての特徴量

(2) 分析の対象に依存するが本文から取得できるもの
　　例）対象物の位置，格情報，係り受け関係

(3) 分析の対象に依存し，本文だけからでは抽出できないもの
　　例）対象物と，評価属性の大規模コーパス中での共起頻度 [34, 72]

(1) に関しては，全てを事前に抽出し，部分情報要素タプルの付加情報（属性の 1 つ）としてもたせることができる．(2) は，キーワード入力時に特徴量の再計算を行うか，何らかの形でインデックスしておくことにより利用できる．例えば，関係抽出法 (d-2) などで利用される基本言語解析の結果を，文書中の出現位置をキーとしてインデックスすることが考えられる．しかしながら，(3) については，キーワードを与えた時点で，外部リソースなどにアクセスする必要が生じるため，本研究の枠組みでは高速化のメリットを得ることは難しい．このように，必要となる応答速度と精度によって，どのレベルまでの特徴量を利用するのかを決定する必要がある．

5.6　動的なリレーション生成のまとめ

本章では，対象物を表すキーワードが入力されるようなオンラインの情報集約タスクにおいて，動的にリレーションを生成する手法について述べた．情報要素タプルの一部は，入力されるキーワードとは独立に生成できることに着目し，情報要素の一部をあらかじめ格納しておき，これと入力キーワードとを結びつける処理だけをオンラインで実行する動的タプル生成手法を提案した．この手法を評判情報のリレーション生成に適用し，未知語に対応しながら，約 800 件の検索された文書を対象とした場合の情報要素リレーションの生成時間を，約 200 秒から 1 秒程度に短縮できることを示した．また，動的タプル生成が有効かどうかは，利用する特徴量に大きく依存することを述べた．

今後，様々なタスクに本手法を適用する際には，オンラインの処理で必要となる特徴量のレベルを明らかにし，通常の全文検索エンジンでは，高速に取得できないようなものについては，新たなインデックス構造を検討していきたい．

第6章　情報集約言語

文書データから情報要素リレーションが生成されれば，SQL などの問合せ言語によって，検索処理及び集計処理を行うことがある程度できる．しかしながら，SQL は，数値や文字列の検索を主な目的として開発された言語であり，情報要素を集約するという目的で必ずしも最適なものではない．本章では，情報集約言語がもつべき要件として，(a) 階層的な内訳をもつ集約結果の生成，(b) 表記ゆれなどに対応した柔軟な集計，を挙げ，これらの要件を満たす情報集約言語を提案する．

6.1　情報集約言語がもつべき要件

ある製品の評判情報について，好評/不評の割合と，具体的に記述された各評判（評価属性，評価表現）を好評/不評ごとに集計する集約処理を考える．情報要素リレーションは，第 3 章の図 3.1 を想定する．この集約処理を SQL で行うために，例えば，Group by 句を用いて次のように記述したとする．

```
SELECT 評価極性,評価属性,評価表現,count(*) FROM 情報要素リレーション
WHERE 対象物="製品 A"
GROUP BY 評価極性,評価属性,評価表現
```

この問合せでは，製品 A に関する評判を，評価極性，評価属性，表現表現が全て一致するタプルごとに集計し，その個数を出力するため，製品 A の各評判が実際にどのような用語で記述されていたのかを取得できる．しかしながら，好評と不評の割合を求めるためには，集計結果のリレーションに対して評価極性が一致するものの個数を自前で再集計するか，グループ化条件を ‘評価極性’ だけにして，再度の問い合わせを行う必要がある．ただし，再度の問合せを行う場合は，評判（評価極性，評価属性，評価表現）の集計用と，評価極性の集計用の 2 回の問合せに分かれてしまうため，各評判が好評か不評のどちらに結びつくのかのデータ構造をもつことができない．このように，通常の SQL では，何らかの内訳をもつような階層構造をもつ集約結果の生成を行うことはできない．

一方，評価属性や評価表現など，情報要素属性の多くは文書中から自動的に抽出されたものであり，自然言語で記述されている．そのため，単純に集計を行うと表記ゆれなどがあるために，正しい集計を行うことができない．例えば，‘すばらしい’ と ‘素晴らしい’ は，別の用語に分かれて集計される．

以上のように，情報要素リレーションを対象とした問合せ言語は，次の要件を満たす必要があり，現状の SQL の記述能力では不十分である．

要件 (a): 階層的な内訳をもつ集約結果を生成することができる．

```
<query>              ::= "SC=\"" <search cond> "\""
                         "MC=\"" <summary cond list> "\""
<search cond>        ::= <search unit>
                       | <search unit> " and " <search cond>
<search unit>        ::= "(" <attribute> "=" <value set> ")"
<attribute>          ::= <text>
<value set>          ::= <value> | <value> " or " <value set>
<value>              ::= "'" <text> "'"
<summary cond list>  ::= <summary cond>
                       | <summary cond> "/" <summary cond list>
<summary cond>       ::= <grouping func>
                       | <grouping func> "," <summary cond>
<grouping func>      ::= <func name> "(" <parameters> ")"
<func name>          ::= <ascii>
<parameters>         ::= <parameter> | <parameter> "," <parameters>
<parameter>          ::= <text>
```

図 6.1 情報集約言語の定義

要件 (b): 表記ゆれなどに対応した柔軟な集計ができる．

6.2　情報集約言語の提案

本研究では，6.1 節で述べた要件 (a) と，要件 (b) を満たすものとして，情報集約言語を提案する．図 6.1 に情報集約言語の BNF（Backus-Naur Form）による定義を示す．ここで，<text>と<ascii>は，それぞれ任意の文字列と，任意の ASCII 文字列を表す．問合せは，

SC="<検索条件>" MC="<集計条件の列>"

のように，情報要素リレーションから対象となるタプル集合を検索するための検索条件（<search cond>）と，検索条件を満たしたタプル集合を集計するための集計条件（<summary cond>) の列とから構成される．次に，各条件の指定方法の詳細について述べる．

6.2.1　検索条件

検索条件は，リレーションに対する単純なタプルの絞込みなので，SQL の WHERE 句 [37] に相当する．ここで，<value set>を定義し，複数の表記の論理和を少ない記述で指定できるようにしている．

SC="(対象物='製品 A' or '製品 A1' or '製品 A2') and (評価極性='好評')"

上記の式は，“対象物が，‘製品 A’（又はその表記ゆれの ‘製品 A1’ や ‘製品 A2’）であり，評価極性が ‘好評’ である情報要素タプル” を検索する．このようにして，検索時の表記ゆれに対応している．集計時の表記ゆれについては，6.2.2 節で述べる．

6.2.2　集計条件

要件 (b) を満たすために，BNF 中のグループ化関数（<grouping func>）を導入している．グループ化関数は，情報要素タプル集合が入力された際に，次を実行する．

(1) 各情報要素タプルに対して，何らかの基準でグループキーを動的に生成する．

(2) グループキーが同じタプルを同じグループとして 1 つにまとめる．

(3) 必要に応じてグループキーの順序でグループをソートする．

ここで，(1) の処理はグループ化の方法によって異なるため，次のグループキー生成関数を外部関数によって定義できるようにする．

- **makeGroupKey**

  入力： 情報要素タプル集合，集計条件中のパラメータ配列（<parameters>）

  出力： <グループキー，情報要素タプル>の集合

このような関数の例として，クラスタリングがある．クラスタリング関数では次の処理を実行する．

(1) 入力として与えられた情報要素タプル間の類似度を計算する．

(2) 類似度が高いものを 1 つのクラスタとする．

(3) 各クラスタの ID をグループキーとする．

(4) グループキーを，そのクラスタに属する各情報要素タプルに付与して返却する．

この結果，類似度の高いタプル集合が同じグループとして後段の処理で利用できるようになる．各 makeGroupKey に対応したグループ化関数名を定義し，この関数名を集計条件で指定すると，対応付けられた makeGroupKey 及びグループ化関数の (2)，(3) の処理を実行する．このように，外部関数を追加するだけで，任意のグループ化方法の組込みができる．外部関数呼出しの実現方法は，6.3 節で詳しく述べる．

また，要件 (a) を満たすために，集計条件に，グループ化関数の列を記述できるようにしている．グループ化関数の列が指定されると，あるグループ化関数の出力となる情報要素タプル集合の配列の各要素（情報要素タプル集合）に対して，再度，次の階層のグループ化関数を実行する．更に，これらを再帰的に実行することによって，最終的には，階層的なグループの構造を表す集計木を生成する．次に，集計木の各ノードに対して配下の総タプル数と直下の子ノード数を付与し，これを集計値としている．配下の総タプル数は，そのグループに対応するグループキーをもつタプル数，直下の子ノード数は，1 つ下の階層のグループキーの異なり数を表す．

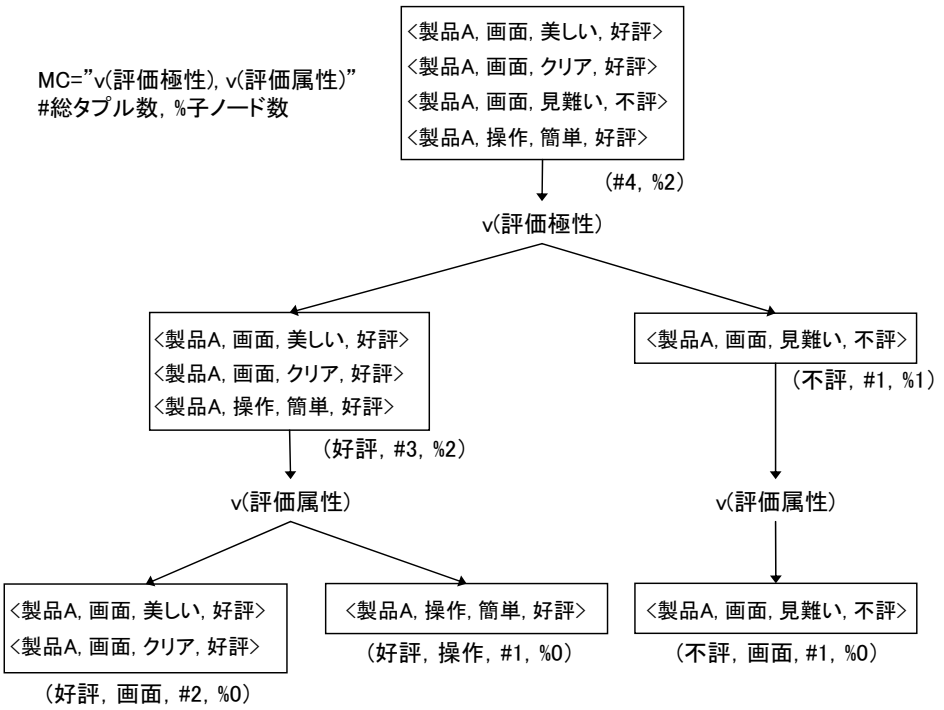

図 6.2 階層的な集計処理

図 6.2 に，入力された 4 個のタプルを次の集計条件にしたがってグループ化する例を示す.

```
MC="v(評価極性),v(評価属性)"
```

ここで，‘v(属性名)’ は，引数の属性名の属性値をそのままグループキーとして，グループ化するグループ化関数である．まず，集計条件の 1 つ目のグループ化関数 ‘v(評価極性)’ を呼び出し，‘評価極性’ が ‘好評’ と ‘不評’ の 2 つのグループに分ける．次に，各グループに対して，2 つ目のグループ化関数 ‘v(評価属性)’ を呼び出し，結果として，各グループについて ‘評価属性’ が ‘画面’ や ‘操作’ であるグループをそれぞれ生成する．このように，再帰的にグループ化関数を呼び出すことによって，集計の観点を変えた任意の組合せのクロス集計を実現する．また，問合せ式中に集計条件の列を記述できるため，検索条件が同じで複数の集計結果が必要な場合にも，一度の問合せで済む.

6.3　グループ化関数呼出しの実現方法

グループ化関数の呼出し機構を実現するために，オブジェクト指向の抽象クラスと，そのクラスの抽象メソッドとして，次を定義する.

抽象クラス名: BaseTupleGrouping

抽象グループキー生成関数: makeGroupKey(情報要素タプル集合，パラメータ配列)
返却値: <グループキー，情報要素タプル>の集合

新しいグループ化方法を追加したい場合は，BaseTupleGrouping クラスを継承したクラスを定義し，makeGroupKey を実装する．このクラス名とグループ化関数名との対応表をシステムに追加することで，集計条件中でグループ化関数が使用できるようになる．

実際に 2 つのグループ化関数を定義する例を示す．1 つ目は，6.2.2 節で述べたクラスタリングである．クラスタリングを実行するために次のクラスを実装する．

クラス名: ClusteringTupleGrouping

グループキー生成関数: makeGroupKey(情報要素タプル集合，[属性名集合])

処理概要:

(1) 入力された各属性名の属性値から概念ベクトル [7] を生成し，これらを合成することで各情報要素タプルに対応する特徴ベクトルを生成する．

(2) 特徴ベクトル間の類似度を計算する．

(3) 類似度に基づきクラスタリングを行う．クラスタリングには，最長距離法に基づく階層型クラスタリングを用い，各クラスタに属する全タプルとそのクラスタ重心との距離の合計が小さい順に一定個数のクラスタを選択する*．この手法によって，非常に似ているタプル集合からなるクラスタを生成できる．

(4) 生成された各クラスタに属するタプルに同一のグループキーを付与し，返却する．

2 つ目のグループ化関数は，時系列のグラフでの利用を想定し，ある与えられた期間ごとに情報要素タプルを集計する関数である．この関数を実現するために次のクラスを実装する．

クラス名: DateTermTupleGrouping

グループキー生成関数: makeGroupKey(情報要素タプル集合，[日付カラム名，集計期間])

処理概要:

(1) 各情報要素タプルから日付カラム名に対応するカラム値（日付）を取得する．

(2) 日付を集計期間で，除算し，整数化した商をグループキーとする．

(3) グループキーを各情報要素タプルに付与し，返却する．

この関数を用いることで，あらかじめ集計期間を表す値をそれぞれ格納しておかなくても，絶対日時を格納しておけば，集計条件で指定した集計期間ごとに集計できるようになる．

次に，これらのクラスについて，集計条件中に指定するグループ化関数名と，同一階層内でのグループの出力順を決めるためのソート方法を次のように指定する．

- **クラスタリング用グループ化関数**

*上位のクラスタが選択された場合には，下位のクラスタは選択しない．

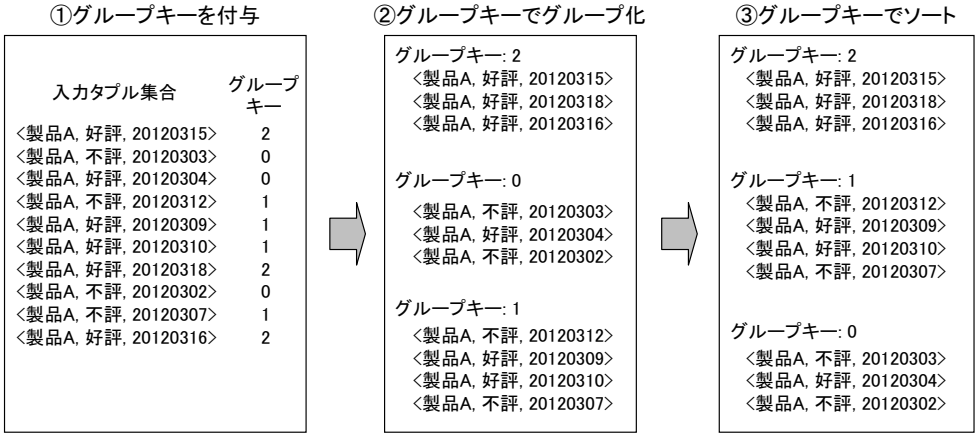

図 6.3 日付が集計されるフロー

グループ化関数名: cl

対応クラス名: ClusteringTupleGrouping

ソート方法: 文字列，降順

- **期間集計用グループ化関数**

  グループ化関数名: dt

  対応クラス名: DateTermTupleGrouping

  ソート方法: 数値，降順

これで，関数を追加する作業は完了となる．

実際の関数呼出し方法を，dt を用いた例で説明する．まず，集計条件が，次のように記述されていたとする．

```
MC="dt(DATE,7),v(評価極性)"
```

ここで，DATE には，基準日からの経過日数が含まれていることを想定している．図 6.3 に，‘dt(DATE,7)’ の処理フローを示す．

(1) DateTermTupleGrouping クラスの makeGroupKey(情報要素タプル集合，[”DATE”, ”7”]) が呼び出され，7 日ごとに同じグループキーを付与する．

(2) グループキーが等しい情報要素タプルをグループ化する．

(3) グループキー（DATE を 7 で割った商) を数値化し，各グループをグループキーの降順にソートする．

この処理によって，7 日ごとにグループ化され，日付の新しい期間順にグループがソートされたデータ構造をもつ情報要素タプル集合が生成される．更に，集計条件の v(評価極性) を適用し，各グループを評価極性（好評/不評) でグループ化し，配下のタプル数の集

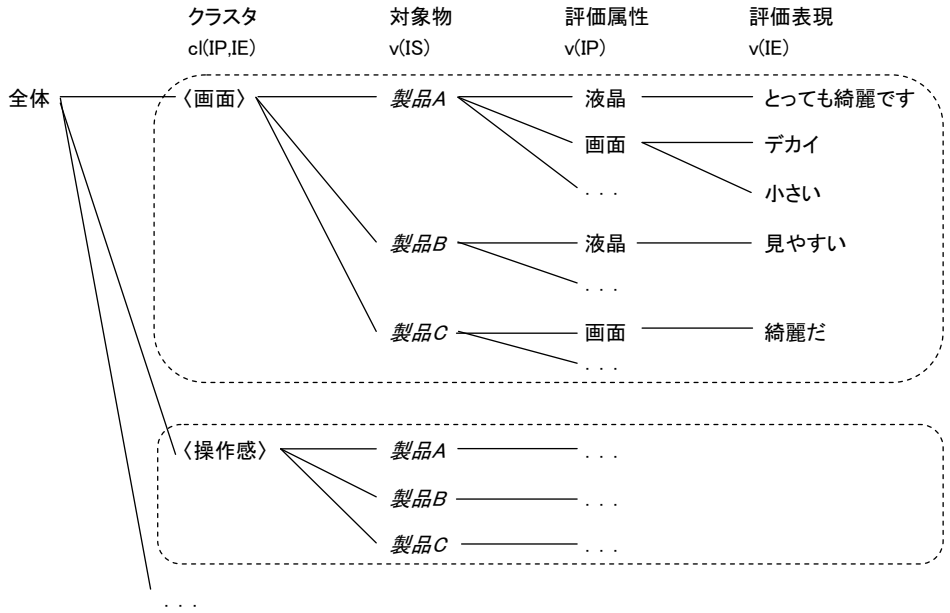

図 6.4 情報集約結果のもつデータ構造

計を行う．この結果，時系列の好評と不評の評判の個数の変化をもつデータ構造が生成される．

このように，BaseTupleGrouping を継承したクラスで，makeGroupKey を実装し，これにグループ化関数名とソート方法を対応付ける指定をするだけで，任意の集計方法を組み込むことができる．

6.4　実際の問合せ例

情報集約言語を用いた実際の問合せ例として，評判情報の集約タスクにおいて，ユーザが指定した複数の製品名を，似た観点の評判で比較して表示するという集約方法を用いて説明する．

情報要素リレーションは，第 3 章の図 3.1 を想定する．ここで，問合せを次のように記述したとする．

```
SC="(IS=’製品 A’ or ’製品 B’ or ’製品 C’)"
MC="cl(IP,IE),v(IS),v(IP),v(IE)"
```

この問合せに対する集約結果は，図 6.4 の階層的なデータ構造となる．また，この集約結果を生成するフローを次に示す．

(1) 検索条件（SC）に従い，対象物（IS）として製品 A，製品 B，製品 C のいずれかを含む情報要素タプルを検索する．

(2) 検索条件を満たした情報要素タプル集合に対して，集計条件の先頭のグループ化関数 cl(IP,IE) を呼び出す．ここで，cl は，6.3 節で述べた ClusteringTupleGrouping に対応

	よさ	薄さ	画面	消耗	操作感
■ *製品A* [redacted]	いいですね いいかなー 良かった 良かった	ギリギリのサイズが限界かな 辺りのサイズが良い 大きさが気になった	画面がデカイ 液晶がとっても綺麗です 画面が小さい	バッテリーの消耗が激しい 電池パックが貰えない 電池が大容量だ	ブラウザの操作感が分かる 私の操作が悪かった
■ *製品B* [redacted]	うまい 特徴が欲しいよねー いいね	薄くなっていますね	液晶のサイズも見やすい 画面が暗くなった	バッテリー消耗は激しい バッテリーもちが悪い 電池持ちがわるい	ソフトウェアキーボードのミスタッチも少ない マルチタッチも完璧になる ブラウザの操作感が分かる
■ *製品C* [redacted]	よいですか いいですね いいですよねー	薄さも良い 薄くていいですねぇ 薄い	画面が綺麗だ	電池以外は合格点	ブラウザの操作感が分かる

図 6.5 クラスタリングを用いた評判比較表示

し，IP(評価属性) と IE(評価表現) をパラメータとして，このクラスの makeGroupKey を呼び出す．この結果，評価属性と評価表現の類似度が高いタプルがグループ化される．

(3) グループ化関数 v(IS) を呼び出し，(2) の処理でクラスタリングされた各グループ内の評判情報を，対象物（製品 A, 製品 B, 製品 C）ごとにグループ化する．

(4) グループ化関数 v(IP) と v(IE) を順に呼び出し，(3) の処理で対象物ごとにグループ化された情報要素タプルを，更に，評価属性，評価表現でグループ化する．この結果，各リーフノードに，<評価属性, 評価表現>が同じタプル集合をもつグループが生成される．

このように，対象物に関連する情報要素タプルを評判情報（評価属性と評価表現）の類似度でクラスタリングし，各クラスタを複数の入力された対象物ごとにグループ化している．この集約結果から似た観点の評判情報が縦に並ぶようにして比較を行う画面（図 6.5）を生成できる†．ここで，この画面は，第 7 章で詳しく述べる評判分析サービスの実際の出力である．

このように，表記ゆれに対応しながら，階層的な内訳をもつ集計結果を生成することで，従来では難しかった可視化に適した集約結果を簡易な指定によって取得できる．

†カラムのラベル（薄さ，画面等）は各クラスタで頻度が最大となる評価属性又は評価表現の名詞形である．

6.5　SQL の拡張仕様などとの比較

6.1 節で述べたように，純粋な SQL では，クラスタリングを呼び出すなどの柔軟な集約処理を記述できない．ただ，SQL Server の Transact-SQL，PostgresSQL やオラクルの関数呼出し機能では，テーブル（タプル集合）を入力として，テーブルを返す関数が用意されている．この関数を用いると各グループ化関数と同等の処理を記述できるため，要件 (b) の柔軟な集計を実現できる．しかしながら，SQL では，SELECT 句の出力はリレーションであるため，グループ化を行った後で，各グループ（出力のリレーションを分割した一部）に対して，再度，グループ化関数を適用するといった，再帰的なグループ化処理を行うことはできない．提案手法では，グループ化後の各グループに対して，再帰的にグループ化関数を呼び出し，その結果を木構造で取得する．このため，複数階層のクロス集計を簡易に記述できる．

一方，オラクルなどの一部のデータベースでは，ROLLUP 機能が実装されていて，複数階層の Group 化をある程度実現している．しかしながら，この機能の出力結果はリレーションであるため，階層的なデータ構造を生成するためには，出力結果から木構造を再構成する必要がある．また，完全一致だけをサポートし，グループ化関数が必要となる柔軟な集計を記述することはできない．この他，主に表示用の HTML や XML の生成を目的とした問合せ言語として，SuperSQL[2] が提案されている．SuperSQL は，任意のリレーションから XML ビューを生成できるため，要件 (a) の階層的な内訳をもつ集約結果の生成に利用できる．しかしながら，グループ化関数の再帰的な呼出しといった機能は持たないため，表記ゆれなどに対応した柔軟な集計を行うことはできない．

このように，提案を行った情報集約言語は，表記ゆれなどに対応した柔軟な集計を行いつつ，内訳を保存した階層的なデータ構造を生成できる点が他の問合せ言語とは異なる．

6.6　情報集約言語のまとめ

本章では，通常の SQL では対応できていない，情報集約言語がもつべき要件として，次の 2 つを挙げた．

要件 (a): 階層的な内訳をもつ集約結果を生成することができる．

要件 (b): 表記ゆれなどに対応した柔軟な集計ができる．

次に，これらの要件を満たす情報集約言語を定義し，その実現方法を述べた．提案言語は，グループ化のための関数を容易に定義し，集計条件の指定に応じて再帰的な呼出しができるため，情報集約タスクに必要な集約結果を簡易な記述によって取得できる．第 7 章では，実際にいくつかのグループ化関数とその再帰的な呼出しによって，有用な情報集約結果が得られることを示す．

第7章　情報集約データベースの実現と評価

本章では，第3章で提案を行い，第4章から第6章で個別の技術課題の検証を行った情報集約データベース（IADB）を実現し，その有効性を検証する．まず，IADBを用いた情報集約システムの構成を示し，実際の情報集約システムを構築する方法を明確にする．次に，その構築方法に沿って，評判情報を集約する評判分析システムを構築できることを示す．また，評判分析システムを用いたサービスを実現し，実サービスとして公開することで，実際のサービスで必要となる情報集約結果を，情報集約言語による問合せで生成できることを示す．更に，評判とは異なるタスクとして，将来情報の集約タスクにIADBが適用できることを示し，また，他の研究との比較を行うことで，IADBの位置づけを明確にする．最後に，IADBが第1章で述べた情報集約フレームワークがもつべき3つの要件を，どの程度満たしているのかについて述べる．

7.1　情報集約システムの実現

IADBを用いた情報集約システム全体の構成図を(図7.1）に示す．アプリケーションプログラムの開発者は，まず，情報要素リレーションのスキーマを定義し，このリレーションに対する検索と集計によって，所望の集約結果を得ることができるのかを確認する．次に，3種類の外部関数を実装する．

文書解析関数（analyzeDocument）事前処理の文書解析機能から呼び出され，入力文書を解析し，情報要素タプル集合を返却する．対象物の抽出を行う処理も，この中で行われるため，必要に応じて，第4章で述べたような固有表現辞書を利用する．

タプル評価関数（calcTupleScore）オンライン処理の検索機能から呼び出され，“入力されたキーワードと事前抽出された部分情報要素タプルが結合されて，完全情報要素タプルとなるかどうかの確信度” を表すタプルスコアを返却する．

グループキー生成関数（makeGroupKey）オンライン処理の集計機能から呼び出され，情報要素タプル集合の各タプルに対して，グループキーを付与する．このグループキーに基づきグループ化が行われる．

文書解析関数とタプル評価関数は，動的タプル生成で必要なものであり，第5章の5.3節で詳しく述べた．グループキー生成関数は，集計時に必要なものであり，第6章の6.2節で詳しく述べた．ここで，グループキー生成関数は，必要なグループ化方法の種類数分を実装する*．各情報集約タスクに依存する実装は，上記の3つの外部関数だけであり，これらの処理の実装によって，情報要素リレーションへの問合せができるようになる．

*各関数はすでに実装されているものを利用することもできる．

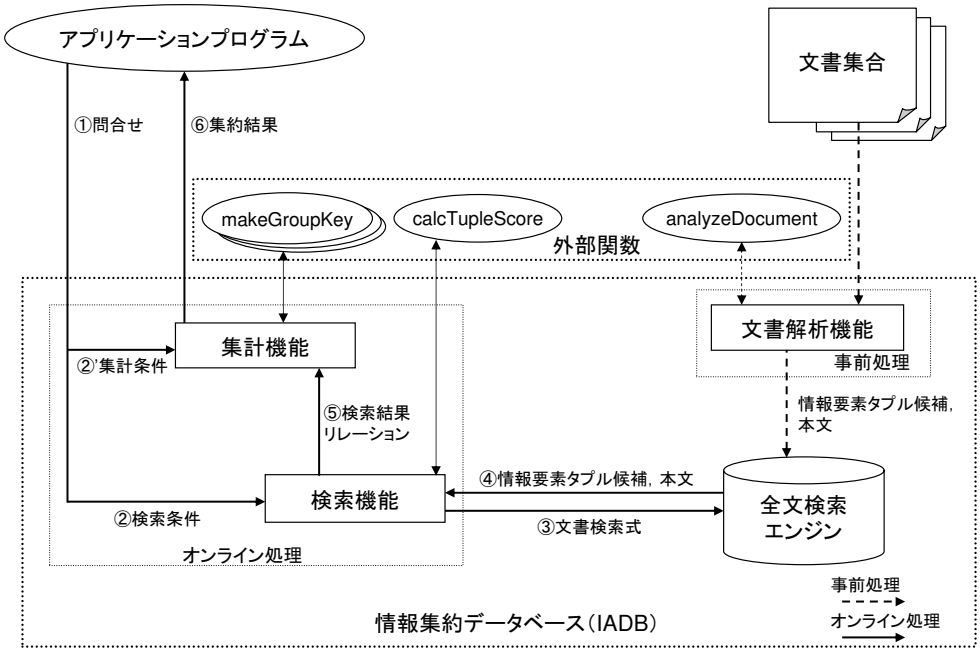

図 7.1 情報集約データベースを用いたシステム構成図

実際の処理手順は，次のとおりである．

(1) クローラなどが集めた文書集合を IADB に入力すると，文書解析機能は，analyzeDocument を呼び出すことで，情報要素タプル集合の候補を生成し，これらを全文検索エンジンンに本文とともに格納する．

(2) アプリケーションプログラムから，情報集約言語での問合せを受信すると，検索機能は，検索条件から文書検索式を生成し，全文検索エンジンにて検索を実行し，対象文書集合を絞込み，それらの中から情報要素タプルの候補を取得する．

(3) 各情報要素タプルの候補に対して，calcTupleScore を呼び出し，その確信度を付与し，確信度の高い候補に検索条件を適用し，検索結果リレーションとする．

(4) 集計機能は，検索結果リレーションに集計条件を適用する．この際，各集計の観点に対応するグループ化関数は，内部で，makeGroupKey を呼び出し，グループキーが一致するものを同じグループとする．

(5) 集計条件に応じて，グループ化関数を再帰的に呼び出し，最終的には，木構造の集約結果を生成する．

(6) 集約結果を XML に変換して返却する．

このように，ある情報集約タスクについて，集約の対象となる情報要素のスキーマを定義することができ，ある情報集約タスクに依存する全ての処理を，analyzeDocument，calcTupleScore，makeGroupKey の 3 種類の関数によって記述でき，かつ，必要となる情

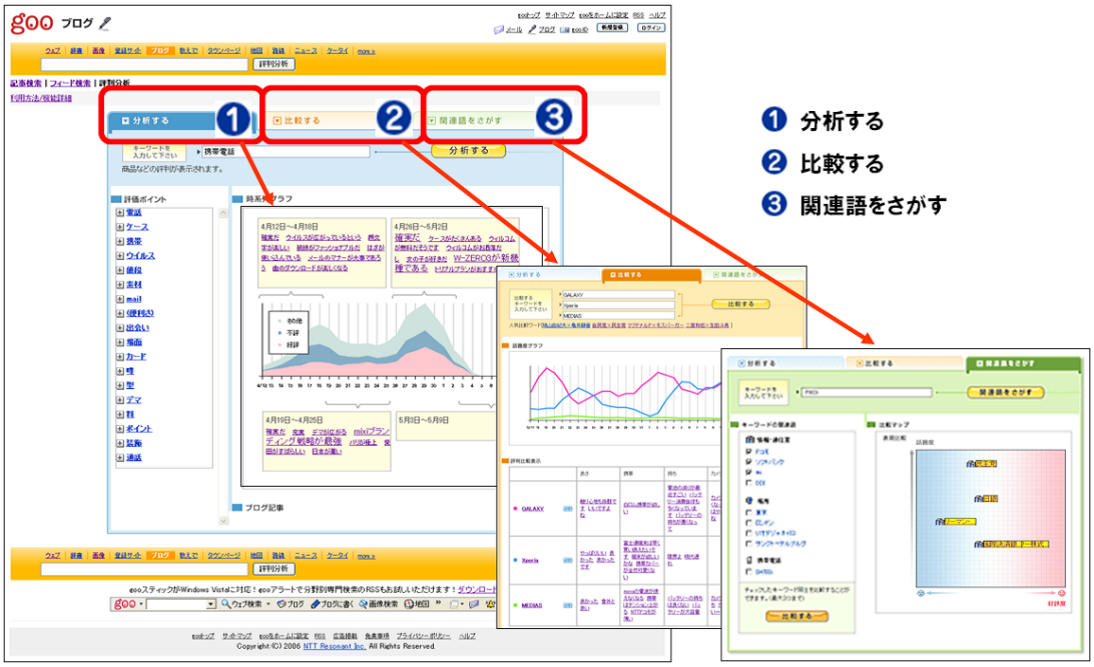

図 7.2 評判分析サービス

報集約要求を第 6 章の図 6.1 の問合せ式によって記述できる場合に，そのタスクは IADB 上に実現できる．

7.2 節では，実際に評判分析サービスに対して，IADB を適用する例を示す．

7.2　評判分析サービスへの適用

本節では，IADB を実際の評判情報を集約するタスクに適用しその有効性を検証する．評判分析サービスは，ポータルサイト goo 上の実サービスとして，2007 年 10 月から 2009 年 7 月までの間に公開を行った．評判分析サービスは，数千万件のブログ記事の中から評判情報を抽出しておき，ユーザがキーワードを入力すると，‘分析する’，‘比較する’，‘関連語をさがす’ という 3 つの画面で，その結果をオンラインで確認できるサービスである (図 7.2)．オンラインの処理として，ユーザからキーワードが入力されると，アプリケーションプログラムは問合せ式を生成し，情報集約データベースへの問合せを行う．IADB は，7.1 節で述べたように，XML 形式で評判の集約結果を返却する．アプリケーションプログラムは，この集約結果 XML を元に画面表示を行う．次に，(1) 評判分析システムの実現方法と，(2) 問合せと集約結果の可視化方法について述べる．

7.2.1　評判分析システムの実現方法

文献 [34] では，評判情報を，

<Nokia 6800, color screen, nice, the writer> (<対象，側面，評価，評価者>)

表 7.1 問合せ式

画面名称	検索条件	集計条件	集計名
分析する	(IS=’ 製品 A’)	(A) v(IP),v(IE)	評価属性分類表示
		(B) i(DD,7),v(IP),v(IE)	評判時系列表示
		(C) i(DD,3),v(IO)	好評/不評時系列表示
比較する	(IS=’ 製品 A’ or ’ 製品 B’ or ’ 製品 C’)	(D) i(DD,3),v(IS)	話題度時系列比較表示
		(E) cl(IP,IE),v(IS),v(IP),v(IE)	評判同一観点比較表示
関連語をさがす	(kw=’ サッカー’)	(F) v(ISC),v(IS)	関連語クラス分類表示
		(G) v(IS),v(IO)	関連語マップ表示

という 4 つ組で表現している．また，文献 [71] では，

<ラーメン屋 A，スープ，美味しい>（<対象，属性，評価>）

という 3 つ組で評判情報を表現している．これらの表現はいずれも対象物（対象）と，それに付随する属性（側面，評価，評価者など）の集合と見なすことができるため，情報要素タプルと等価な表現である．本研究では，これらの先行研究を参考に，第 3 章の図 3.1 に示した情報要素リレーションのスキーマを定義した．

次に，情報集約システムを構成するためにタスクごとに必要な次の 3 種類の関数を実装した．

analyzeDocument: 入力された文書から，評判情報を抽出する．第 5 章の 5.5.1 節で述べたものである．

calcTupleScore: 応答速度を重視し，第 5 章の 5.5.3 節の距離スコア法によって，タプルスコアを計算する．

makeGroupKey: 第 6 章で述べたグループ化関数 ‘v(属性名)’，’dt(日付カラム名，集計期間)’，’cl(属性名集合)’ に対応する処理を実行する．

このように，情報要素リレーションを定義し，3 つの種類の関数を定義することによって，評判分析システムを実現した．7.2.2 節では，このシステムに対する問合せによって，実際に有用な集計結果が得られることを述べる．

7.2.2　問合せと集約結果の可視化方法

表 7.1 に実際の問合せ式を示す．また，‘分析する’，‘比較する’，‘関連語をさがす’ の各集約結果画面をそれぞれ，図 7.3，図 7.4，図 7.5 に示す．

図 7.3 の ‘分析する’ の画面では，ユーザがキーワードを入力すると，IADB は，第 5 章の動的なリレーション生成を行い，表 7.1 の (A)，(B)，(C) の集計条件にしたがって，集約結果を生成する．アプリケーションプログラムは，取得した集約結果からグラフなどを生成し，図 7.3 を表示する．各集計条件に応じた動作の概要を次に示す．

図 7.3 ‘分析する’ の画面

(A) 同一の評価属性をもつ情報要素タプルをグループ化し，その配下に関連する評価表現を出力する．その結果をそのままツリー表示している．例えば，図 7.3 では，入力された製品 A の ‘サイズ’ に関連する評価表現（‘小さい’ や ‘いいかな’）を ‘サイズ’ 配下にまとめて表示している．

(B) 7 日の期間ごとに，評価属性，評価表現の 2 つ組（評判表現）の個数を集計する．この評判表現の個数の上位語を画面に表示している．この表示によって，入力キーワードに対する評判表現が各期間ごとに，どのように変化しているのかが分かる．

(C) 3 日の期間ごとに，情報要素タプルを集計し，各期間内は，評価極性（好評，不評，その他）で集計を行う．各評価極性の個数をタイムチャートに表示している．この表示によって，好評や不評の評判数の変化や注目度（評判の総数）の推移が分かる．

このように，1 つのキーワードに対する評価の側面（評価属性）で分類された評価表現や，評判表現や好評/不評などの推移を時系列に見ることができる．

図 7.4 の ‘比較する’ の画面では，ユーザが 2〜3 個のキーワードを入力すると，‘分析す

図 7.4 ‘比較する’ の画面

る’ と同様に動的なリレーション生成を行い，(D)，(E) の集計条件にしたがって，集約結果を生成する．アプリケーションプログラムは，取得した集約結果からグラフなどを生成し，図 7.4 を表示する．各集計条件に応じた動作の概要を次に示す．

(D) 3 日の期間ごとに，情報要素タプルを集計し，各期間内は，入力された各キーワードで集計を行う．この結果を元に，各対象物についての情報要素タプルの個数をタイムチャートに表示している．この表示によって，製品などの注目度の推移を比較することができる．

(E) 複数のキーワードに関する評判情報を似た観点が縦に並ぶようにクラスタリングして表示している．この表示方法については，第 6 章の 6.4 節で詳しく述べた．

このように，入力された複数のキーワードに関する注目度の時系列での比較や，似た観点での評判表現の比較ができる．

図 7.5 の ‘関連語をさがす’ の画面は，入力されたキーワードにヒットした文書中に存在する対象物を集計して表示している．この画面の出力では，ユーザから，対象物自体を表

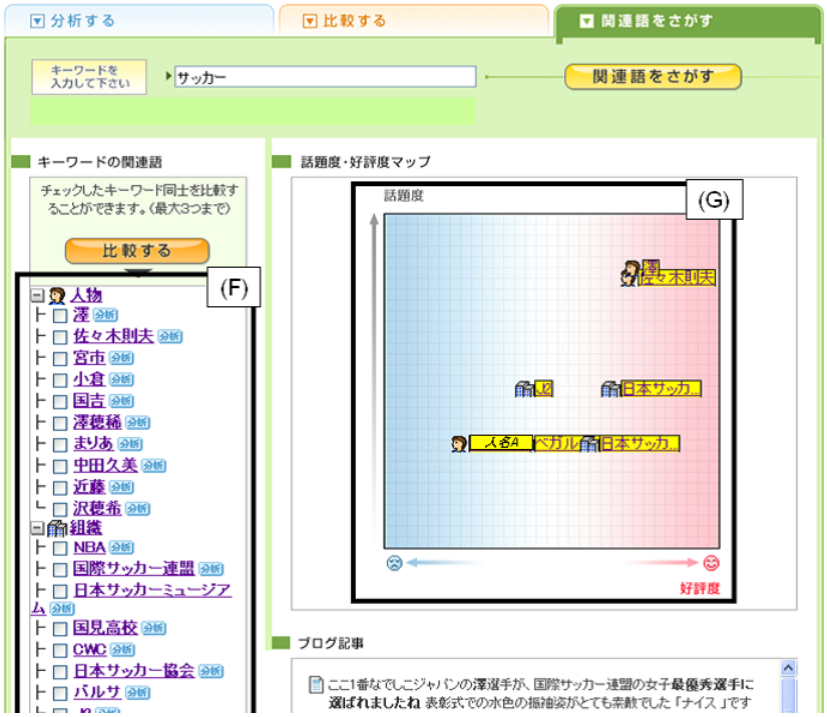

図 7.5 ‘関連語をさがす’ の画面

すキーワードが入力されないため，動的なリレーション生成は行われず，事前抽出された対象物だけを表示している．現在の実装では，人手で作成した辞書と汎用の固有表現抽出器を用いて対象物の抽出を行っているが，第 4 章で述べた手法で自動構築した辞書を利用することもできる．生成したリレーションに対して，(F)，(G) の集計条件にしたがって，集約結果を生成する．各集計条件に応じた動作の概要を次に示す．

(F) 事前抽出された対象物に付与されたクラス（人物名や組織名など）で対象物を分類し，各対象物を表示している．この表示方法によって，例えば，‘サッカー’ に関する人物を探すことができる．

(G) 情報要素タプルの対象物ごとの頻度を集計し，更に，各対象物について好評と不評の割合を集計する．頻度を Y 軸，好評と不評の割合を X 軸とした 2 次元マップにそれぞれの対象物を配置して表示している．この表示によって，例えば，‘サッカー’ に関して，話題となり，かつ，好評の評価がされている人物を探すことができる．

ここで，表 7.1 の ‘関連語をさがす’ の検索条件中の ‘kw=<キーワード>’ は，“キーワードを含む文書に含まれる事前抽出された完全情報要素タプルを取得する演算子” である．また，属性名の ‘ISC’ は対象物のクラス（人名，組織名など）を表し，対象物が事前抽出された場合だけ付与している．このように，入力されたキーワードに関する対象物を，そのクラス（人物や組織名など）や，その話題度や好評か否かに応じて発見することができる．

以上に述べた様々な集約結果画面は，全て，情報集約言語の問合せ結果である木構造のデータから実時間で生成している．このように，ある特定の構成の評判分析システムとい

う限定された範囲内ではあるが，本研究で提案している IADB を用いて，実際にオンラインの実サービスを実現できることを確認した．

7.3　情報集約データベースの課題

IADB を用いて評判分析サービスを構築，公開して得られた知見や課題を，次の観点で述べる．

(1) 情報集約言語の記述能力

(2) 情報集約結果の妥当性

(3) 情報集約サービスの適用範囲

7.3.1　情報集約言語の記述能力

7.2.2 節で示したように，各文書中の評判情報を情報要素タプルで表現し，情報集約言語による問合せによって，実サービスに利用できる有効な集約結果を数多く生成できた．この他，サービス企画を行っている担当から要望があった中で，本質的に情報集約言語の記述能力を超えるものは，特になかった．ただし，追加の外部関数定義や上位のアプリケーションプログラムでの実装が必要となる項目がいくつかあった．このような情報集約言語の課題を次に示す．

(1) **グループ化関数の設計**
各評判表現（評価属性と評価表現）を集計して出力するために，集計条件の最後が，‘v(IP)，v(IE)’ で終わるものが複数あった (表 7.1 の (A)，(B)，(E))．この集計条件は，同一の評価属性をもつタプルを 1 つのグループとし，各グループの配下に同一の評価表現のグループをもつ 2 階層の木構造を生成する．この場合，各葉ノードには，<評価属性，評価表現>が同じものがグループ化されるため，評判情報の語句を出力する上で必要なデータ構造は得られている．しかしながら，上位のアプリケーションプログラムの開発者から，<評価属性，評価表現> は，ペアで 1 つの単位として扱いたいため，評価属性を表す余分なノードをもつ木構造では，実装が複雑になるという指摘があった．この指摘に対しては，1 つの単位として扱いたい複数の属性値を文字列結合したものをグループキーにする makeGroupKey を追加すれば対応できたと思われる．このように，アプリケーションプログラムで利用したいデータ構造を想定し，木構造が必要以上に深くならないように，グループ化関数を設計することが重要である．

(2) **対象物に応じた観点での集計**
評判分析サービスとは別サービスである goo 映画[†]に，評判分析サービスと同一の IADB から評判情報を提供した．このとき，ストーリ，出演者，映像，音楽など評価の観点を固定して，この観点に沿った集約を行いたいという要望があった．この要

[†]http://movie.goo.ne.jp/

望に対応するために，各評価属性や評価表現がそれぞれ，どの観点に分類されるかの辞書を人手で用意した．次に，作成した辞書を用いて，評価属性や評価表現を各観点に変換し，これをグループキーとする makeGroupKey 関数を実装することで対応した．映画のように対象ドメインが固定されていれば，このような対応ができる．しかしながら，任意の商品名が入力されるサービスの場合に，商品カテゴリごとに評価の観点を固定したいという場合も起こりえる．例えば，テレビなら画質，音質，デザイン，掃除機なら吸引力，静音性，使い勝手などの観点がある．現状のアーキテクチャでは，入力されたキーワードから，商品カテゴリは分からないので，事前抽出できる固有表現を除くと，このような商品カテゴリごとに集計の観点を変更することはできない．このように，IADB では，入力キーワードのカテゴリを何らかの方法で推定する機能をもつことが望ましい．

(3) **分類されたグループからの集計値の生成**
情報集約言語では，階層的なグループ化に加えて，各ノードの配下の総タプル数と，直下の子ノード数を集計値として，各ノードに付与するようにしている．しかしながら，比率などを求めるためには，上位のアプリケーションプログラムで，集計木を走査し，関連するノードの総タプル数などを取得し，比率などの計算を行う必要がある．

一方，いくつかの可視化画面の生成では，各ノードにグループを表すラベルを振る必要があった．例えば，第 6 章の図 6.5 のクラスタリングの例では，各グループを表すラベルとして，頻度が最大となる評価属性を付与している．この機能は，上位のアプリケーションプログラムで個別に実装を行ったが，ラベル生成方法を呼び出せるなどの拡張が望ましい．

このように，現状の情報集約言語では，主に情報要素タプルをグループ化する機能だけをサポートしているが，各グループや場合よってはもう少し広い範囲の集計木を対象に，各種の統計量計算やラベル生成を行うための外部関数を呼び出す機能をもつことが望ましい．この機能によって，上位アプリケーションプログラムの実装範囲を更に削減できる．

(4) **文書などを単位とした集計**
情報要素タプルを単位とするのではなく，文書を単位として集計を実施したいという要望もあった．例えば，対象キーワードに言及した記事数の折れ線グラフを生成するなどである．現状では，文書をいったん情報要素タプルに分解した上で情報要素リレーションを生成し，更に文書 ID をグループキーとすることで所望の結果を得ることができるが，このような処理では時間がかかる．一方，文書をタプル単位に分解することなしに，集計を行うようにすれば高速化が期待できる．このように，問合せ処理の最適化機能をもつことが望ましい．

7.3.2　情報集約結果の妥当性

全体的な傾向として，出現頻度が高く，かつ，多義性がなく対象物を特定できるキーワードの情報集約結果は，比較的高い精度であった．一方で，典型的な問題として次のような

ものがあり，精度を低下させる要因となっていた．

- **出現頻度が低いキーワード**
  Web 上にほとんど記述されていない用語に関しては，当然，集約結果の生成は困難である．しかしながら，現状は，入力キーワードの完全一致だけをサポートしているが，検索時に表記ゆれを展開するなどの手法を用いることによって，そのままでは出現頻度が低いキーワードをカバーできる．ただし，これには次の課題がある．

  (1) 表記ゆれの自動展開を行うための辞書やアルゴリズム

  (2) 検索条件の展開だけでなく，集計条件も同時に展開を行うための処理方法

  現状，IADB への問合せを生成するのは上位のアプリケーションプログラムであるが，表記ゆれの自動展開を行う機能を個別に実装することは難しい．このため，IADB 内部で，何らかの自動展開機能をもつことが望ましい．

- **特定性が低いキーワード**
  短い識別番号などの場合，多義性が大きく，関係のない情報要素属性を抽出していた．また，一般語の場合，特定性の低い誤った情報を抽出する傾向にあった．例えば，“昨日の A カントリーでのゴルフは楽しかった” という特定のゴルフ場に紐づく評判情報から，<ゴルフ，楽しかった>という一般的なゴルフに関する評判情報を抽出してしまうなどである．これは，動的タプル生成では，固有表現であるかどうかにかかわらず，入力された任意のキーワードを対象物として情報要素タプルを生成してしまうことに原因がある．更に，‘VAIO’ などの固有表現ではあるが，同一ブランドのグループを表す表現が入力されることもあった．これに対して，文書には，ある ‘VAIO’ の機種に関する評判が記述されていたが，ユーザが望んだ機種と，文書中の機種が一致しないこともあった．このように，対象物の特定は難しい課題であり，今後の研究が必要である．

- **対象文書のランキング**
  IADB では，処理効率のために全数での集計は難しく，対象文書をランキングし，上位の文書だけから情報要素タプルを生成している．文書の検索時に，評判を含むものや，対象とする属性をもつもので絞込みができることを第 5 章の 5.4.2 節に示したが，これに加えて，有用な文書をサンプリングするための文書のランキング手法が重要であった．例えば，各文書のレビューらしさや，スパムページのランクを下げるなどの対応を行わないと良好な集約結果は得られなかった．このため，文書検索式に独自の追加を行い精度を調整するなどの機能が必要となった．このように，IADB では，対象文書を取得するための戦略（レビューを優先し，スパムを下げるなど）を柔軟に組み込めることが望ましい．

7.3.3　情報集約サービスの適用範囲

ブログ記事を対象とした評判情報の集約を行うオンラインサービスが実際にどのようにユーザに利用されていたのかを検証するために，第 5 章の 5.5.2 節の方法で収集した 100

表 7.2 評判分析サービスに投入されたキーワードのクラス

拡張固有表現階層		個数	例
人名		31	二宮和也，舞花，加藤ミリヤ，江崎，YUKI，obama，ラルク，氷川きよし，土佐尚子，海野フミ子，松井秀喜など
組織名		27	JCB，大丸，山形オートリサイクルセンター，大地を守る，よこしまブロッコリー，民主党，NEDO，ベストくすりなど
製品名		24	
	商品名	9	vaio，ナノックス，クロックマン，レガシィ，新型フィット，リポビタン D，グインサーガ，重力ピエロ，セブンティーン
	商品識別番号	3	DT615，DTV–H400S, D–11M
	上記以外	12	雄魂姓名録，wakwak，デジプリ，ジャンナビ，コラショ，マイエリア，エコポイント，はやぶさ，ガルギールなど
施設名		13	
	店舗・遊戯施設名	8	さやの湯，ソルレヴァンテ，岸権，スイーツきたがわ，ありそ鮨し，武道館，吉祥寺美術館，パゴン本店
	上記以外	5	電気通信大学，川女，京都橘大学，田園調布，首都高
その他の拡張固有表現		2	通風，長崎
拡張固有表現以外		3	ゴルフ，トイガン，Government

語を用いる．これらのキーワードを収集したサービスは，本章で述べた評判分析サービスとはユーザインタフェースが異なるが，内部で IADB を利用し，ユーザのキーワード入力に対して，ブログ記事内の評判情報の集約結果を可視化して表示している．

まず，これら 100 語に対して，第 4 章の 4.1 節の拡張固有表現階層のクラスを人手で付与した．最上位の階層の各クラスに含まれるキーワード数は表 7.2 のとおりである．ここで，分析のために次の独自のクラスを定義した．

商品名: 製品名の中で，特に，EC（Electronic Commerce）サイトなどで，販売されると想定されるもの

商品識別番号: 商品名の中で，その表記が英数字と記号で構成されるもの

店舗・遊戯施設名： 施設名の中で，特に，商品や食事などのサービスをユーザに提供するもの

表 7.2 のように，拡張固有表現に分類されないキーワードは，3 語だけで，ユーザがキーワードを入力する際には，ほとんどの場合で，特定性の高い固有表現を入力する傾向にあった．また，想定では，評判というサービスの性格上，商品名や店舗・遊戯施設名が多いと予想していたが，実際には，それぞれ，全体の 12%（商品識別番号を含む），8%と少なく，実際のユーザの入力は，人名，組織名が特に多かった．商品名や，店舗・遊戯施設名については，すでに多くの EC，グルメ，旅行サイトなどでレビューを公開している．一方，その他の評判情報の多くは，どこかの特定のサイトに存在しているわけではない．このため，任意のキーワードを入力とし，大規模なブログ記事から評判情報を抽出するサービスは，商品名や店舗・遊戯施設名以外のキーワードで多く利用されていたと考えている．

このように，いくつかの種類の情報については，特定のサイトで得ることが難しく，このような情報に対して IADB が特に有効である．

7.4　将来情報の集約タスクへの適用

評判情報の集約サービスとは別のタスクに対しても，7.1 節で述べた方法を用いて，IADB を適用することで，その汎用性を示す．文献 [29] では，“キーワードを入力すると，キーワードに関連する将来を予測した情報をイベントごとに分け，時系列のタイムライン上に可視化する手法” を提案している．ここで，このタスクを将来情報の集約タスクと呼ぶ．将来情報の集約タスクに，IADB を適用するために，まず，集計の対象となる情報の断片であるイベントを次の情報要素で表現する．

<対象物 (IS)，動作 (EV)，イベント発生時刻 (ED)，文書の記述された日時 (DD)>

次に 3 種類の外部関数を定義する．

analyzeDocument:　入力された文書から，日時表現を抽出する．また，各日時表現について，前後の文字列中の単語集合から特徴ベクトルを生成する．各日時表現をイベント発生時刻（ED）とし，特徴ベクトルを動作（EV）とし，文書の記述された日時 (DD) と，文書中の日時表現の出現位置からなる部分情報要素タプルを生成し返却する．

calcTupleScore:　入力キーワードとイベント発生時刻（ED）との本文中での相対的な出現位置，及び，文書の記述された日時（DD）からタプルスコアを計算し返却する．

makeGroupKey　動作 (EV) とイベント発生時刻 (ED) を用いて，情報要素タプル間の距離を計算し，この距離に応じてクラスタリングを行い，グループ化した結果を返却する．この関数を，$ecl(EV, ED)$ というグループ化関数に対応付ける．

IADB への問合せは，

```
SC="(IS='対象物')" MC="ecl(EV, ED), v(ED)"
```

と記述する．この結果，ある入力された対象物に関連するイベントが，そのイベント発生時刻の周辺単語を特徴ベクトルとした類似度でクラスタリングされ，各クラスタの中が，イベントの起こった日時で分類された集約結果が得られる．このように階層的にグループ化されたタプル集合が得られるので，各タプル集合に対して，イベントを表すラベルを生成すれば，文献 [29] の可視化画面を生成できる．

このように，将来情報を集約するといった，評判とは全く異なる情報集約タスクに対しても IADB は適用できる．更に，時間が掛かると考えられる日時表現の抽出や前後の文脈からの特徴ベクトルの生成はオフラインで実行されるので，高速化が期待できる．

表 7.3 関連研究との比較

	提案手法	大島らの手法	eHyouban
データモデル	単一リレーション	複数リレーション	なし
集約方法	独自クエリ	SQL	追加モジュール
情報の表現	情報要素タプル	任意タプル	評判タプル
集計結果	木構造	リレーション	個別の可視化
拡張性	外部関数	外部関数	なし
インデックス方法	通常の全文検索 + 情報要素	通常の全文検索	独自タプル
集約速度	早い	遅い	早い

7.5　関連研究との比較

提案手法を，文献 [48] の大島らの手法，及び，文献 [40, 71] の eHyouban の手法と比較する（表 7.3）．まず，提案手法では，大規模な文書集合中の全ての情報の断片を，単一の情報要素リレーションとして扱う．この結果，様々な情報要素の集約処理を，検索条件と集計条件からなる問合せで記述できる．eHyouban では，このようなデータモデルを持たないため，集約の方法が増えるたびにモジュールを追加する必要がある．一方，大島らの手法では，文書解析処理を行った結果を，リレーションの 1 つとして扱うことができるため，情報要素リレーションの集約に加えて，既存の RDB に蓄積された任意の情報を統合利用することができる．提案手法では，このような統合利用を行うことはできないが，多くの情報集約タスクは情報要素タプルを単位とする集約処理に置き換えることができる‡ため，この機能に特化することによって，問合せの簡易化と高速化を行っている．また，大島らの手法では，問合せ言語に SQL を用いているが，本研究の情報集約言語では，SQL では実現が困難な (a) 階層的な内訳をもつ集約結果の生成，(b) 表記ゆれなどに対応した柔軟な集計，が実現できる．

他のタスクへの拡張性を考えた場合，提案手法では，analyzeDocument，calcTupleScore，makeGroupKey を定義するだけで良いので，開発コストは低く抑えられる．一方，特定のアプリケーションプログラムである eHyouban では，このような関数の呼出し機構を持たないため全体の再実装が必要である．

大島らの手法では，各タスクで独自のインデックスをもつ必要はないが，外部に通常の全文検索用インデックスが必要である．提案手法では，通常の全文検索用インデックスに加えて，情報要素を格納するカラムが必要であるが，そのサイズは，通常，全文検索用インデックスのサイズと比べて小さい（7.2 節の評判分析システムでは約 17%程度）．また，通常の検索サービスや他の情報集約サービス間で全文検索用インデックスを共有することができるので，新しい情報集約タスクに適用するための格納コストの増分はわずかである．更に，情報要素属性の属性名を各属性値のプレフィックスとして格納しているため，任意の属性（例：評価極性が好評）をもつものに文書の絞込み検索ができる．一方，eHyouban

‡ここで，文書や文書集合を単位とした集計（例えば，対象物を含む‘文書数’の変化) は，情報要素タプルを，文書 ID でグループ化して集計すれば良いので，情報要素を単位とする集計に置き換えることができる．

では，評価属性と評価表現の 2 つ組を単位として，その前後の文字列をインデックスしている．これにより，保存された情報の取得は，提案手法よりも高速であると考えられるが，“前後の文字列サイズ × 文書中の 2 つ組の個数” のインデックスサイズが必要となり，前後の文字列のサイズが大きいと，通常の全文検索用インデックス相当の独自インデックスが必要となる．また，任意の属性による絞込みもできない．

以上述べたように，多くの情報集約タスクは，情報要素タプルの集約処理に置き換えることができ，本手法は，このような置き換えができるタスクに対して，様々な検索と集計をユーザの簡易な問合せに応じて実時間で実行することができる．また，新しいタスクに適用する場合の格納コストと開発コストは，既存手法と比べて小さい．

7.6　情報集約フレームワークの要件検証

第 1 章で情報集約フレームワークがもつべき要件として，次の 3 つを挙げた．

要件 (1) Web 上に公開されている自然言語で記述された大量の文書情報を対象とできること

要件 (2) 文書に含まれる情報の断片を対象として，問合せに応じて複数の操作を組み合せて実行し，情報集約結果を即座に生成できること

要件 (3) 様々な情報集約タスクに対して，少ない開発コストで適用できること

本節では，これらの要件が IADB において，どのように満たされたのかを検証する．

- **要件 (1) について**
  評判分析システムは，Web 上に公開されている任意のブログ記事をクロールし，文書解析処理を行うことで，自動で情報要素リレーションを生成している．また，通常の全文検索エンジンと全文検索用のインデックスを共有できるインデックス構造であるため，既存の検索システムに対して少ない設備コストで，このような情報集約システムの導入ができる．この結果，数千万件のブログ記事といった大量の文書を処理する評判分析システムを構築できた．ここで，対象文書の件数は，主に設備面の制約によって決まっていたので，設備コストを考えなければ，更に大規模な文書を対象とできる．

- **要件 (2) について**
  7.3 節で述べたように，上位のアプリケーションプログラムでいくつかの機能の追加の実装が必要となるなど，まだ，課題はあるものの，実サービスで実際に必要とされた評判情報の集約結果を情報集約言語による問合せで取得できることを確認した．また，動的なリレーション生成によって，集約結果の取得を実時間で行うことができるため，オンラインの情報集約サービスを実現できた．

  しかしながら，精度面にはいくつかの課題がある．特に，出現頻度が小さいキーワードなどでは，情報集約結果の精度が低い傾向にあり，このような低頻度語の精度を向上させるためには，個々の情報要素タプル自体の抽出の精度を向上させる必要が

ある．そのためには，まず，文書中の対象物の抽出及びクラスの判定精度の向上が必要である．対象物を抽出し，そのクラスが特定できれば，高精度の情報要素タプルの抽出ができるようになる．例えば，対象物が‘食べ物’であると分かると，‘おいしい’は，結びつきやすいが，‘使いやすい’は，結びつきにくいといった知識が利用できるようになる．このように，対象物のクラスを考慮することによって，対象物と属性との関係付けを行う関係抽出技術の高精度化が期待できる．また，7.3 節に述べたように，カテゴリに応じた観点での集計ができるようになる．

一方，動的なリレーション生成の精度を向上させるためには，様々な特徴量をオンライン処理で利用できる必要があるため，このような特徴量の新たなインデックス方法を考案することは，高速・高精度のリレーション生成には不可欠である．

これらの手法によって，情報集約結果の精度を向上させることができると考えている．

- **要件 (3) について**
  IADB では，analyzeDocument，calcTupleScore，makeGroupKey の 3 種類の外部関数を実装するだけで，評判情報の集約タスクを実現し，また，将来情報の集約タスクにも同様の方法で適用できることを示した．しかしながら，現状の情報集約言語では，主に情報要素タプルのグループ化だけをサポートしているため，上位のアプリケーションプログラムにおいて，各ノードの配下タプル数などから，統計値の再計算や，グループを表すラベルを生成する処理を実装する必要があった．これらの集計機能を IADB が提供することによって，上位のアプリケーションプログラムの実装が更に容易になると考えている．

  また，analyzeDocument での処理は，基本言語解析などの共通化できる部分が多い．そこで，これらの共通機能をライブラリ化し，用途に応じて組み合わせて利用できる仕組みを構築することが，開発コスト削減につながる．

上記に述べたように IADB は，情報集約フレームワークに必要とされる 3 つの要件を満たすものとして実現されている．今後，各機能の強化を行い，更なる精度向上と他のタスクへ適用する際の開発コストの削減を行っていきたい．

第8章　結論

8.1　まとめ

本研究では，ユーザの処理しなければならない情報量が急速に増大していく中で，複数の文書に含まれる情報を収集し，まとめて提示する情報集約タスクについての検討を行った．このような情報集約タスクを実行できるようにするためには，まず，情報を適切な形式で表現し，この表現に対する複数のテキスト操作を組み合わせて実行できるフレームワークが必要であることを述べた．特に，集約の対象となる文書中の情報の断片は，多くの場合，対象物とそれに紐づく属性の集合（情報要素）で表現できることに着目した．ここで，対象物を固有表現の名前に限定することで，対象物の特定性が高まり，情報要素を文脈から分離したとしても，元の文書の該当する箇所の内容に近いものとなることを述べた．そして，大量の文書データから情報要素を自動抽出し，これをタプルとした仮想的なリレーションを生成し，簡易な問合せによって，検索と集計ができる情報集約データベース（IADB）の基本アーキテクチャを提案した．

IADB を実現するためには，(1) 文書中の情報の断片からの情報要素タプルの自動生成，(2) 集約処理を実行するための問合せ言語，の 2 つの技術課題があることを述べた．(1) に関しては，(A) 固有表現辞書の自動構築手法と (B) 動的なリレーション生成手法，(2) に関しては，(C) 情報集約言語，を提案し，次のように解決策を示した．

(A) 固有表現辞書の自動構築手法

文書から情報要素タプルを生成するためには，対象物を網羅的に抽出することが課題であり，このためには，固有表現辞書が必要であることを述べた．次に，辞書に登録する用語や各用語がもつ語義を網羅的に収集することが重要な課題であることを述べた．この課題を解決するために，推定対象の表記が出現する個々の文脈ごとに推定を行い，推定結果であるスコアの上位の集合から表記の所属スコアを計算する表記出現特徴量法を提案した．表記出現特徴量法では，推定対象とするクラス以外の語義で，その表記が用いられている文脈の影響を軽減させることができる．拡張固有表現階層を対象とした評価を行い，表記出現特徴量法は，使用頻度の少ない語義に対するクラス判定の精度を向上させることができることを示した．更に，この結果から，タグなしコーパスを増やすことで多義語がもつ複数の語義を網羅的に獲得できるようになることを述べた．

(B) 動的なリレーション生成手法

多くの情報集約タスクでは，ユーザの入力したある対象物に関連する情報を集約したい場合が多いため，対象物を事前抽出するのではなく，入力情報を利用して情報要素リレーションを生成することが有効であることを述べた．次に，対象物に依存

する処理と，依存しない処理が多くの場合に分離できることに着目した．そして，対象物に依存しない処理を事前に行い，ユーザのキーワード入力時には，事前抽出してインデックスした情報と入力キーワードとを結びつける処理だけを実行することで，高速にリレーションを生成する手法を提案した．評判情報のリレーションを生成する処理において，未知語に対応しながら，約 800 件の検索された文書からの情報要素リレーションの生成処理を約 200 秒から 1 秒程度に短縮できることを示した．

(C) 情報集約言語

情報集約タスクでは，情報要素リレーションに対して，表記ゆれに対応しながら，複数の観点のクロス集計を行う必要があることを述べ，この課題を解決できる情報集約言語を提案した．提案言語では，任意の集計方法を外部関数として定義し，これをグループ化関数という形で集計条件から呼び出すことができる．また，各グループ化関数の実行結果である各タプル集合に対して，再度グループ化関数を適用できる．この結果，クラスタリングなどのグループ化方法を柔軟に組み込むことができるため表記ゆれにも対応でき，また，複数の観点のクロス集計が簡易な記述によって実現できることを示した．

IADB を用いた評判情報の集約を行うシステムをポータルサイト goo 上の実サービスとして提供することで，本アーキテクチャの有効性を検証した．この検証結果から，数千万件のブログ記事に対して，情報集約言語による簡易な記述によって，実時間で有効な情報集約結果が得られることを示した．次に，実際のユーザからの入力キーワードを分析することによって，このような評判情報を集約するオンラインサービスでは，商品名や店舗・遊戯施設名よりも，人名や組織名のニーズが高いことを明らかにした．この結果から，IADB は，特定サイトだけからでは得られないような情報に対して，特に有効であると考えている．更に，本アーキテクチャの汎用性に関して検証を行い，評判とは全く異なる将来情報の集約タスクに対しても IADB を適用できることを示した．最後に，関連研究との比較から，従来手法と比べ，本アーキテクチャは，汎用性と，高速性の両方を兼ね備えている点で優位性があることを述べた．

このように，IADB は，第 1 章の 1.4 節で挙げた 3 つの要件を満たす，情報集約タスクのためのフレームワークとして実現できたといえる．

8.2　今後の展望

Web の急速な普及や利用形態の変化が急速に進む中で，本研究では，特に，自然言語で記述された文書情報を対象した情報集約手法に関して検討を行った．特に対象物を表す固有表現を中心として情報を表現する IADB を実現し，この枠組みは，評判情報や将来情報などの数多くの情報集約タスクに適用できることを示した．しかしながら，研究のサーベイを行うタスクなどでは，集約する対象や集約のための観点を，どのように定義するのかが，そもそも難しいため，本モデルの適用範囲を超えると考えている．また，IADB では，即時性を重視し，従来の全文検索サービスと同様に，集約の対象となる文書をあらかじめ全てクロールし，決められたスキーマにマッピングするアーキテクチャを採用した．しか

しながら，このアーキテクチャでは，今後，集約対象となる情報がクロールしきれないほど大規模になってくると対応することが難しくなる．

一方，セマンティック Web のコミュニティでは，データを決められた形式で Web 上に公開し，それらを連携利用することで，情報の新たな利用方法を提案し始めている．このように Web 上に公開されたデータと，自然言語で記述された文書情報とを融合させることができれば，現状の IADB の適用範囲を超えるような，新たな情報集約サービスを実現できる可能性がある．また，分散された情報をクロールせずに，連携させる枠組みを用いることで，本アーキテクチャでは対応しきれない大規模な情報に対する集約処理を実現できるかもしれない．更に，本研究では未検討であった，集約結果の信憑性に関しての何らかの解決策を見いだせる可能性もある．

しかしながら，このためには，対象物を同定することや，情報集約タスクごとに異なる観点として規定される属性名に対する統制など数多くの課題もある．そのため，自然言語処理，セマンティック Web，データベース，人工知能など様々なコミュニティと協調しながら，これらの課題に取り組むことが重要である．今後は，様々な分野のアプローチを取り入れながら，文書情報だけでなく，データも含めたあらゆる Web 上の情報を集約するシステムの実現に貢献していきたい．

謝 辞

本研究は，筆者が慶應義塾大学大学院理工学研究科後期博士課程在学中に，同大学理工学部 山本喜一教授のご指導のもとに行ったものです．山本教授には，1997 年の筆者の修士課程修了以来，今日に至るまで長きに渡り，様々な場面でご助言を頂きました．山本教授のご指導がなければ，本論文の完成には至らなかったと思います．山本教授のご指導・ご鞭撻に，心より感謝申し上げます．また，本論文の執筆に際し，多くの貴重なご意見を頂きました慶應義塾大学理工学部 山口高平教授，斎藤博昭准教授，遠山元道准教授，ならびに，本研究を進めるにあたり，数々のご助言を頂きました芝浦工業大学工学部 福田浩章准教授に厚く御礼申し上げます．

本研究は，NTT サイバーソリューション研究所，及び，NTT サイバースペース研究所において研究開発を行い，NTT レゾナント株式会社において商用サービス化を行ったシステムに関連するものです．本システムの研究開発・実用化の機会を与えてくださり，ご支援を頂きました NTT サイバーソリューション研究所 片岡良治氏，NTT アドバンステクノロジ株式会社 濱野輝夫氏，NTT レゾナント株式会社 小澤英昭氏，竹野浩氏，NTT サイバースペース研究所 森本正志氏に深く感謝申し上げます．また，プロダクトの開発に携われた NTT サイバースペース研究所 松尾義博氏，NTT コミュニケーションズ株式会社 浅野久子氏，小田寿則氏，NTT サイバーソリューション研究所 廣嶋伸章氏，NTT アドバンステクノロジ株式会社 熊本睦氏，及び，関係者各位に，心より御礼申し上げます．特に松尾氏には，本研究を進める上でも，数多くの貴重なご意見・ご指導を頂きました．ここに，心より感謝申し上げます．

また，研究開発に取り組むための基本姿勢をご指導頂きました筑波大学大学院図書館情報メディア研究科 佐藤哲司教授，大阪大学大学院言語文化研究科 林良彦教授，岡山県立大学情報工学部 菊井玄一郎教授，静岡県立大学経営情報学部 池田哲夫教授，NTT サイバーソリューション研究所 木原民雄氏，石井恵氏，米国ワシントン大学 Oren Etzioni 教授，同大学 Stephen Soderland 氏に深く感謝申し上げます．

最後に，筆者をここまで育ててくれた両親，様々な場面で心の支えとなった兄 啓一，長男 聡太，次男 賢太，筆者を気遣い私生活を支えてくれた妻 紀子に感謝の意を表したいと思います．

参考文献

[1] Agichtein, E., Gravano, L., Pavel, J., Sokolova, V. and Voskoboynik, A.: Snowball: A Prototype System for Extracting Relations from Large Text Collections, *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD '01)*, p. 612 (2001).

[2] 赤堀正剛，有澤達也，遠山元道：SuperSQL による関係データベースと XML データの統合利用，情報処理学会論文誌：データベース（TOD10），Vol. 42, No. SIG08, pp. 66–95 (2001).

[3] Allan, J. and et al.: Challenges in Information Retrieval and Language Modeling, *SIGIR Forum*, Vol. 37, No. 1, pp. 31–47 (2003).

[4] 浅野久子，平野　徹，小林のぞみ，松尾義博：Web 上の口コミを分析する評判情報インデクシング技術，NTT 技術ジャーナル，Vol. 20, No. 6, pp. 12–15 (2008).

[5] Baeza-Yates, R. A. and Ribeiro-Neto, B. A.: *Modern Information Retrieval*, ACM Press / Addison-Wesley (1999).

[6] Berry, M. J. A.，Linoff, G.，江原 淳 (邦訳)，佐藤 栄作 (邦訳)，SAS インスティチュートジャパン (邦訳)：データマイニング手法，海文堂出版 (1999).

[7] 別所克人，内山俊郎，内山　匡，片岡良治，奥　雅博：単語・意味属性間共起に基づくコーパス概念ベースの生成方式，情報処理学会論文誌，Vol. 49, No. 12, pp. 3997–4006 (2008).

[8] Bizer, C.，Heath, T.，Berners-Lee, T.，荻野達也（邦訳）：Linked Data の仕組み Linked Data – The Story So Far，情報処理，　Vol. 52, No. 3, pp. 284–292 (2010).

[9] Brin, S.: Extracting Patterns and Relations from the World Wide Web, *Selected Papers from the International Workshop on the World Wide Web and Databases (WebDB '98)*, pp. 172–183 (1998).

[10] Broad, W. J.: Study Finds Public Science is Pillar of Industry, *The New York Times* (1997).

[11] Cutting, D. R., Karger, D. R., Pedersen, J. O. and Tukey, J. W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92)*, pp. 318–329 (1992).

[12] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A.: Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science (JASIS)*, Vol. 41, No. 6, pp. 391–407 (1990).

[13] Etzioni, O., Cafarella, M. J., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. and Yates, A.: Unsupervised Named-Entity Extraction from the Web: An Experimental Study, *Artificial Intelligence*, Vol. 165, No. 1, pp. 91–134 (2005).

[14] Feldman, R. and Hirsh, H.: Mining Associations in Text in the Presence of Background Knowledge, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 343–346 (1996).

[15] Frakes, W. B. and Baeza-Yates, R. A.(eds.): *Information Retrieval: Data Structures & Algorithms*, Prentice-Hall (1992).

[16] Grishman, R. and Sundheim, B.: Message Understanding Conference- 6: A Brief History, *Proceedings of the 16th International Conference on Computational Linguistics - Volume 1 (COLING '96)*, pp. 466–471 (1996).

[17] 橋本泰一，乾　孝司，村上浩司：拡張固有表現タグ付きコーパスの構築，情報処理学会研究報告自然言語処理 (2008-NL-188)，pp. 113–120 (2008).

[18] Hearst, M. A.: Automatic Acquisition of Hyponyms from Large Text Corpora, *Proceedings of the 14th International Conference on Computational Linguistics - Volume 2 (COLING '92)*, pp. 539–545 (1992).

[19] Hearst, M. A.: Untangling Text Data Mining, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*, pp. 3–10 (1999).

[20] Hearst, M. A. and Plaunt, C.: Subtopic Structuring for Full-Length Document Access, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*, pp. 59–68 (1993).

[21] 平野　徹，松尾義博，菊井玄一郎：関係名詞らしさを用いた固有表現間の関係同定，言語処理学会第 15 回年次大会（NLP2009) (2009).

[22] 廣嶋伸章，戸田浩之，松浦由美子，片岡良治：概念ベースに基づく Web 検索のクエリタイプ判定手法とその評価，情報処理学会論文誌：データベース，Vol. 3, No. 3, pp. 33–45 (2010).

[23] 細見　格，長野伸一，岡部雅夫：次世代の医薬品開発を支える知識流通，情報処理，Vol. 52, No. 3, pp. 300–308 (2010).

[24] Huang, R. and Riloff, E.: Inducing Domain-Specific Semantic Class Taggers from (Almost) Nothing, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 275–285 (2010).

[25] 今村賢治，齋藤邦子，浅野久子：テキストからの知識抽出の基盤となる日本語基本解析技術，NTT 技術ジャーナル，Vol. 20, No. 6, pp. 20–23 (2008).

[26] 石井　恵，渡辺一成：分類体系と名詞句を用いた検索インターフェースの提案とその評価，情報処理学会研究報告ヒューマンインターフェース（1999-HI-087），pp. 1–6 (2000).

[27] 磯崎秀樹：メタルールと決定木学習を用いた日本語固有表現抽出，情報処理学会論文誌，Vol. 43, No. 5, pp. 1234–1244 (2002).

[28] 岩崎　学：データマイニングの考え方と特色，日本ファジイ学会関東支部セミナー：データの発見と活用のための技術講演資料，pp. 1–25 (2002).

[29] 金澤健介，Adam, J.，小山　聡，田中克己：Web 上の将来情報の集約的提示，Web とデータベースに関するフォーラム（WebDB Forum 2009） (2009).

[30] 笠原　要，松澤和光，石川　勉：国語辞書を利用した日常語の類似性判別，情報処理学会論文誌，Vol. 38, No. 7, pp. 1272–1283 (1997).

[31] 河合英紀，水口弘紀，土田正明：ブートストラップ式辞書構築における検索効率の向上，データベースと Web 情報システムに関するシンポジウム（DBWeb2007），pp. 36–48 (2007).

[32] 北　研二，津田和彦，獅々堀正幹：情報検索アルゴリズム，共立出版 (2002).

[33] 清木　康，金子昌史，北川高嗣：意味の数学モデルによる画像データベース探索方式とその学習機構，電子情報通信学会論文誌 D-II，Vol. J79-DII, No. 4, pp. 509–519 (1996).

[34] Kobayashi, N., Inui, K. and Matsumoto, Y.: Opinion Mining from Web Documents: Extraction and Structurization, 人工知能学会論文誌， Vol. 22, No. 2, pp. 227–238 (2007).

[35] 小原恭介，山田剛一，絹川博之，中川裕志：ウェブを利用した関連用語収集，第 3 回情報科学技術フォーラム（FIT2004），pp. 183–184 (2004).

[36] Kosala, R. and Blockeel, H.: Web Mining Research: A Survey, *SIGKDD Explorations Newsletter*, Vol. 2, No. 1, pp. 1–15 (2000).

[37] 増永良文：リレーショナルデータベースの基礎–データモデル編–，オーム社 (1990).

[38] Matthew Richardson, A. P. and Brill, E.: Beyond PageRank: Machine Learning for Static Ranking, *Proceedings of the 15th International Conference on World Wide Web (WWW2006)*, pp. 707–715 (2006).

[39] 三末和男，渡部　勇：テキストマイニングのための連想関係の可視化技術，情報処理学会研究報告情報学基礎 (1999-FI-55) (1999).

[40] 水口弘紀，土田正明，久寿居大：Weblog を対象にしたリアルタイム評判情報分析システム eHyouban，データ工学ワークショップ（DEWS 2008）(2008).

[41] Montes-y-Gómez, M., Gelbukh, A. F. and López-López, A.: Text Mining at Detail Level Using Conceptual Graphs, *Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces (ICCS 2002)*, pp. 122–136 (2002).

[42] Montes-y-Gómez, M., Gelbukh, A. F., López-López, A. and Baeza-Yates, R. A.: Flexible Comparison of Conceptual Graphs, *12th International Conference on Database and Expert Systems Applications (DEXA 2001)*, pp. 102–111 (2001).

[43] 長尾　真，佐藤理史，黒橋禎夫，角田達彦：自然言語処理，岩波書店 (1996).

[44] 永田昌明，平　博順：テキスト分類–学習理論の「見本市」，情報処理， Vol. 42, No. 1, pp. 32–37 (2001).

[45] 那須川哲哉：コールセンターにおけるテキストマイニング，人工知能学会誌，Vol. 16, No. 2, pp. 219–225 (2001).

[46] 大澤幸生：チャンス発見:アクティブマイングの最右翼，日本ファジイ学会関東支部セミナー：データの発見と活用のための技術講演資料，pp. 111–135 (2002).

[47] Ohsawa, Y., Soma, H., Matsuo, Y., Matsumura, N. and Usui, M.: Featuring Web Communities based on Word Co-occurrence Structure of Communications, *The Eleventh International World Wide Web Conference (WWW 2002)*, pp. 736–742 (2002).

[48] 大島裕明，小山　聡，田中克己：Web 集約質問処理のための検索エンジンの関係データベースインタフェース，情報処理学会論文誌：データベース（TOD36），Vol. 48, No. SIG20, pp. 50–60 (2007).

[49] 奥村　学，難波英嗣：テキスト自動要約に関する研究動向，自然言語処理， Vol. 6, No. 6, pp. 1–26 (1999).

[50] 乙守信行，湯本正典：Linked Data とメディア – メディアが Linked Data を活用する理由，情報処理， Vol. 52, No. 3, pp. 293–299 (2010).

[51] Pantel, P. and Pennacchiotti, M.: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations, *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL2006)*, pp. 113–120 (2006).

[52] Pantel, P. and Ravichandran, D.: Automatically Labeling Semantic Classes, *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, pp. 321–328 (2004).

[53] Robertson, S. E. and Walker, S.: Okapi/Keenbow at TREC-8, *NIST Special Publication 500-246: the Eighth Text REtrieval Conference (TREC 8)*, pp. 151–162 (1999).

[54] 櫻井茂明：テキストデータを活用する最新技術，日本ファジイ学会関東支部セミナー：データの発見と活用のための技術講演資料，pp. 53–85 (2002).

[55] Salton, G.: *Automatic Information Organization and Retrieval*, McGraw-Hill (1968).

[56] Salton, G., Allan, J. and Buckley, C.: Approaches to Passage Retrieval in Full Text Information Systems, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*, pp. 49–58 (1993).

[57] 佐藤宏之，飯塚京士，三島和恵：オープンガバメントとオープンデータ，情報処理，Vol. 52, No. 3, pp. 309–317 (2010).

[58] 関根　聡，竹内康介：拡張固有表現オントロジー，言語処理学会第 13 回年次大会（NLP2007），pp. 23–26 (2007).

[59] Sekine, S. and Isahara, H.: IREX: IR and IE evaluation project in Japanese, *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*, pp. 1475–1470 (2000).

[60] 新納浩幸，関根　聡：拡張固有表現タガーの作成とその問題点の考察，言語処理学会第 12 回年次大会（NLP2006），pp. 105–108 (2006).

[61] 清水　昇，三島和恵，山口章平，津田　宏，粂　照宣：Linked Data と地理空間情報，情報処理，　Vol. 52, No. 3, pp. 318–325 (2010).

[62] Sowa, J. F.: Conceptual Graphs for a Data Base Interface, *IBM Journal of Research and Development*, Vol. 20, No. 4, pp. 336–357 (1976).

[63] Takano, A., Niwa, Y., Nishioka, S., Hisamitsu, T., Iwayama, M. and Imaichi, O.: Associative information access using DualNAVI, *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS 2001)*, pp. 771–772 (2001).

[64] 武田英明：セマンティック Web と Linked Data，電子情報通信学会技術研究報告ソフトウェアインタプライズモデリング（SWIM），Vol. 108, No. 316, pp. 25–28 (2008).

[65] 武田英明：日本における Linked Data の現状と普及に向けた課題，情報処理，Vol. 52, No. 3, pp. 326–333 (2010).

[66] 徳永健伸：情報検索と言語処理，東京大学出版会 (1999).

[67] Tombros, A. and Sanderson, M.: Advantages of Query Biased Summaries in Information Retrieval, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pp. 2–10 (1998).

[68] Tomita, J., Soderland, S. and Etzioni, O.: Expanding the Recall of Relation Extraction by Bootstrapping, *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pp. 56–63 (2006).

[69] 富田準二，石井　恵，中渡瀬秀一，片岡良治：文書情報統合のためのテキスト表現モデルの提案と主題グラフを用いた実現，情報処理学会論文誌：データベース (TOD25), Vol. 46, No. SIG5, pp. 70–83 (2005).

[70] 富田準二，竹野　浩，菊井玄一郎，林　良彦，池田哲夫：グラフモデルの提案とテキスト検索システムへの適用による評価，情報処理学会論文誌：データベース（TOD13），Vol. 43, No. SIG02, pp. 94–107 (2002).

[71] 土田正明，水口弘紀，久寿居大：ブログからの対象，属性，評価のオンデマンド評判情報分析システム：eHyouban，言語処理学会第 14 回年次大会（NLP2008），pp. 899–902 (2008).

[72] 土田正明，水口弘紀，久寿居大：評判検索のための対象，属性，評価の 3 項関係のランキング法，第 22 回人工知能学会全国大会（JSAI2008） (2008).

[73] 辻井潤一ら：ヒューマンインターフェース技術に関する調査報告書，電子情報技術産業協会 (2003).

[74] 津田宏ら：特集「テキストマイニング」，人工知能学会誌， Vol. 16, No. 2, pp. 191–238 (2001).

[75] 山本一晴，獅々堀正幹，柘植　覚，北　研二：出現 URL の類似性に着目した WWW 空間からの関連語自動収集手法，情報処理学会研究報告自然言語処理（2005-NL-170），pp. 127–134 (2005).

[76] 山西健司：テキストマイニングと NLP ビジネス，JEITA 自然言語処理技術に関するシンポジウム 2003 講演資料 (2003).

著者論文目録

論文誌

(1) 富田準二，松尾義博，福田浩章，山本喜一：大規模データを対象とした文書情報集約データベースと評判分析サービスにおける検証，電子情報通信学会論文誌 D, Vol. J95-D, No. 2, pp. 250-263 (2012).

(2) 富田準二，福田浩章，山本喜一：多義性を考慮した拡張固有表現のクラス判定手法，情報処理学会論文誌：データベース，Vol. 4, No. 4, pp. 34–47 (2011).

(3) 富田準二，石井　恵，中渡瀬秀一，片岡良治：文書情報統合のためのテキスト表現モデルの提案と主題グラフを用いた実現，情報処理学会論文誌：データベース (TOD25), Vol. 46, No. SIG 5, pp. 70–83 (2005).

(4) 富田準二，竹野　浩，菊井玄一郎，林　良彦，池田哲夫：グラフモデルの提案とテキスト検索システムへの適用による評価，情報処理学会論文誌：データベース（TOD13），Vol. 43, No. SIG02, pp. 94–107 (2002).

(5) 富田準二，山本喜一：分類と階層化に基づく情報提供エージェントの実現，コンピュータソフトウェア，Vol. 15, No. 6, pp. 517–528 (1998).

国際会議

(1) Tomita, J., Soderland, S. and Etzioni, O.: Expanding the Recall of Relation Extraction by Bootstrapping, *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pp. 56–63 (2006).

(2) Tomita, J., Nakawatase, H. and Ishii, M.: Calculating Similarity between Texts Using Graph-based Text Representation Model, *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management (CIKM 2004)*, pp. 248–249 (2004).

(3) Tomita, J., Nakawatase, H. and Ishii, M.: Graph-based Text Database for Knowledge Discovery, *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters (WWW Alt. '04)*, pp. 454–455 (2004).

(4) Tomita, J., Ikeda, T. and Satoh, T.: Text mining framework based on graph-based text representation, *Proceedings of Knowledge-based Intelligent Information Engineering Systems & Allied Technologies (KES 2002)*, pp. 204–208 (2002).

(5) Tomita, J., Ikeda, T., Kihara, T. and Satoh, T.: Knowledge discovery from Mixed-model XML documents, *Proceedings of SIGIR 2002 Workshop on XML and Information Retrieval*, pp. 33–39 (2002).

(6) Tomita, J. and Kikui, G.: Interactive Web Search by Graphical Query Refinement, *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters (WWW Alt. '04)*, pp. 190–191 (2001).

(7) Tomita, J. and Hayashi, Y.: Improving Effectiveness and Efficiency of Web Search by Graph-Based Text Representation, *Poster Proceedings of the Ninth International World Wide Web Conference (WWW9)*, pp. 54–55 (2000).

国内研究会・技術レポート

(1) 富田準二：goo を支える検索技術，電子情報通信学会技術研究報告データ工学（DE），Vol. 109, No. 293, pp. 43–48 (2009).

(2) 富田準二：ビジネスインテリジェンスをめぐる展望: 意思決定を支援するテキスト集約技術，電子情報通信学会技術研究報告オフィスインフォメーションシステム（OIS），Vol. 103, No. 707, pp. 51–58 (2004).

(3) 富田準二：XML 文書検索システム LISTA，NTT 技術ジャーナル， Vol. 52, No. 2, pp. 85–91 (2003).

(4) 富田準二：グラフによるテキスト表現と XML 検索エンジンを用いた高度情報アクセス，ACM SIGMOD 日本支部 第 19 回大会 講演論文集，pp. 53–61 (2001).

(5) 富田準二，菊井玄一郎，林　良彦：構造化文書をランキング可能な全文検索システム，情報処理学会研究報告データベースシステム（DBS），Vol. 2000, No. 69, pp. 361–368 (2000).

(6) 富田準二，竹野　浩：主題グラフ及び関連度情報からの単語重要度付与を用いた情報検索システムの提案，情報処理学会研究報告情報学基礎（FI），Vol. 98, No. 109, pp. 17–24 (1998).

(7) 富田準二，山本喜一：数式を含む文書を対象にした文書校正システム，日本ソフトウェア科学会第 12 回大会，pp. 237–240 (1995).