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

富田準二[†] 石井恵[†]
中渡瀬秀一[†] 片岡良治[†]

複数の文書の内容を統合することによって，単一の文書からは得られないような重要な情報を取得することができる．このような文書情報の統合を行うためには，検索，分類等の様々なテキスト操作を柔軟に組み合わせて実行する必要がある．本稿では，リレーショナルデータモデルとのアナロジを用いたテキスト表現モデルを提案する．テキスト表現モデルは，文書を形式的な表現（テキスト表現）で表し，これらテキスト表現のリストに対する演算からなるテキスト操作の体系（テキスト表現代数）を提供する．テキスト表現に，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を採用し，主題グラフに基づくテキスト用のデータベースおよび文書情報統合アプリケーションを構築する．その動作例を用いて，主題グラフに対する演算の組合せによって，有用な情報が得られることを示すとともに，テキスト表現モデルのカバー範囲を明らかにする．さらに，主題グラフが，一般に広く利用されているタームベクトルよりもテキスト表現として適していることを，演算のタスク適用性および分析結果の可読性の観点から示す．また，計算量に関する考察から主題グラフが大規模文書集合にも適用可能であることを示す．

Text Representation Model for Integrating Document Contents and Its Implementation Using Subject Graphs

Junji Tomita,[†] Megumi Ishii,[†] Hidekazu Nakawatase[†]
and Ryoji Kataoka[†]

Integrating the contents of several documents reveals important facts, that can not be acquired from a single document. The integration requires heterogeneous combinations of text handling operations such as text search and clustering. Drawing an analogy with the relational data model, we propose a text representation model that represents documents in a formal manner, i.e. text representation, and that provides a text representation algebra that consists of procedures for handling the lists of the representations. We use subject graphs as the representation; node weight is used to represent the significance of each term, and link weight is used to represent that of each term-term association. This paper introduces a graph-based text database based on the model and an application for integrating document contents. Examples show that the proposed technique can discover important facts. Furthermore, evaluations show that subject graphs are more suitable for representation than term vectors with regard to the applicability of procedures and readability. We also show the limitations of the model and that the computational complexity of subject graphs is reasonable.

1. はじめに

Web を中心とした大規模文書データの中には様々な有用な情報が含まれている．このような情報は単一の文書から得られるものばかりではなく，複数の文書情報を統合することによって，初めて得られるものも数多くある．たとえば，ある製品に対する評判や，競合他社の動向等は，通常，単一の文書だけからは得ることができない．そのため，「複数の文書に記述されている内容を統合し，ユーザに分かりやすく提示する処理」が必要となる．本稿では，この処理を表すものとして，「文書情報統合」という用語を用いる．

文書情報統合を実現するためには，検索，分類等の複数の有効なテキスト操作を，ユーザの用途に応じて柔軟に組み合わせる必要がある．Hearst[1] は，Broad[2] の研究を例にとり，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．現在，情報検索の分野では，文書検索，分類等の研究が活発に行われている．

† 日本電信電話株式会社，NTT サイバーソリューション研究所
NTT Cyber Solutions Laboratories, NTT Corporation

しかしながら，これらの研究は，個々の機能の精度や性能向上を目指したものがほとんどであり，このような複数のテキスト操作を統合的に扱うための枠組みについての研究は進んでいない[3]．

一方，構造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（RDM）がある[4]．RDMに基づくリレーショナルデータベース（RDB）を用いると，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．このようにアプリケーションとデータを切り放し，低い開発コストで様々な用途に応じたデータ利用が実現されている．しかしながら，RDBは，主に構造化されたデータのみを想定し，テキストの格納には向いていない．特に，文書どうしの照合といった曖昧性のあるテキスト操作の扱いや，大量のテキストデータを扱う場合の処理効率に問題がある．

本稿では，文書情報統合を実現するために，個別の機能として提供されている検索や分類等のテキスト操作を統合的に扱うモデルとして，テキスト表現モデルを提案する．我々は，RDMがデータ操作を柔軟に組み合わせて実行できる点に着目し，RDMにおけるリレーションとリレーショナル代数と似た形式化を文書データに対して行う．テキスト表現モデルは，文書を形式的な表現（テキスト表現）で表し，テキスト表現を要素とするリストに対する演算からなるテキスト操作の体系（テキスト表現代数）を提供する．検索や分類といったテキスト操作を，テキスト表現リストに対する演算として定義することによって，これらの柔軟な組合せを可能とする．これにより，様々な文書情報統合の場面に適用できるテキスト用のデータベース（以下，テキストデータベース）を実現することができる．

テキスト表現モデルにおいては，どのようなテキスト表現を用いるのかが非常に重要である．文書の内容をより忠実に表し，コンピュータ上での扱いが容易な表現を用いれば，様々な有効なテキスト操作を実現することができる．ここでは，テキスト表現の採用基準として，(a) 文書変換可能性，(b) 演算のタスク適用性，(c) 可読性を定め，この基準に基づき，主題グラフを採用する．主題グラフ[5]は，単語の重要度をノードの重み，単語どうしの関連の強さ（関連度）をリンクの重みとしたグラフである．主題グラフは，単語とその重みの集合で文書を表現するタームベクトル[6]の拡張であるため，その利点を継承しつつ，単語どうしの文書内での関連性を利用できることが大きな特徴である．

2章では，文書情報統合の例から複数のテキスト操作の組合せの必要性と関連研究を示す．3章では，RDMとのアナロジからテキスト表現モデルを提案し，テキスト表現の採用基準を示す．4章では，いくつかのテキスト表現の候補の中から主題グラフを採用し，複数の有効な演算を定義する．5章では，グラフに基づくテキストデータベース（GTB: Graph-based Text dataBase）と文書情報統合アプリケーションの構築について述べる．6章では，分析ステップの実現例とテキスト表現モデルのカバー範囲を示すとともに，テキスト表現の3つの採用基準と計算量の観点から主題グラフを評価する．7章では，まとめを述べる．

2. 文書情報統合と関連研究

2.1 Web上の文書情報統合の例

Web上から，ユーザが「最も評判の良いRDBのベンダはどこか?」,「そのベンダの製品に対するユーザの評価は?」「競合製品にはどのようなものがあるのか?」といった情報を取得する場合を考える．ユーザはこのような情報を取得するために，たとえば，以下の分析ステップを行う．

(1) キーワードとして’リレーショナルデータベース’を指定して検索を行う．
(2) すべての検索されたWebページの中からリンクを抽出し，各ベンダごとに参照数をカウントする．
(3) 最も参照数が多いベンダを参照しているWebページを収集する．
(4) 収集したページを，RDBについて書かれている部分に着目しながら，分類，要約する．

ステップ(1)，(2)によって，最も注目を集めているベンダ名が分かり，ステップ(3)，(4)によって，そのベンダや競合他社の製品名や評判を取得できる．

このような分析を行う場合，ユーザは検索のみツールを利用し，その後は各文書をユーザ自身が実際に読み要約していくのが実情である．この理由として，各テキスト操作は個別の機能として提供されてきてはいるが，これらを，ユーザのその場の用途に応じて低いコストで柔軟に組み合わせて利用できないことがあげられる．したがって，複雑な分析ステップからなる文書情報統合タスクを効率的に行うためには，複数のテキスト操作を組み合わせるためのモデルが必要である．さらに，このモデル上で，テキスト操作の組合せを「分析式」という形で記述できれば，日々更新される情報ソースに対して継続して上記のような分析ステップを実行することもできる．

2.2 関連研究

文書情報統合において，ユーザへの結果の提示方法を特に文書形式に限定すれば，これは複数文書要約とほぼ同じものである．複数文書要約では，1)「関連文書の収集」，2) 重要箇所，共通点，相違点の抽出，3)「要約文書の生成」といった手順を想定し，この手順に沿って各機能の研究が行われている[7]．しかしながら，これらの機能を柔軟に組み合わることを目的とした研究は行われていない．そのため，2.1 節の例のような，「参照数が最も多いものだけを収集する」といった操作を，その場の用途に応じて利用することができない．

一方，データベースの分野では，RDM 上のデータ操作に加えて，文書のランキング検索を統合的に扱う手法に関する研究も数多く行われている．Fuhr らは，RDM の各タプルに各リレーションへ所属する確率を付与する確率リレーショナルモデルを提案している[8]．Fuhr らの手法では，検索スコアを，各タプルが検索結果リレーションへ所属する確率として取り扱うことができるため，DB 検索（事実に基づく検索）上に IR 検索（ランキング検索）を統合できる．しかしながら，ランキング検索以外のテキスト操作は考慮されておらず，クラスタリング等を行うためには，モデルの外部に別機能として構築する必要がある．また，質問文のタームベクトルを格納したリレーションと各対象文書との join 演算として類似文書検索を記述しなければならない等，問合せが複雑である．

このように，文書情報を対象とし，検索だけでなく分類等のテキスト操作を柔軟に結び付けることができるモデルは提案されていない．

3. テキスト表現モデル

3.1 テキスト表現モデル概要

我々は，様々な文書情報統合タスクへの適用を目指し，複数のテキスト操作を柔軟に組み合わせて実行できるテキスト表現モデルを，RDM とのアナロジに基づき提案する．RDM は，構造化されたデータを形式的な表現（リレーション）で表し，リレーション集合に対する演算からなるデータ操作の体系（リレーショナル代数）を提供する．そのため，複数のデータ操作を柔軟に結び付けて実行することができる．これと同様に，テキスト表現モデルは，文書の形式的な表現（テキスト表現）と，テキスト表現リストに対する演算からなるテキスト操作の体系（テキスト表現代数）を提供する．RDM との対応関係を**表 1** に示す．

テキスト表現 t は，内容表現 C と属性表現 A によって構成される．

表 1 リレーショナルデータモデルとテキスト表現モデル
Table 1 RDM and Text Representation Model.

	リレーショナルデータモデル	テキスト表現モデル
対象	構造化データ	テキストデータ
形式的な表現	リレーション	テキスト表現リスト
操作の体系	リレーショナル代数	テキスト表現代数

内容表現 C： 文書の内容を表すための表現

属性表現 A： 文書にあらかじめ付与された属性（作成日等）および，演算の結果，付与される属性（検索スコア等）を表すための表現

たとえば，テキスト表現の例として内容表現にタームベクトル，属性表現に属性名と属性値のペアの集合を用いた場合は以下のようになる．

$$t = (C, A)$$
$$C : \{(i, w_i)\}$$
$$A : \{(n_k, v_k)\}$$

ここで，w_i は単語 i の重要度，n_k，v_k は属性 k の名前と値を，$\{x\}$ は，x を要素とする集合を表す．

このようにテキスト表現が，内容表現と属性表現を持つことで，文書の内容に対する操作と属性に対する操作を統合的に扱うことができる．たとえば，内容に基づく類似文書検索の結果を，著者といった属性ごとに集計することができる．また，後述するように，属性表現の中に，検索スコアや分類カテゴリ ID 等の演算の結果を含めることによって，非常にシンプルな形で，テキスト表現代数を定義できる．

テキスト表現代数は，以下の形式の演算によって構成されるテキスト操作言語の体系である．

$$func_{arg}(T_1, T_2, T_3, \ldots, T_n) \rightarrow T_r \qquad (1)$$

ここで，$func$ は演算の名前，arg は $func$ に固有の引数である．T_1〜T_n，T_r は，それぞれテキスト表現リストである．このように，入力が n 個のテキスト表現リスト，出力が 1 個のテキスト表現リストであるため，任意の演算の出力を，任意の演算の入力として使用することができる．この形式の演算として，検索や分類といったテキスト操作を定義することによって，これらのテキスト操作を柔軟に組み合わせて実行することができる．6.1.1 項では演算の組合せによる分析ステップの実現例を示す．

3.2 モデルの全体構成とテキスト表現の採用基準

実際の文書への適用を考えた場合，当然ながら文書からテキスト表現を自動生成する必要がある．また，各演算の出力は，テキスト表現リストであるので，最終的な結果もテキスト表現リストとなる．したがってテキスト表現リストといったコンピュータの内部表現をユーザに対し理解しやすい形で提示する必要がある．

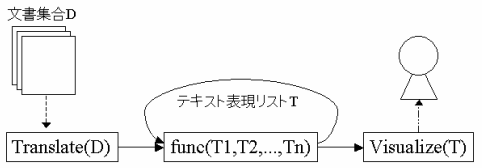

図 1 テキスト表現モデルの全体構成
Fig. 1 Overview of Text Representation Model.

そのため，以下の 3 つのコンポーネントからテキスト表現モデルは構成される（**図 1**）．

(A) **Translate**$(D) \to T$
文書集合 D を入力とし，テキスト表現リスト T に変換し出力する文書変換関数．

(B) $func_{arg}(T_1, T_2, \ldots, T_n) \to T$
n 個のテキスト表現リストを入力とし，1 つのテキスト表現リストを出力する演算集合．

(C) **Visualize**(T)
テキスト表現リスト T を入力とし，その内容を可視化しユーザに提示する可視化関数．

テキスト表現には，様々なものが考えられるが，その採用には，上記の各コンポーネントに対応した以下の 3 つの基準を考慮する必要がある．

(a) **文書変換可能性**
対象文書からテキスト表現が自動生成できる，すなわち，Translate(D) が定義できる必要がある．対象文書のドメインが限られ，そのドメインの特徴が利用できる場合は，深い言語解析に基づく複雑な表現が可能である．一方，ドメインが限定されていない場合は，簡単な統計量等で作成可能な表現が適している．このように，対象文書に基づき，この基準を検討する必要がある．

(b) **演算のタスク適用性**
必要十分な演算（$func$）が定義できる必要がある．多くの文書情報統合のタスクにおいては，検索および分類は必須のテキスト操作であり，当然，その精度が高いことが望ましい．また，重要部分を抽出する，まとめる，比較する等の操作も重要である．当然，演算の種類や精度が不十分であると，ユーザの目的を達成することができない．そのため，対象とするタスクに基づき，この基準を検討する必要がある．

(c) **可読性**
ユーザに，分析の結果を分かりやすく提示できる可視化関数（Visualize(T)）が定義できる必要がある．この際，可視化結果だけを見ることを想定する場合（informative な可視化）と，可視化結果だけでなく原文書の参照が許される場合（indicative な可視化）のどちらなのかを考慮することが重要である☆．そのため，ユーザが求める結果の信頼性，結果確認のためのユーザの許容負荷に基づき，この基準を検討する必要がある．

上記の 3 つの基準はトレードオフの関係にあり，すべてを完全に満たすことは難しい．たとえば，(b) 演算のタスク適用性を満たすために，より高レベルの表現を使用すると (a) 文書変換可能性を満たさなくなる．したがって，想定される対象文書，タスク，要求される結果の信頼性や許容ユーザ負荷に応じてテキスト表現を採用する必要がある．

また，テキスト表現は，属性表現と内容表現によって構成されるが，属性表現については，多くの場合，属性名と属性値のペアの集合によって表現できる．そこで，ここでは，属性表現は属性名と属性値のペアの集合に限定し，内容表現を中心に扱う．

4. グラフに基づくテキスト表現モデル

4.1 主題グラフによる内容表現

テキスト表現を採用する際には，(a) 文書変換可能性，(b) 演算のタスク適用性，(c) 可読性の 3 つの採用基準を，対象文書，想定タスク，結果の信頼性と許容ユーザ負荷に基づき考慮しなければならないことを述べた．我々は，特に 2.1 節で示した例のような，「Web 文書を対象に，検索・分類を行い，関連する内容をまとめる」というタスクを想定している．また，Web 文書は通常簡単にアクセスすることができるため，必要に応じて原文書確認のためのユーザ負荷を許すこととする．このような想定のもとで，テキスト表現の採用を判断する．

文書の内容表現として，一般に，タームベクトルと呼ばれる単語とその重要度の組の集合が用いられている[6]．タームベクトルは，単語抽出を行い，その重要度を統計的な手法によって計算することで作成できるため，Web のような雑多な文書に対しても文書変換を定義することができる．また，内積やコサインによる文書どうしの照合，ベクトルの合成ができるため，検索や分類に対応する演算を定義可能である．しかしながら，文書を単語の集合で表しているため，単語間

☆ これらの用語は通常，利用目的に応じて要約のタイプを分けるときに使用される[9]．
indicative： 原文の適切性を判断する等，原文を参照する前の段階で用いる．
informative： 原文の代わりとして用いる．

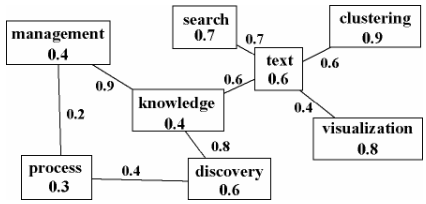

図 2 主題グラフの例
Fig. 2 An example of a subject graph.

の関係を利用した演算の定義ができず，可視化は単語のリストを表示するのみとなる．また，LSI（Latent Semantic Indexing）[10]，言語モデリング[3]，テキストモデル[11]，概念ベース[12] 等が提案されているが，いずれも検索や分類に特化した表現である．これらの手法では，単語の語彙そのものの近さ等を扱うことはできるが，各文書は，タームベクトルと同様に，単語や概念の集合として表現され，ある文書内で単語どうしがどのような関係にあるかは表現できない．また，抽象化された概念で文書が表現された場合には，文書内容の可視化は難しい．

一方，他の内容表現として conceptual graphs [13] と呼ばれる単語をノード，単語間の関係（Object，Agent 等）をリンクとしたグラフがある．conceptual graphs は文書中の各文を深く解析することによって作成される．この表現では，単語の関係を用いた可視化や複雑な演算を定義可能である．しかしながら，実際に conceptual graphs を検索や分類に利用した例はあるもの[14),15)]，限られたドメインの比較的小規模な文書でしかその効果は示されていない．これは，文書全体ではなく各文の深い解析に基づくために，辞書が必要になり，また，多数の文からなる長い文書や文書集合を表現する際には，表現が巨大になりすぎるからである．また，2 単語の文中での共出現（共起）関係を利用してグラフを生成する KeyGraph が提案されている[16]．しかしながら，KeyGraph が利用されるのは，主に文書からのキーワード抽出や可視化[17] であり，単語間の関連性を利用したテキスト操作を想定したものではない．

我々は，上記の考察に基づき内容表現に主題グラフを採用する．主題グラフは単語の重要度をノードの重みとし，単語間の関連度をリンクの重みとしたグラフである（**図 2**）．図 2 では，単語「knowledge」の重要度が 0.4 で，単語「knowledge」と「text」の関連度が 0.6 であることを表している．主題グラフを内容表現に用いたテキスト表現を形式的に記述すると以下のようになる．

$$t = (C, A)$$
$$C \ : \ (\{(i, w_i)\}, \{(i, j, a_{ij})\})$$
$$A \ : \ \{(n_k, v_k)\}$$

a_{ij} は，単語 i と単語 j の関連度である．上記定義から明らかなように主題グラフはタームベクトルに単語どうしの関連度を表すベクトル（関連度ベクトル）を追加したものである．単語の頻度情報から単語の重要度，2 つの単語の共起頻度からこれらの単語間の関連度を計算するため，広範囲の文書で文書変換を定義することができる．重要度に加え，関連度の内積や加算を行えば主題グラフ間の類似度計算や合成ができる．そのため，検索や分類等の重要なテキスト操作に対応する演算を容易に定義することができる．特に複数の主題を持った文書や合成された文書間の類似度をタームベクトルよりも主題グラフは精度良く計算できることが示されている[5]．さらに，関連度を用いた複雑な演算（4.3 節の Extract 等）を定義できる．また，可視化の際には単語リストだけではなく，単語間の関連も同時に可視化することができる．このような単語間の関連の可視化はいくつかの先行研究で実施され，その有効性が示されてきている[18)~21)]．このように主題グラフは，広い (a) 文書変換可能性を持ち，タームベクトルの利点を継承しつつつ，より (b) 演算のタスク適用性と (c) 可読性を満たすものである．詳しい評価は，6.2 節に述べる．以下，内容表現に主題グラフを持つテキスト表現をグラフ表現，グラフ表現に対する演算からなるテキスト操作の体系をグラフ表現代数と呼ぶ．

4.2 グラフ表現の作成方法

各文書から，グラフ表現を作成するには，文書から単語を抽出し，単語の重要度，単語間の関連度を計算し，属性を抽出する．属性の抽出方法には，様々なものが考えられるが，本稿では内容表現を中心に扱っているため，簡単なパターンマッチを用いることとする．

単語の重要度は，単語の出現頻度を用いて tf*idf 法によって計算する．tf*idf 法には，出現頻度の正規化手法がいくつかあるが，ここでは，freeWAIS-sf [22] で使用されている方法を用いる．単語 i の文書 t における単語の重要度 w_{it} は以下のように求める．

$$w_{it} = \frac{tf'_{it}}{\sqrt{\sum_r tf'^2_{rt}}} \times \log \phi_i \tag{2}$$

$$tf'_{it} = 0.5 + 0.5 \times \frac{tf_{it}}{\max_r tf_{rt}} \tag{3}$$

$$\phi_i = \frac{N}{n_i} \tag{4}$$

tf_{it} は文書 t における単語 i の出現回数，N は文書

集合中の文書の総数，n_i は単語 i が出現する文書数を表す．

単語間の関連度は，文や節や特定の単語数内での 2 つの単語の共起頻度を用いて計算する．単語 i と単語 j の文書 t 内での関連度 a_{ijt} は，式 (2) とのアナロジから以下のように計算する．

$$a_{ijt} = \frac{tc'_{ijt}}{\sqrt{\sum_{r,s} tc'^2_{rst}}} \times \log \psi_{ij} \quad (5)$$

$$tc'_{ijt} = 0.5 + 0.5 \times \frac{tc_{ijt}}{\max_{r,s} tc_{rst}} \quad (6)$$

$$\psi_{ij} = \frac{N}{n_{ij}} \simeq \phi_i \times \phi_j \quad (7)$$

ここで，tc_{ijt} は文書 t における単語 i と j の共起回数，n_{ij} は単語 i と j が共起する文書数であるが，この値の計算コストは高いので，n_{ij} の代わりに n_i，n_j を用いる上記の近似式を用いた．

4.3 グラフ表現代数

グラフ表現リストに対する演算として，テキスト表現とは独立の演算，グラフ表現に対する基本的な演算，有効なテキスト操作に対応する演算を定義する．これらの演算は，必ずしもあらゆるタスクにおいて必要十分とはいい難いが，4.1 節で述べた想定タスク，WWW 上のテキスト情報の知的統合の例[23] を考慮し，必要なものを定義した．

テキスト表現とは独立の演算

リストに対する基本的な演算を定義する．各演算では，要素（テキスト表現）の内容は変更しない．これらは基本的なリスト演算であるため，LISP[24] を参考にした．

LCar$(T) \to T_r$： T の先頭要素を取得し，その要素を 1 つ持つリスト T_r を返す．LISP における car と list の組合せである．

Cdr$(T) \to T_r$： T の先頭要素を除いたリスト T_r を返す．

Append$(T_1, T_2, \ldots, T_i, \ldots, T_n) \to T_r$： 複数のテキスト表現リスト T_i を連結した，リスト T_r を返す．

上記，3 つの演算を組み合わせることによって，テキスト表現リストの任意の部分リストを取得することができる．たとえば，T の 2 番目と 3 番目の要素からなるリストの生成は，

$$\mathrm{Append}(\mathrm{LCar}(\mathrm{Cdr}(T)), \mathrm{LCar}(\mathrm{Cdr}(\mathrm{Cdr}(T))))$$

である．

グラフ表現に対する基本演算

主題グラフの線形結合は，内容をまとめるための合成や，比較するための差分等を定義するために必須である．そこで，属性に対する選択演算を加えた 3 つの基本演算を定義する．

Add$(T_1, T_2) \to Tr$： T_1 の各グラフ表現へ，T_2 の平均グラフを加算し，グラフ表現リスト T_r を返す．T_2 の平均グラフとは，T_2 の全要素のタームベクトル，関連度ベクトルそれぞれの重心ベクトルを持つグラフである．

Multiple$_\alpha(T) \to T_r$： T の各グラフ表現の重要度，関連度を α 倍したグラフ表現リスト T_r を返す．

Select$_c(T) \to T_r$： T の中で属性集合が条件 c を満たすグラフ表現を選択しそのリスト T_r を返す．c として記述できる範囲は詳しくは述べないが，SQL の where 句[25] に記述できるもののサブセットとした．

有効なテキスト操作に対応する演算

4.1 節の想定タスクから，検索・分類は必須の操作である．また，ある内容に関連する部分を抽出する操作も重要である．そこで，以下の 3 つの演算を定義する．

Search$_n(T_1, T_2) \to T_r$： T_2 の平均グラフを質問グラフとし，T_1 の各グラフ表現と質問グラフとの類似度を計算し，類似度の高い n 件からなるグラフ表現リスト T_r を返す．この際，属性名' 検索スコア' の属性値を各類似度に設定する．ここで，類似度は，論文 5) に従いタームベクトルの内積と関連度ベクトルの内積の線形和によって計算する．

Clustering$_k(T) \to T_r$： T の各グラフ表現を類似度の近い k 個のクラスタに分類する．属性名' 分類カテゴリ ID' の属性値を各カテゴリの ID 値に設定したグラフ表現リスト T_r を返す．

Extract$_l(T_1, T_2) \to T_r$： T_2 の平均グラフを条件グラフとして，T_1 の各グラフから l 個のノードからなる部分グラフを抽出し．各部分グラフを要素とするグラフ表現リスト T_r を返す．

多くの演算は，内容表現がタームベクトルの場合でも定義可能であるが，Extract は，単語どうしの関連を利用しないと定義できない☆．そこで，ここでは，主題グラフに特徴的な演算である Extract の部分グラフ抽出アルゴリズムとその利用方法の詳細を述べる．

条件グラフ G_c が与えられた際に，主題グラフ G_t

☆ タームベクトルの各単語に出現位置を持たせれば Extract と似た演算が定義できる．ただし，この場合は，Extract の実行時に，出現位置から共起関係の計算を行うことになり，文書変換時に主題グラフを作成するか，演算実行時に作成するかの違いとなる．当然，単語位置を含めたテキスト表現を用いる場合，他のすべての演算においても単語位置の扱いを考慮する必要がある．

から，部分グラフを抽出するアルゴリズムは以下のとおりである．

(1) G_t 上のすべての単語の重要度と単語間の関連度を正規化する．

$$w'_{it} = \frac{w_{it}}{\max_r w_{rt}} \quad (8)$$

$$a'_{ijt} = \frac{a_{ijt}}{\max_{rs} a_{rst}} \quad (9)$$

(2) G_c 上の各単語 k ごとに，k と，G_t 上の各単語 i との間の間接的な関連の強さを表す値を要素とするベクトル $\boldsymbol{n_{kt}}$ を作成する．

$$\boldsymbol{n_{kt}} = (n_{1kt}, \ldots, n_{ikt}, \ldots, n_{mkt}) \quad (10)$$

n_{ikt} は，k が G_t 上になければすべての i において 0 とする．そうでなければ，以下のように計算する．

(a) G_t 上の k から i に至るすべてのパスを抽出する．

(b) 各パス p について，パス上のすべての単語の重要度と単語間の関連度の積を各パスの重みとする．

$$h_{ikpt} = hw_{ikpt} \times ha_{ikpt} \quad (11)$$

ここで，

$$hw_{ikpt} = \prod_{r \text{ on path } p} w'_{rt} \quad (12)$$

$$ha_{ikpt} = \prod_{r,s \text{ on path } p} a'_{rst} \quad (13)$$

である．

(c) パスの重みの最大値を求め，n_{ikt} とする．

$$n_{ikt} = \max_p(h_{ikpt}) \quad (14)$$

n_{ikt} は，最短経路問題のためのダイクストラのアルゴリズム[26] を用いて，効率的に計算することができる．

(3) G_t 上の各単語 i について，近接度（nw_{ict}）を n_{ikt} の重み付きの和によって計算する．

$$nw_{ict} = \sum_k w_{kc} \times n_{ikt} \quad (15)$$

(4) nw_{ict} の上位 l 個の単語を持つノードとそれらの間のリンクからなる部分グラフを抽出する．

図 3 の G_t が与えられ，$k1$='discovery'，$k2$ ='text' とし，$w_{k1c} = 0.8$，$w_{k2c} = 0.2$ からなるタームベクトルを持つ G_c が与えられたときの部分グラフ抽出の例を示す．まず，i = 'management' の近接度を計算する．図 3 のパス $P1$ について，h_{ik1P1t} は，以下のように計算される．

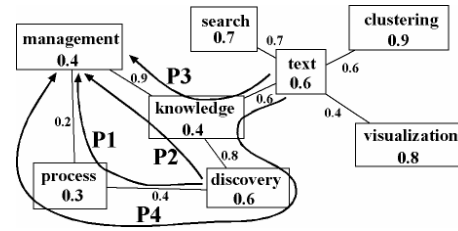

図 3 近接度の計算例
Fig. 3 An example of calculating neighboring weight.

$$\begin{aligned} h_{ik1P1t} &= hw_{ik1P1t} \times ha_{ik1P1t} \\ &= 0.6 * 0.3 * 0.4 \times 0.4 * 0.2 \\ &= 0.00576 \end{aligned}$$

同様に $h_{ik1P2t} = 0.06912$ と計算されるため，$n_{ik1t} = 0.06912$ となる．同様に，$n_{ik2t} = 0.05184$ と計算される．その結果，近接度 nw_{ict} は，以下のように計算される．

$$nw_{ict} = 0.8 * 0.06912 + 0.2 * 0.05184 = 0.06566$$

このようにして計算される近接度の大きな l 単語からなる部分グラフを抽出する．したがって，このアルゴリズムは，重要であり，かつ条件グラフとの関連の強い単語集合からなる部分グラフを抽出することができる．そのため，たとえば，条件グラフとして，検索条件と同じものが与えられた場合は，query-biased summarization[27] と同様に検索条件に対応した部分グラフを抽出できる．抽出した部分グラフは，そのまま可視化することもできるし，次の演算の入力にも利用できる．

ここに示した以外にも，入力が複数のグラフ表現リスト，出力が 1 つのグラフ表現リストである演算ならば，どのようなものでも追加することが可能である．当然ながら，上記に述べた演算は，任意の順序で，組み合わせて実行することができる．詳しくは，6.1.1 項で述べる．

4.4 可 視 化

主題グラフは，単語の重要度と単語間の関連度によって構成される．したがって，最も直接的な可視化手法は，バネモデルによる可視化である．バネモデル[28] は，リンクの重みに比例した引力と，ノード間の距離に応じた斥力を定め繰返し計算によって，各ノードの配置を決定する方法である．主題グラフにバネモデルを適用した場合は，関連度の高いものが近くに配置できるという特徴を持っている．さらに，重要度をノードの大きさ，関連度をリンクの太さに対応させる（図 6）．

明らかに，バネモデルによる方法は informative な可視化手法としては不十分であるため，原文書に対するアクセス手段を提供する．具体的には，可視化され

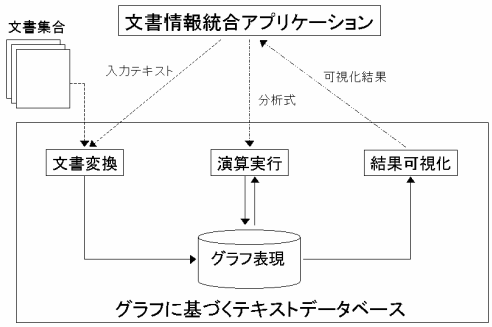

図 4 グラフに基づくテキストデータベースと文書情報統合アプリケーション
Fig. 4 Graph-based text database and an integrating document contents application.

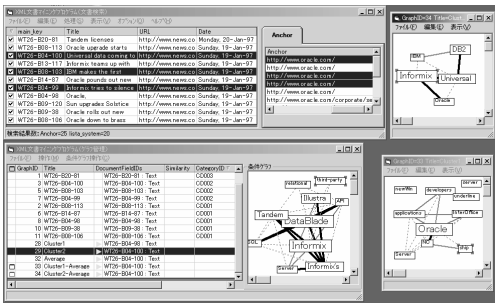

図 5 文書情報統合アプリケーションのインタフェース
Fig. 5 The user interface of the integrating document contents application.

たグラフ上の任意のノード（単語）をクリックすると，その単語を含む原文書の文を表示する．このように少ないユーザ負荷で結果の確認ができる．

5. グラフに基づくテキストデータベースと文書情報統合アプリケーション

グラフ表現モデルを用いて，テキストデータベース（GTB: Graph-based Text dataBase）を構築した．また，GTB を用いた文書情報統合アプリケーションを実装した．これらの構成を**図 4** に示す．GTB は，文書変換，演算実行，結果可視化モジュールからなり，それぞれ，3 章で述べた方法で，文書からのグラフ表現作成，演算の実行，結果の可視化を行う．主題グラフは，文書 ID をキーとし，各タームベクトルおよび各関連度ベクトルをそれぞれ値とする独自のテーブルに格納される．また，これとは別に，検索の高速化のために単語をキーとする転置インデクスを作成している．検索時には転置インデクスを用いて対象文書の同定を行う．処理対象の文書が同定された後は，文書 ID をキーとしてタームベクトル，関連度ベクトルを取得し各演算を実行する[5]．また，属性表現は，属性名を列名とするリレーションとして，RDB に格納する[29]．

文書情報統合アプリケーションのユーザインタフェースを**図 5** に示す．左上は，文書属性ウインドウで，各文書が持つ属性を表示する．左下は，グラフ管理ウインドウで，1 つまたは複数の文書に対応したグラフ表現の属性を表示する．その他のウインドウは，グラフ表示ウインドウで，選択されたグラフ表現がバネモデルによって可視化される．ユーザは，操作対象のレコード（グラフ表現）を選択し，演算を指定することによって，インタラクティブに演算を実行することができる．

6. 評　　価

6.1 テキスト表現モデルの評価

テキスト表現モデルの有効性を示すため，本モデル上で実際の文書情報統合がどのように実現できるのかの例を示す．また，テキスト表現モデルのカバー範囲について考察する．

6.1.1 分析ステップの実現例

まず，基本演算を基に以下の演算を定める．

- $\mathrm{Merge}(T) = \mathrm{Add}(\mathrm{LCar}(T), \mathrm{Merge}(\mathrm{Cdr}(T)))$
  T の全要素を合成した主題グラフを 1 つ持つグラフ表現リストを返す．
- $\mathrm{MergeGroup}_{n_k}(T) = \mathrm{Append}(\mathrm{Merge}(\mathrm{Select}_{n_k=a}(T)), \mathrm{Merge}(\mathrm{Select}_{n_k=b}(T)), , , ,)$
  属性名 n_k の属性値ごとに，合成を行い合成されたグラフ表現リストを返す．
- $\mathrm{Subtract}(T_1, T_2) = \mathrm{Add}(T_1, \mathrm{Multiple}_{-1}(T_2))$
  T_1 の各グラフ表現から，T_2 の平均グラフを減算したグラフ表現リストを返す．

本システムを用いて，WT2G[30] に対して 2.1 節に示した分析ステップを実行する例を以下に示す．WT2G は，TREC Web Track☆で使用されたコンテンツであり，1997 年に集められた 2 GB の Web ページからなる．WT2G の各 Web ページの`<title>`と`<body>`タグに含まれるテキストから主題グラフを作成し，`<title>`，各ページの URL，最終更新日時，`<a>`（アンカータグ）の href 属性の値を属性として抽出した．共起を計算する範囲は，処理効率の観点から clause（カンマまたはピリオドまで）とした．

(S1) ユーザは，質問文'relational database' を入力し，対象を全文書 D_{all} として検索を実行する．Translate({ 質問文 }) と Translate(D_{all}) に

☆ TREC http://trec.nist.gov/

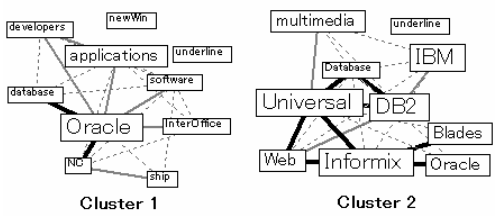

図 6 2 つのクラスタの可視化結果
Fig. 6 Visualization results of two clusters.

よって質問文に対応したグラフ表現リスト Q と，全文書 D_{all} に対応したグラフ表現リスト T が作成され，$\mathrm{Search}_n(T, Q)$ が実行される．その結果 $T1$（256 文書）を表示する．

(S2) ユーザは，各ベンダを参照しているページ数を画面上で集計し，最も参照数が多いのは Oracle であると分かる．そこで，`<href LIKE '%oracle%'>`を条件 c として，$\mathrm{Select}_c(T1)$ を実行する．`<a>`の href 属性に oracle を含むページ $T2$（11 ページ）を表示する．

(S3) ユーザは，$\mathrm{Clustering}_k(T2)$ を実行する．k は文書数 11 を考慮し，5 とした．結果として分類カテゴリ ID 属性に属性値の入ったグラフリスト $T3$ を得る．各分類カテゴリに分類された文書数は，(7, 3, 1, 1, 1) であった．

(S4) ユーザは，$\mathrm{MergeGroup}_{cid}(T3)$ を実行する．ここで，cid（分類カテゴリ ID）の異なる属性値は 5 個なので，分類カテゴリごとに合成された 5 つのグラフ表現を要素とする $T4$ が出力される．

(S5) ユーザは，$\mathrm{Subtract}(T4, T2)$ を実行する．この操作によって，各分類ごとに合成されたグラフ表現から全体の平均グラフを減算したグラフ表現リスト $T5$ が出力される．

(S6) ユーザは，$\mathrm{Extract}_l(T5, Q)$ を実行する．この操作によって，$T5$ の各グラフから'relationl database' に関連する部分グラフが抽出される．

これらのステップによって，RDB に関連し，oracle を参照している Web ページを内容の近いものに分類し，各分類されたカテゴリに特に特徴的で，かつリレーショナルデータベースと関連が深い単語集合からなるグラフ表現リストが得られる．**図 6** は，このようなグラフ表現のうち，Cluster1（7 個の文書を含む）および Cluster2（3 個の文書を含む）を可視化したものである．これらのステップによって，得られた情報は以下のとおりである．

- 最も注目を集めているベンダはオラクルである．
- オラクルのプロダクトの名前は，InterOffice☆である．製品の仕様や評判は，Cluster1 に割り当てられている文書から読み取れる．
- オラクルの競合製品の名前は，Informix と DB2 である．製品の比較は，Cluster2 に割り当てられている文書から読み取れる．

ただし，たとえば，InterOffice は，Oracle の製品であるといった情報は，可視化結果だけから読み取ることは不可能であり，ユーザは必要に応じて可視化されたグラフ上の単語を選択し，その単語を含む文を確認する必要がある．このように，文書をテキスト表現に変換し，複数の演算を実行し，その結果を可視化し，対応部分を原文書で確認することによって，複数の文書にわたって記述されている有用な情報を取得することができる．

上記のステップでは，インタラクティブに演算を実行する例を示したが，上記のステップを実現する分析式を以下のように記述することもできる．

$$
\begin{aligned}
Q &= \mathrm{Translate}(\{\text{ 質問文 }\}) \\
T &= \mathrm{Translate}(D_{all}) \\
T2 &= \mathrm{Select}_c(\mathrm{Search}_n(T, Q)) \text{ ※ (S1)〜(S2)} \\
Tr &= \mathrm{Extract}_l(\mathrm{Substract}(\mathrm{MergeGroup}_{cid}(\\
&\qquad \mathrm{Clustering}_k(T2)), T2), Q) \text{ ※ (S3)〜(S6)} \\
&\mathrm{Visualize}(T4)
\end{aligned}
$$

もし，ある程度分析パターンが特定できるのであれば，このように組み合わせた演算を再利用することができる．たとえば，日々変わる情報ソースに対して，ユーザの所望の分析パターンを少しずつ変えながら，毎日実行するといった使用方法も可能である．このように，グラフ表現モデルでは，単なる検索だけでなく，分類，特定部分の抽出，比較といったテキスト操作を柔軟に結び付けた分析を行うことができる．

グラフ表現モデルの分析式の記述能力を示すために，上記とは別の疑似適合フィードバック（pseudo relevance feedback）の例を示す．疑似適合フィードバックは，ある初期検索式で検索を行い，検索結果の上位 k 件に含まれる単語集合およびその重みを自動的に初期検索式に追加して再検索を行う手法であり，検索精度の向上に大きな効果がある[31]．疑似適合フィードバックは，従来，検索システムの外側で，検索式の再構成機能として個別に実現されていたが，グラフ表現モデルでは，以下の分析式で実現できる．

☆ InterOffice は，Oracle 製品であり，Informix と DB2 は，IBM 製品である．

$$Q1 = \mathrm{Translate}(\{\ \text{質問文}\ \})$$
$$T = \mathrm{Translate}(D_{all})$$
$$Q2 = \mathrm{Add}(Q1, \mathrm{Multiple}_{\alpha}(\mathrm{Search}_{k}(T, Q1)))$$
$$Tr = \mathrm{Search}_{n}(T, Q2)$$

ここで，初期検索式に特に関連の強い単語のみを再検索に利用したいという場合には，Extract を利用して，上記の Q2 の代わりに，

$$Q2' = \mathrm{Add}(Q1, \mathrm{Multiple}_{\alpha}(\mathrm{Extract}_{l}(\mathrm{Search}_{k}(T, Q1)), Q1))$$

と記述すればよい．このように柔軟な検索式の再構成もグラフ表現代数の記述範囲である．

6.1.2 テキスト表現モデルのカバー範囲

6.1.1 項の例で示したように，テキスト表現モデルでは，検索だけでなく，分類や特定部分の抽出といった様々なテキスト操作を分析式の中に記述できる．新しいテキスト操作が必要になった場合でも，これを演算として定義すれば容易にモデル内に取り込めるため拡張性がある．また，テキスト操作と演算がほぼ 1 対 1 で対応しているため分析式はシンプルで直感的な記述となる．さらに，アルゴリズムの変更等も演算内部に閉じたものとなり，システム全体にその影響が及ばない．

しかしながら，あらゆるテキスト操作に対応する演算が定義できるわけではない．「テキスト表現は，1 つのテキストまたはテキスト集合の内容の写像であり，演算は，このような写像された内容を対象とするもの」でないといけない．たとえば，情報抽出（information extraction）では，あるイベントが起こった日時や場所等の決められた項目情報を抽出するが，文書をテキスト表現に変換してしまった後では，このような項目情報の抽出は不可能である．当然，文書変換時に文書内で必要な項目情報をすべて抽出し，文書の属性としてあらかじめ格納しておく方法も考えられる．ただし，複数の項目情報間の関係を保持するためには，本稿で対象外とした属性の扱いを工夫する必要がある．

また，テキスト表現モデルでは，テキスト操作と演算がほぼ 1 対 1 の対応であるため，1 つの演算の中に複雑なアルゴリズムを含むことになる．さらに，演算内部で属性値の書き換えも行う．そのため，理論的な扱いが難しく，RDM のリレーショナル論理[25] に対応するような論理体系を持つことは困難であると考えている．

6.2 主題グラフの評価

我々は主要な対象文書として Web ページを想定している．そのため，3.2 節で定めた採用基準のうち (a) 文書変換可能性に関しては，形式の異なる様々な文書に対して文書変換が定義できる必要がある．ここで，主題グラフは，その作成方法から明らかなように，タームベクトルと同様に，この基準を満たす．そこで，比較対象としてタームベクトルを用い，(b) 演算のタスク適用性，(c) 可読性について比較評価する．また，計算量に関する検証を行う．

6.2.1 (b) 演算のタスク適用性

有効なテキスト操作に対応する 3 つの演算のうち，Search と Clustering は，タームベクトルでも定義可能である．ただし，タームベクトルでは，「同じ単語が含まれる文書どうしに高い類似度を与える」のに対して，主題グラフでは，「同じ単語が含まれかつそれらが同じように関連している文書どうしに高い類似度を与える」ことができる．このような主題グラフどうしの類似度判定の精度は，前述したようにタームベクトルに比べて高い[5]．また，ユーザが，単語間の関連を直接利用した検索を行いたい場合は，単語と単語をリンクさせた質問グラフを直接図 5 のユーザインタフェース上から指定できる．このように主題グラフでは，関連を利用した精度の高い Search や Clustering を実現できる．

残る Extract は，関連度を用いた主題グラフ特有の演算である．そこで，ここでは，Extract によって，重要であり，かつ入力された条件に関係の深い単語集合が取得できるかどうかを評価する．検索質問文から作成された主題グラフを条件グラフとして，近接度順に抽出される単語リストと，文書内での単語の重要度順に抽出される単語リストとを比較する．

定量的に，抽出された単語リストの適切さを測定するため，検索タスクを想定した可視化精度評価を行う．この評価では，質問文に対応する検索結果の各文書を可視化し，可視化結果（ここでは単語リスト）のみを被験者が見て，質問文と可視化された文書が relevant（関連あり）であるかどうかを判断する．ここで，各文書に，あらかじめ各質問文に対する relevance judgement（関連性の判断）がついていれば，これらを比較することによって，精度を測定できる．つまり，検索質問という条件に対して，適切な単語が抽出されていれば，ユーザは関連性判断が正確に行え，精度が高くなる．具体的には，以下の尺度[32] で精度を測定する．

$$\text{再現率} = \frac{|R \bigcap S|}{|R|} \tag{16}$$

$$\text{適合率} = \frac{|R \bigcap S|}{|S|} \tag{17}$$

$$\text{F 値} = \frac{2 \times recall \times precision}{recall + precision} \tag{18}$$

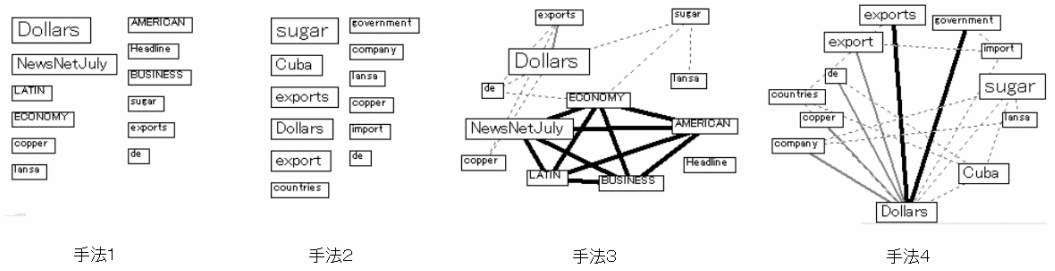

図 7 4 つの手法による結果の可視化
Fig. 7 Visualizing results by four methods.

表 2 検索タスクに基づく可視化精度の評価結果
Table 2 The results of searching task-based evaluation of visualization.

	再現率				適合率				F 値			
ユーザ	手法 1	手法 2	手法 3	手法 4	手法 1	手法 2	手法 3	手法 4	手法 1	手法 2	手法 3	手法 4
ユーザ 1	0.550	0.625	0.550	0.600	0.557	0.630	0.657	0.510	0.553	0.627	0.599	0.551
ユーザ 2	0.625	0.900	0.725	0.750	0.670	0.625	0.577	0.485	0.647	0.738	0.642	0.589
ユーザ 3	0.550	0.875	0.650	0.725	0.490	0.590	0.603	0.706	0.518	0.706	0.625	0.716
ユーザ 4	0.125	0.325	0.225	0.250	0.233	0.383	0.250	0.500	0.163	0.352	0.237	0.333
平均	0.462	0.681	0.537	0.581	0.487	0.557	0.521	0.550	**0.470**	**0.605**	**0.525**	**0.547**

R はユーザに提示された文書の中で relevant な文書の集合，S は提示された文書の中でユーザが relevant と判断した文書集合である．

実験には，WT2G を用いた．TREC サイトでは，WT2G に対する 50 個の質問文，各質問文に対する relevance judgement を提供している．英作文を職業とする 4 人のネイティブスピーカを被験者とした．各手法について，10 個の質問文（ID: 411-420），各質問文について 5 個の文書から抽出された単語リストを用いたので，各手法毎に 50 個の単語リストについて，relevant かどうかを判断した．フォントサイズや単語の個数（12 個）は，予備実験（ID: 401-410 を使用）の結果から適当に定めた．**図 7** に，'Cuba, sugar, exports' というタイトルを持つ質問文（ID: 414）に対する relevant な文書'WT04-B05-12' から得られた単語リストを示す．ここで，

手法 1： 重要度順–リスト表示
手法 2： 近接度順–リスト表示
である．

実験結果を**表 2** に示す．表 2 から手法 2 は，手法 1 よりも F 値がすべてのユーザで高い．つまり単語を重要度順に選択するよりも近接度順に選択した方が精度が高い．このように Extract は，条件を与えると，その条件に応じて，文書内容を表すうえで適切な部分グラフを抽出できている．

以上のように，タームベクトルと比べ Search，Clustering の精度が高く，また，Extract といった関連を用いた有効な演算を定義できる主題グラフは，(b) 演算のタスク適用性がタームベクトルと比べて高い．

6.2.2 (c) 可読性

可読性に関して，単語に加えて，単語間の関連を可視化する効果を検証する．主題グラフをバネモデルによって可視化する方法（グラフ表示）と，単語を重要度順に並べ，関連度を可視化しない方法（リスト表示）とを比較する．評価には，演算のタスク適用性の評価と同様に，可視化精度評価を行った．可視化結果を図 7，実験結果を表 2 に示す．ここで，

手法 3： 重要度順–グラフ表示
である．表 2 から手法 3 は，手法 1 と比べてほとんどすべてのユーザで，F 値が高い．この結果は，単語間の関連を可視化することによって，可読性を向上させることができることを示している．

ただし，ここで注意したいのは，手法 1 の重要度順で選択された単語リストは，検索質問に対する関連性の判断に適した並び順ではないということである．そこで，検索質問に対してより適切に単語が選択され，ソートされている近接度順でも同様にグラフ表示の可視化精度評価を行った．可視化結果を図 7，実験結果を表 2 に示す．ここで，

手法 4： 近接度順–グラフ表示
である．表 2 を見ると，手法 4 は，手法 2 よりもほとんどのユーザで F 値が低い．このように，リスト表示とグラフ表示の精度が重要度順と近接度順で逆の結果となったのは，人間が注目する単語の順番が影響したためと考えられる．そこで，関連性を判断する際に注目した単語の順番を，被験者に別途回答してもらっ

表 3 判定の際に注目した上位 5 個の単語
Table 3 The top five attention-grabbing terms for relevance judgment.

	ユーザ 1		ユーザ 2		ユーザ 5		ユーザ 6		ユーザ 7	
順位	手法 2	手法 4	手法 2	手法 4	手法 2	手法 4	手法 2	手法 4	手法 2	手法 4
1	Sugar	Dollars	Sugar	exports	Cuba	Dollars	Sugar	Cuba	Sugar	Cuba
2	Cuba	Cuba	Cuba	countries	Sugar	Cuba	Cuba	Sugar	Cuba	Sugar
3	exports	Sugar	exports	Cuba	exports	Sugar	Dollars	Dollars	exports	government
4	Dollars	exports	export	Sugar	Dollars	exports	export	exports	Dollars	exports
5	export	export	government	export	export	export	exports	government	export	Dollars

た．図 7 の手法 2 と手法 4 の関連性判断の際に注目した上位 5 個の単語の比較結果を**表 3** に示す☆．表 3 と図 7 を見比べると，手法 2 のリスト表示では，左上の単語からほぼ順番に注目しているのが分かる．これに対して，グラフ表示では，太いリンクでつながれている単語（Dollars，exports，government）の順位が高くなっている．

このように，ユーザは，可視化された関連自体を内容判断の手がかりに利用するというよりは，むしろ内容判断に重要な単語がどれなのかを選択する際に関連を利用している．つまり，「本来内容の把握に重要であるにもかかわらず，出現頻度からは重要度が低いとされた単語を，関連を可視化することによって，ユーザが見つけやすくなる」という効果がある．ただし，もともと重要な単語が正確に抽出され，ソートできている状況では，この効果はなく，むしろ逆に精度の低下につながることもある．このため，単語を重要度順で選択していている手法 3 は手法 1 に比べて精度が高く，近接度順に選択している手法 4 は手法 2 と比べて精度が低くなったと考えられる．

6.2.3 計 算 量

主題グラフは，タームベクトルに関連度ベクトルを加えたものであるため，関連度ベクトルが計算量に対して与える影響について検証する．

- **文書数の増加に対するコスト**
  式 (7) の近似式により，文書集合全体での集計が必要なのは，n_i，n_j だけであり，n_{ij} の集計は必要ない．ここで，n_i，n_j は文書数の増加に依存するが，タームベクトルでも同様に必要なため，文書数の増加に対する計算量は，タームベクトルと主題グラフで等しい．
- **各文書サイズの増加に対するコスト**
  共起単語ペアの個数は，最悪で文書に含まれる異なり語の個数 m のすべての組合せである $m(m-1)/2$ 個となる．ただ，一度も文書に出現しない共起単語ペアの関連度は 0 であるため，実際の関連度ベクトルの大きさは，実際にその文書で出現した共起ペアの個数にできる．このような共起ペアの個数は，m の 2 乗よりもかなり少ない[5]．WT2G の各文書に含まれる異なり語 m と共起ペアの個数の測定結果を両対数グラフに示す（**図 8**）．図 8 から得られる近似式は，

  $$\text{関連度ベクトルのサイズ} = 0.505\, m^{1.36}$$

  である．つまり，実際の関連度ベクトルのサイズは，タームベクトルのサイズの 1.36 乗程度であることを示している．

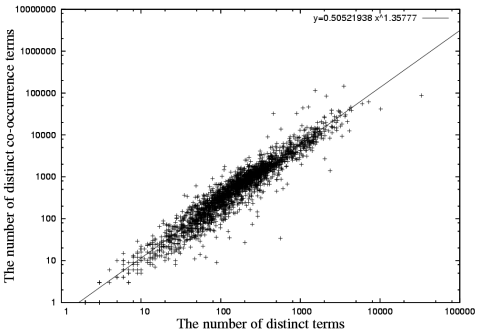

図 8 異なり語と異なり共起語の関係
Fig. 8 Distinct terms v.s. distinct co-occurrence terms.

以上により，現実の大規模な文書集合に対しても主題グラフは適用可能である．ただし，ここで述べたものは，基本的なベクトルサイズについての考察であり，実際の適用に際しては，それぞれの演算に適したインデクス構造を考慮にいれる必要がある．

7. む す び

本稿では，まず，文書情報統合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て実行する必要があることを述べ，これを実現するテキスト表現モデルを提案した．本モデルでは，テキスト表現を，内容表現と属性表現で構成することによって，内容に対する

☆ ここで，ユーザ 1，ユーザ 2 は，可視化精度評価と同一被験者であるが，残り 3 名は別の被験者である．本来同一被験者とするべきであるが，同一被験者の人材確保ができず別の被験者とした．また，可視化精度評価と注目単語の順番の判定は，約 1 年の期間が空いている．

テキスト操作や属性に対するテキスト操作を統合的に実現できる．また，動的な属性を含めることによって，検索や分類といった一見結合が難しいテキスト操作に対しても，入力を複数のテキスト表現リスト，出力を1つのテキスト表現リストとする演算が定義でき，これらの柔軟な組合せができる．テキスト表現モデルにおいては，どのようなテキスト表現を採用するのかが重要であり，その採用基準として，(a) 文書変換可能性，(b) 演算のタスク適用性，(c) 可読性を定めた．この採用基準に基づき，主題グラフを内容表現として採用し，主題グラフ上に，実際に様々な有効な演算が定義できることを示した．

テキスト表現モデル自体の評価では，グラフ表現上に定義された演算の組合せによって有効な文書情報統合タスクが記述できることを例によって示した．また，本モデルのカバー範囲として，分析式がシンプルで新規のテキスト操作に対する拡張性があるものの，「演算は文書そのものに対してではなく，文書変換後のテキスト表現に対してのみ定義可能である」，「理論的な扱いが難しい」という制限があることを述べた．今回は，特に内容表現を中心的に扱ったが属性表現の効果的な扱いが，定義できる演算の範囲をさらに広げられるのではないかと考えている．

主題グラフの評価では，タームベクトルと同様に，広い範囲の文書で文書変換が定義でき，計算量の面からも大規模文書に適用できることを述べた．演算のタスク適用性に関して，Search，Clustering の精度が高いことを述べ，関連度を用いることで初めて定義が可能となる Extract の効果を定量的に示した．また，可読性に関して，関連の可視化には，本来内容の把握に重要であるにもかかわらず出現頻度からは重要度が低いとされた単語を，ユーザが見つけやすくなるという効果があることを示した．この効果を利用した有効な可視化手法は今後の課題である．

参 考 文 献

1) Hearst, M.: Untangling text data mining, *Proc. 37th Annual Meeting of the Association for Computational Linguistics* (*ACL'99*), pp.3–10 (1999).
2) Broad, W.J.: Study finds public science is pillar of industry, *The New York Times* (1997).
3) Allan, J., et al.: Challenges in Information Retrieval and Language Modeling, *SIGIR FORUM*, Vol.37, No.1, pp.31–47 (2003).
4) Codd, E.F.: A relational model of data for large shared data banks, *Comm. ACM*, Vol.13, No.6, pp.377–387 (1970).
5) 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，Vol.43, No.SIG 2(TOD13), pp.94–107 (2002).
6) Frakes, W.B. and Baeza-Yates, R.: *Information Retrieval Data Structures & Algorithms*, Prentice Hall (1992).
7) 奥村　学，難波英嗣：テキスト自動要約に関する最近の話題，自然言語処理，Vol.9, No.4, pp.97–116 (2002).
8) Fuhr, N. and Rölleke, T.: A Probablistic Relational Algebra for the Integration of Information Retrieval and Database Systems, *ACM Trans. Inf. Syst.*, Vol.15, No.1, pp.32–66 (1997).
9) 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).
10) Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W. and Harshman, R.A.: Indexing by Latent Semantic Analysis, *J. Am. Soc. Inf. Sci.* (*JASIS*), Vol.41, No.6, pp.391–407 (1990).
11) 上田修功，斉藤和己：多重トピックテキストの確率モデル，情報処理，Vol.45, No.2, pp.184–190 (2004).
12) 笠原　要，稲子　望，加藤恒昭：単語の属性空間の表現方法，人工知能学会論文誌，Vol.17, No.5, pp.539–547 (2002).
13) Sowa, J.F.: Conceptual graphs for a database interface, *IBM Journal of Research and Development*, Vol.20, No.4, pp.336–357 (1976).
14) Liddy, E.D. and Myaeng, S.H.: DR-LINK: A System Update for TREC-2, *NIST Special Publication 500-215: The 2nd Text REtrieval Conference* (*TREC2*), pp.85–100 (1993).
15) Montes-y-Gómez, M., Gelbukh, A.F., López-López, A. and Baeza-Yates, R.: Flexible Comparison of Conceptual Graphs, *Proc. 12th International Conference on Database and Expert Systems Applications*, pp.102–111 (2001).
16) 大澤幸生，ネルス E. ベンソン，谷内田正彦：KeyGraph：語の共起グラフの分割・統合によるキーワード抽出，電子情報通信学会論文誌，Vol.J82-D-I, No.2, pp.391–400 (1999).
17) Ohsawa, Y., Soma, H., Matsuo, Y., Matsumura, N. and Usui, M.: Featuring Web Communities based on Word Co-occurrence Structure of Communications, *Proc. 11th International World Wide Web Conference* (2002).
18) 渡部　勇：ビジュアルテキストマイニング，人工知能学会誌特集：テキストマイニング，Vol.16, No.2, pp.226–232 (2001).

19) 藤井 敦：百科辞典としての WWW，人工知能学会誌特集：WWW 上の情報の知的アクセスのためのテキスト処理，Vol.19, No.3, pp.296–301 (2004).
20) Feldman, R.: Link Analysis: Current State of the Art, *Tutorial note of the 11th international Conference on Information and Knowledge Management* (*CIKM'02*) (2002).
21) Takano, A.: Associative information access using DualNAVI, *Proc.6th Natural Language Processing Pacific Rim Symposium* (*NLPRS2001*), pp.771–772 (2001).
22) Pfeifer, U., Fuhr, N. and Huynh, T.: Searching structured documents with the enhanced retrieval functionality of free WAIS-sf and SFgate, *Proc. 3rd International World-Wide Web conference on Technology, tools and applications*, pp.1027–1036, Elsevier North-Holland, Inc. (1995).
23) 難波英嗣：WWW 上のテキスト情報の知的統合，人工知能学会誌特集：WWW 上の情報の知的アクセスのためのテキスト処理，Vol.19, No.3, pp.311–316 (2004).
24) P.H. ウインストン，B.K.P. ホーン（著），白井良明，安部憲広（訳）：情報処理シリーズ 4—LISP，培風館 (1982).
25) 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).
26) 岩畑 清：言語と計算 4 アルゴリズムとデータ構造，岩波書店 (1989).
27) Tombros, A. and Sanderson, M.: Advantages of query biased summaries in information retrieval, *Proc. 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.2–10 (1998).
28) Eades, P.: A heuristic for graph drawing, *Congressus Numerantium*, Vol.42, pp.146–160 (1984).
29) Tomita, J., Ikeda, T., Kihara, T. and Satoh, T.: Knowledge discovery from Mixed-model XML documents, *SIGIR 2002 Workshop on XML and Information Retrieval*, pp.33–39 (2002).
30) Bailey, P., Craswell, N. and Hawking, D.: Engineering a multi-purpose test collection for Web retrieval experiments DRAFT (2001). http://es.cmis.csiro.au/TRECWeb/
31) 岸田和明，岩山 真，江口浩二：検索実験の方法と実際：NTCIR ワークショップでの試み，*NTCIR-3 Workshop Lecutre note* (2002).
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).

(平成 16 年 9 月 20 日受付)
(平成 17 年 1 月 13 日採録)

（担当編集委員 國島 丈生）

富田 準二（正会員）

日本電信電話株式会社サイバーソリューション研究所所属．1997 年慶應義塾大学大学院理工学研究科計算機科学専攻修士課程修了後，日本電信電話株式会社に入社．2005 年 1 月より，米国ワシントン大学客員研究員（Visiting Scholar）．情報検索，自然言語処理，情報抽出，XML 関連の研究開発に従事．日本ソフトウェア科学会会員．

石井 恵（正会員）

日本電信電話株式会社サイバーソリューション研究所所属．1989 年慶應義塾大学理工学部数理科学卒業．同年日本電信電話株式会社に入社．以来，知識獲得，情報検索関連の研究開発に従事．工学博士．

中渡瀬秀一（正会員）

日本電信電話株式会社サイバーソリューション研究所所属．1992 年神戸大学大学院工学研究科修士課程修了後，日本電信電話株式会社に入社．以来，情報検索，自然言語処理の研究開発に従事．

片岡 良治（正会員）

日本電信電話株式会社サイバーソリューション研究所所属．1987 年千葉大学大学院電子工学専攻修士課程修了後，日本電信電話株式会社に入社．以来，トランザクションの並行処理制御方式の研究，マルチメディア情報システムの研究，ポータルサービスシステムの研究開発に従事．電子情報通信学会会員．