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

富田準二[†] 竹野浩[††] 菊井玄一郎[†††]
林良彦[†] 池田哲夫[†]

文，文書，文書集合間の類似性を判定することは，テキスト検索，情報フィルタリング，文書のカテゴリ分け，文書クラスタリング等の要素技術である．我々は，このような文書間の類似性判定のための新しいモデルとして，グラフモデルを提案する．グラフモデルは，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によって文書の主題を表現し，これらグラフ間の類似度計算によって文書間の類似性判定を行うものである．本稿では，まず，グラフモデルの基本概念とその実現方法について述べる．次に，グラフモデルを実際にテキスト検索システムに適用し評価を行う．その評価結果から，特に 1 つの文書内に複数の主題を持つ文書（Web ページ等）や，文書集合（ユーザプロファイル，分類カテゴリの特徴，クラスタ等）を対象とした場合に，グラフモデルがベクトル空間モデルよりも精度の高い類似性判定ができることを示す．さらに，関連度のインデクスを前処理で作成することによって，リアルタイムの類似性判定が必要となるテキスト検索システム等にも，本手法の適用が可能であることを示す．

Proposal of Graph-based Text Representation Model and Its Evaluation in Text Retrieval System

Junji Tomita,[†] Hiroshi Takeno,[††] Genichiro Kikui,[†††]
Yoshihiko Hayashi[†] and Tetsuo Ikeda[†]

Calculating the similarity between sentences, documents, or sets of documents precisely is a very important technique for text retrieval, information filtering, document classification, document clustering and so on. We propose a new model to calculate this similarity; we call it Graph-based text representation model (Graph model). In the Graph model, each document is translated into a subject graph that treats the significance of each word as its node's weight and the significance of each term-term association as its link's weight. The similarity between documents is calculated by the similarity between these graphs. This paper describes the concept and the realization of the Graph model, and evaluates it in a text retrieval system. Through the evaluation, we show that the Graph model is more effective than the Vector space model in calculating the similarity between documents, especially when each document has many topics such as Web pages, or sets of documents such as user profiles, category profiles and clusters. In addition, the similarity calculation is fast enough for text retrieval systems, given the situation where the index of the association is pre-compiled.

1. はじめに

コンピュータ，ネットワークの進歩によって，ユーザの取り扱わなければならない電子化文書の量が急速に拡大している．そのため，大量の電子化文書を効率良く取り扱う手法（システム）に対するニーズが高まっている．このような手法には，テキスト検索，情報フィルタリング，文書のカテゴリ分け，文書クラスタリング等があるが，これらを支える要素技術として，文，文書，文書集合間の類似性判定がある．たとえば，テキスト検索ではユーザの入力した検索キーと検索対象文書との間でこのような類似性判定が行われる．同様に，内容に基づく情報フィルタリングでは入力文書とユーザプロファイル間，文書のカテゴリ分けでは分類対象文書とそれぞれの分類カテゴリの特徴間，文書クラスタリングでは分類対象文書やクラスタ間で，このような類似性判定が行われる[1)]．

文書間の類似性判定を行うためには，通常，文書の

† 日本電信電話株式会社 NTT サイバースペース研究所
NTT Cyber Space Laboratories, NTT Corporation
†† 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 Solution Laboratories, NTT Corporation
††† ATR 音声言語通信研究所
ATR Spoken Language Translation Research Laboratories

主題をある表現に写像し，この表現間に何らかの尺度を導入し類似度を計算する．このような文書の表現として代表的なものにタームベクトルがある．タームベクトルは文書から単語を抽出し，なんらかの方法で単語の重要度を計算し，単語を次元とするベクトルで文書の主題を表現する．そして，これらのタームベクトル間の内積やコサインといった尺度によって，文書間の類似性判定を行っている．このように文書をタームベクトルで表現しタームベクトル間の類似度によって類似性判定を行う手法はベクトル空間モデルと呼ばれている[2]．

ベクトル空間モデルは，処理が単純で高速化が容易であり，また，ベクトルの合成演算（重心や和等）を用いれば，複数文書を 1 つのタームベクトルで表すこともできる．たとえば，文書クラスタリングでは，複数の文書で構成されるクラスタの特徴を，そのクラスタに含まれるそれぞれの文書のタームベクトルを合成した 1 つのベクトルで表している．このように単一の文書だけでなく，文書集合も同様の表現で扱うことができるため，多くの場面で利用されている．しかし，その反面で，ベクトル空間モデルでは，それぞれの文書の主題を，文書に含まれている互いに独立な単語の集合で表現し，“類似している文書とは，同じ単語を使用している文書である” と仮定している．そのため，

問題（1） 単語情報以外の特徴，たとえば，単語間の係り受け関係，文，段落等の特徴を利用できない，

問題（2） 1 つの文書に複数の主題を持つ文書（以下複数主題文書）や文書集合間の類似性判定の精度が低い，

といった問題がある．問題（2）については 2.1 節で詳しく述べるが，複数主題文書を対象としたテキスト検索の問題としてすでに指摘されている[3),4)]．

これとは対照的に，文書に含まれる単語間の意味的な関係に着目して文書を表現するという手法が提案されている．文献 5) では，文の意味を解析するために，単語をノードとし，行為者や対象物等といった単語間の意味的な関係をリンクとしたグラフを用いている．しかし，文書からこのようなグラフを完全自動で生成することは困難であり，文書間の類似性判定に利用するのは現実的ではない．また，文献 6) では，ユーザの検索要求の明確化を目的とし，ユーザの入力した検索要求文書から単語の共起関係を用いて，単語をノード，単語間の関連の強さをリンクの重みとしたグラフを作成している．しかし，文書間の類似性を判定する際に重要となる単語の重要度は考慮されていない．

本稿では，新しい文書間の類似性判定のモデルとしてグラフモデルを提案する．グラフモデルは，まず，それぞれの文書の主題を，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グラフによって表現する．そして，これらグラフ間の類似度という尺度を用いて文書間の類似性判定を行う．そのため，単語の重要度だけでなく単語間の関連度も考慮しながら文書間の類似性判定を行うことができる．実際にグラフモデルをテキスト検索システムに適用し検索精度およびシステム効率の両面での評価を行う．これらの評価を通して，グラフモデルの有効性を検証する．ここで，テキスト検索システムをグラフモデルの適用先として選んだ理由は，実際の利用場面が多いことに加えて，様々なワークショップやテストコレクションがあり，評価手法が確立していることにある[7)~10)]．

以下，2 章では，グラフモデルの基本概念とその実現方法について，3，4 章では，グラフモデルを実際に適用したテキスト検索システムとその評価について，5 章では，その評価結果から得られたグラフモデルの有効性について，6 章では，まとめと今後の課題について述べる．

2. グラフモデル

本章では，まず，グラフモデルの基本概念について，ベクトル空間モデルと対比しながら述べる．次にグラフモデルを実際に実現するために必要となる単語の重要度，単語間の関連度，グラフ間類似度の計算方法について述べる．

2.1 グラフモデルの基本概念

以下の 2 つの文書の類似性判定について考える．

文書 1 “ネットワークから情報を収集するロボット” に関する論文

文書 2 “ネットワーク管理” と “産業用ロボットの開発” 等の研究を行っている研究機関の紹介文書

ベクトル空間モデルでは，まず，それぞれの文書から単語を抽出し以下のようなタームベクトルを作成する．

	ネットワーク	ロボット	情報	···
文書 1	(0.9	0.8	0.6	···)
文書 2	(0.8	0.7	0	···)

ここで，タームベクトルの要素は，それぞれの文書に含まれる単語の重要度であり，単語の出現頻度情報等から計算される．次に，これらのベクトルの内積計算によって類似性の判定を行う．そのため，このように同じ単語を含んでいるタームベクトル間の類似度は大きくなる．しかしながら，文書 1 と文書 2 は違う主題の文書であることは明らかであり，このような文書間

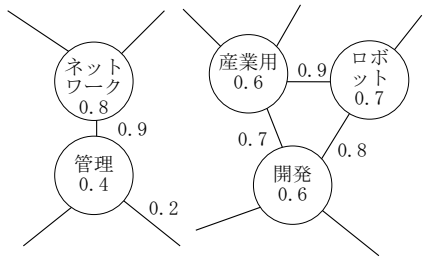

図 1 主題グラフ
Fig. 1 Subject graph.

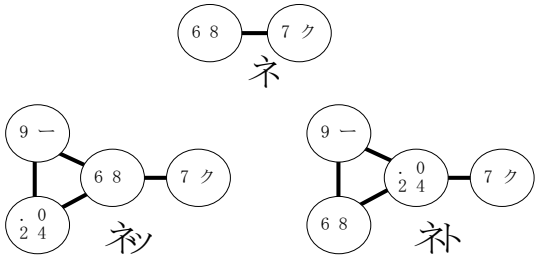

図 2 主題グラフ間の類似度計算
Fig. 2 Calculating similarity between subject graphs.

に高い類似性を与えることには問題がある．このように，ベクトル空間モデルでは，複数主題文書を対象とした類似性判定の精度は低い．同様の理由で，複数の文書をまとめた文書集合を対象とした類似性判定の精度も低い．

そこで，“類似している文書とは，単に同じ単語を使用しているだけではなく，それらの単語どうしが同じように，ある種の関連を持っている文書である” と仮定する．たとえば，ある種の関連を “同一文内での共起関係” にとると，文書 1 と文書 2 の類似性は小さい値となる．なぜなら文書 1 においては “ネットワーク” と “ロボット” は関連しているのに対して，文書 2 においては “ネットワーク” と “ロボット” は関連していないからである．このように単語間の関連という情報を用いることによって類似性判定の精度を向上させることができる．以下，この仮定に基づいた効率的な主題の表現方法とその表現上での類似度計算の方法について述べる．

2.1.1 グラフによる文書の表現（主題グラフ）

前節で述べたベクトル空間モデルの問題を解決することを狙いとし，単語の重要さと単語間の関連の強さの両方を考慮できるように，本手法では，

ノードの重み 単語の重要度
リンクの重み 単語間の関連度

としたグラフによって文書の主題を表現する．以下，このようなグラフを主題グラフと呼ぶ．例として，文書 2 から作成される主題グラフを**図 1** に示す．

主題グラフは，リンクに意味的な関係をラベル付けした意味ネットワーク[11]等と比べるとその表現能力は低い．しかしながら，意味ネットワーク等ではリンクへの意味的な関係の付与の完全自動化が困難であるのに対して，文書から単語の重要度や単語間の関連度を自動的に計算する手法はすでにいくつか提案されているため，主題グラフは少ないコストで自動生成できる．また，類似度計算のアルゴリズムも意味ネットワークと比べて単純にすることができると考えられる．さらに，タームベクトルの合成演算のアナロジを用いて，主題グラフの合成演算を定義すれば，複数の文書から 1 つの表現を生成することもできる．たとえば，単語の重要度の和を計算することに加えて，単語間の関連度の和を計算すれば，複数の主題グラフから合成した 1 つの主題グラフを生成することができる．

2.1.2 主題グラフを用いた類似度計算

以下の主題を持つ文書間の類似性判定を考える．

文書 d 検索効率
文書 $d1$ 情報検索システムの検索効率
文書 $d2$ 情報検索システムのシステム効率

これらの文書から作成される主題グラフを**図 2** に示す．ここでは，簡単のため，すべての単語の重要度，単語間の関連度は等しい値とする．ベクトル空間モデルにおける類似度計算では，$d1$ と $d2$ はまったく同じ単語を使用しているので，$sim(d, d1) = sim(d, d2)$ となる．ここで，$sim(x, y)$ は文書 x と文書 y の類似度を表す．

グラフモデルの類似度計算において，たとえば，同じ単語間に直接のリンクがあるものどうしに，より高い類似度を与えることとする．d 内では，“検索” と “効率” の間に直接のリンクがある．これに対し，$d1$ 内には “検索” と “効率” の間に直接のリンクがあるが，$d2$ 内にはこのようなリンクはない．したがって，$sim(d, d1) > sim(d, d2)$ とすることができる．すなわち，主題グラフを用いれば，ベクトル空間モデルにはできないような，より精度の高い類似度計算尺度を定義できる．このようにグラフモデルは，文書を主題グラフで表現し，これら主題グラフ間の類似度計算によって文書間の類似性判定を行う．

2.2 グラフモデルの実現方法

グラフモデルを実現するためには，単語の重要度，単語間の関連度，主題グラフ間の類似度の具体的な計算方法を規定する必要がある．ベクトル空間モデルの基本となる概念と，単語の重要度の計算方法は独立したものであるのと同様に，グラフモデルの基本概念

とこれらの計算方法も独立したものである．ベクトル空間モデルにおける単語の重要度計算に関しては，数多くの研究が行われているが，その代表的な手法にtf*idf法があり，その有効性が広く認識されている☆．そこで，ここでは，単語の重要度，単語間の関連度の計算方法の1つとして，tf*idf法とそのアナロジを用いた方法を用いる．以下，単語の重要度，単語間の関連度，主題グラフ間の類似度の計算方法の詳細について述べる．

2.2.1 単語の重要度の計算方法

単語の重要度計算にtf*idf法を用いる場合，出現頻度の正規化の方法によって，その類似度計算の精度が異なってくる．ここでは，文献12)で実験的に良いとされるものの類似手法であり，検索システムfreeWAIS-sf[13]で採用されている以下の正規化を行い単語の重要度を計算する．まず，単語 x の逆文書頻度 idf_x を

$$idf_x = \log(N/n_x) \tag{1}$$

によって求める．ただし，N は対象とする文書の総数，n_x は単語 x が出現する文書の個数を表す．

次に単語 x の文書 d 内での出現頻度 tf_{xd} を計算する．

$$ptf_{xd} = 0.5 + 0.5 * \frac{oc_{xd}}{\max_{i \in W} oc_{id}} \tag{2}$$

$$tf_{xd} = \frac{ptf_{xd}}{\sqrt{\sum_{i \in W} ptf_{id}^2}} \tag{3}$$

ここで，oc_{xd} は文書 d 内の単語 x の出現回数，W は文書 d 内の相異なる単語（以下異なり語）の集合，$\max_{i \in W} oc_{id}$ は文書 d 内の単語の最大出現回数を表す．

このようにして求めた，idf_x と tf_{xd} から，単語 x の文書 d 内での重要度 $term\_weight_{xd}$ を，

$$term\_weight_{xd} = tf_{xd} * idf_x \tag{4}$$

によって求める．

2.2.2 単語間の関連度の計算方法

単語間の関連度の計算方法として，ここでは，相互情報量を用いる．相互情報量は2つの事象の間の関連の強さを表す尺度でありテキスト処理の分野では定型表現（イディオム）の抽出等で広く利用されている[14]．以下に，相互情報量を用いる2つの方法を提案する．

2.2.2.1 共起関係に基づく方法

同一文中で共起する単語間にはなんらかの関連がある．そこで，このような共起関係を用いて単語 x と単語 y の文書 d 内での関連度 $terms\_association_{xyd}$ を計算する方法を示す．

まず，句読点やHTMLタグ等を利用して文書を文に分割し，それぞれの文を形態素解析することによって単語を抽出する．

次に，文を基準とした単語の出現確率を求める．この際，文の長さはまちまちなので，長さが短い文に出現する単語程，出現確率が高くなるように，以下のように文の長さで正規化することによって，単語 x の文書 d 内での出現確率 ocs_{xd} を計算する．

$$pocs_{xd} = \sum_{k \in L} \frac{1}{c_k} * \zeta_k(x) \tag{5}$$

$$ocs_{xd} = \frac{pocs_{xd}}{\sum_{i \in W} pocs_{id}} \tag{6}$$

ここで，L は文書内のすべての文からなる集合，c_k は文 k に含まれる単語数を表し，$\zeta_k(x)$ は単語 x が文 k に出現すれば1，しなければ0である．ここで，式(6)の分母に $\sum_{i \in W} pocs_{id}$ を持つのは，確率の性質 $\sum_{i \in W} ocs_{id} = 1$ を満たすように正規化するためである．

次に，同様の方法で単語 x と単語 y の文書 d 内での共起確率 $cocs_{xyd}$ を計算する．

$$pcocs_{xyd} = \sum_{k \in L} \frac{1}{c_k} * \xi_k(xy) \tag{7}$$

$$cocs_{xyd} = \frac{pcocs_{xyd}}{\sum_{i,j \in W} pcocs_{ijd}} \tag{8}$$

ここで，$\xi_k(xy)$ は，単語 x と単語 y が文 k 内で共起すれば1，しなければ0である．

これらの出現確率 ocs_{xd} と共起確率 $cocs_{xyd}$ を用いて単語 x と単語 y の文書 d 内での相互情報量 I_{xyd} を計算する．

$$I_{xyd} = \log(\frac{cocs_{xyd}}{ocs_{xd} ocs_{yd}}) \tag{9}$$

2.2.1項で述べたように，単語の重要度をtf*idf法を用いて計算する場合，単純に出現回数を用いるのではなく，いくつかの正規化処理を行っている．そこで，ここでは I_{xyd} を以下のように正規化する．

$$ptr_{xyd} = 0.5 + 0.5 * \frac{I_{xyd}}{\max_{i,j \in W} I_{ijd}} \tag{10}$$

$$tr_{xyd} = \frac{ptr_{xyd}}{\sqrt{\sum_{i,j \in W} ptr_{ijd}^2}} \tag{11}$$

ここで $max_{i,j \in W} I_{ijd}$ は，文書 d 内での任意の2単語間の相互情報量の最大値である．

tf*idf法では，多くの文書に出現する単語は重要でないと考え，このような単語の重要度を小さい値としている．このアナロジから，多くの文書に出現する単

☆ tf*idf法は，単語 x の文書 d 内での重要度を，文書 d 内での単語 x の出現頻度（tf_{xd} 値）と，対象とする文書集合全体の中で単語 x が出現する文書数の逆数（idf_x 値）との積によって計算する．

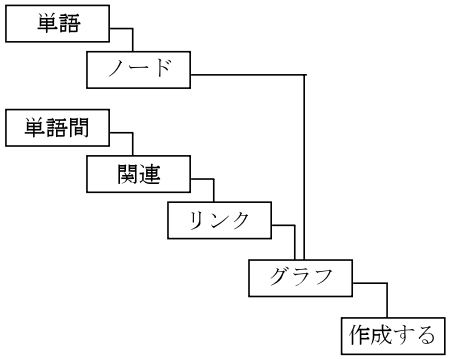

図 3 係り受け木
Fig. 3 Dependency tree.

語との関連度が小さい値となるように，式 (1) で求めた idf 値の和を tr_{xyd} に乗じる☆．単語 x と y の文書 d 内での関連度 $terms\_association_{xyd}$ は

$$terms\_association_{xyd} = tr_{xyd} * (idf_x + idf_y) \quad (12)$$

によって求める．このように，共起関係を用いて式 (12) によって関連度を計算する手法を，以下，共起に基づく手法と呼ぶ．

2.2.2.2 係り受け関係に基づく方法

同一文内で共起するが直接の係り受け関係にない単語間と比べて，直接の係り受け関係にある単語間の関連はより強いと考えられる．このような係り受け関係を用いれば，より正確に単語間の関連度を計算することができる．そこで，係り受け関係を用いて単語間の関連度を計算する方法を以下に示す．

単語 x と単語 y の文書 d 内での係り受け関係の強さ dp_{xyd} を以下の手順で求める．

(1) 文書 d のそれぞれの文 l を係り受け解析し，**図 3** のような係り受け木を生成する．
(2) 単語 x と y 間の係り受け木 l 上での最短パス長 sp_{xyl} を求める．たとえば，図 3 上で “単語” と “リンク” の最短パス長は “ノード”，“グラフ” を経由した 3 となる．
(3) それぞれの係り受け木上での最短パス長の逆数の総和によって係り受けの強さを表す値 dp_{xyd} を計算する．

$$dp_{xyd} = \sum_{k \in L} \frac{1}{sp_{xyk}} \quad (13)$$

ここで，L は文書内のすべての文からなる集合を表す．

dp_{xyd} を式 (7) の共起回数 $pcocs_{xyd}$ の代わりに用いることによって，単語 x と y 間の関連度を求める．このように，係り受け関係を用いて式 (12) によって関連度を計算する手法を，以下，係り受けに基づく手法と呼ぶ．

係り受け関係を用いると，単純に共起関係を用いた場合と比べて正確に関連度が計算できると考えられるが，計算コストが大きいという欠点がある．どちらを選択するかは，類似度計算の精度とコストを考え合わせて決める必要がある．

2.2.3 主題グラフ間の類似度の計算方法

主題グラフ間の類似度を計算するには，グラフ構造を考慮したマッチングを行うことが望ましい．しかし，

(1) グラフ構造を考慮したマッチングは非常に大きな計算量を必要とする，
(2) 単語間の関連度を文内共起等を基に計算しているので，主題グラフは完全グラフを結合したようなグラフとなる．そのため，構造を考慮したとしてもその効果は小さいと考えられる，

等の理由から構造までは考慮せず，単純に，“ノードの重みを要素として持つベクトルどうしの内積 f_v” と，“リンクの重みを要素として持つベクトルどうしの内積 f_r” の線形結合によって主題グラフ間の類似度を計算する．具体的な手順は以下のとおりである，

文書 1 の主題グラフ d_1 と文書 2 の主題グラフ d_2 のノードの重みをそれぞれ以下のベクトルで表す．

$$\boldsymbol{v}_{d_1} = (v_{1d_1}, \ldots, v_{id_1}, \ldots, v_{md_1}) \quad (14)$$

$$\boldsymbol{v}_{d_2} = (v_{1d_2}, \ldots, v_{id_2}, \ldots, v_{md_2}) \quad (15)$$

ここで，m は文書に含まれる異なり語の個数であり，v_{id_z} は，それぞれ文書 z 内での単語 i の重要度に対応し，式 (4) によって計算する．ここで，単語 i が文書 z 内に出現しない場合，$v_{id_z} = 0$ である．これらのベクトルの内積 f_v

$$f_v = \sum_{i \leq m} v_{id_1} * v_{id_2} \quad (16)$$

を求める．

次に，d_1 と d_2 のリンクの重みをそれぞれ以下のベクトルで表す．

$$\boldsymbol{r}_{d_1} = (r_{12d_1}, \ldots, r_{ijd_1}, \ldots, r_{m-1md_1}) \quad (17)$$

$$\boldsymbol{r}_{d_2} = (r_{12d_2}, \ldots, r_{ijd_2}, \ldots, r_{m-1md_2}) \quad (18)$$

ここで，r_{ijd_z} $(i < j)$ は，文書 z 内での単語 i と単語 j の関連度に対応し，式 (12) によって計算する．これらのベクトルの内積 f_r

☆ 多くの文書に出現する単語との関連度を小さくするという性質を満たす式は多数ある．しかし，それらすべてを調べることは困難であるので，ここでは，idf 値を用いた簡単な式として，和，積，相加平均，相乗平均の 4 つについて予備実験を行った．その結果，ほどんど検索精度に差が見られなかったために最も計算の容易な和を採用した．

$$f_r = \sum_{i,j \leq m} r_{ijd_1} * r_{ijd_2} \quad (19)$$

を求める．

このようにして求めた，f_v と f_r から主題グラフ間の類似度を以下のように計算する．

$$\text{グラフ間類似度} = p * f_v + (1-p) * f_r \quad (20)$$

ここで，$0 \leq p \leq 1$ である．この方法では，最悪の場合でも $O(m^2)$ でグラフ間類似度を計算できる．ここで m は文書に含まれる異なり語の個数を表す．ただし，4.2.1 項の評価で示すように，実際の文書で共起する単語のペアの個数は異なり語のすべての組合せである $m(m-1)/2$ 個よりもはるかに少ない．したがって，リンクの重みの格納方法を工夫すれば，実際は $O(m^2)$ よりもはるかに小さい計算量でグラフ間類似度を計算できる．また，式 (20) の係数 p を調整することによって，単語の重要度をより重視した類似度計算とするのか，単語間の関連度をより重視したものとするのかを調整できる．このように 1 つの係数 p だけを持たせたシンプルな式にすることによって，関連度を考慮することによる効果を明らかにすることができる．

3. グラフモデルに基づくテキスト検索システム

グラフモデルを実際に，テキスト検索システムに適用し，その有効性を検証する．従来のテキスト検索システムは，書誌情報等の小規模で比較的まとまった 1 つの内容で記述された文書セットを対象としてきた．しかし，近年，1 つの文書の中に複数の独立した主題を持つ複数主題文書を検索対象とする場合が多くなってきている．たとえば，Web ページには，新聞記事を複数まとめて 1 つのページにしたもの，日記，リンク集等がある．また，あるシステムの紹介文書等では，システム仕様，適用領域，性能といったように，それぞれの章は独立した主題を持っている．

また，自然言語文による検索，類似文書検索，関連性フィードバックのように，単語列だけでなく，文，文書，文書集合をキーとした検索も行われている[15]．このような状況に対して従来のベクトル空間モデルに基づく検索システムでは，1 章で述べた問題（1），問題（2）が生じる．そこで，これらの問題に対してグラフモデルがどの程度有効に働くのかを検証する．

3.1 検索システムの構成

テキスト検索システムをインタラクティブなサービスとして見た場合，ユーザが検索キーを入力してから検索結果が出力されるまでの検索速度が重要である．

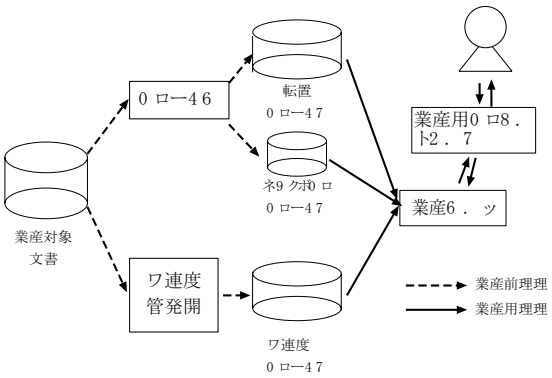

図 4 グラフモデルに基づくテキスト検索システム
Fig. 4 Text retrieval system based on Graph model.

グラフモデルの適用によって検索精度が良くなったとしても検索速度が極端に遅くなってしまってはサービスとしての利用価値が低い．グラフモデルで多くの計算時間が必要となるのは，大量の文書内の単語の重要度や単語間の関連度を計算する処理である．そこで，本検索システムは，あらかじめ重要度と関連度を計算し，インデクス化しておき，これらを用いて検索を行う構成とした（**図 4**）．

図 4 の転置インデクスは，以下のキーと値を持つテーブルであり，テキスト検索において広く利用されている．

転置インデクス

キー： 単語

値： その単語の出現する文書 ID と各文書内でのその単語の重要度の組の集合．

関連度インデクスは，以下のキーと値を持つテーブルである．

関連度インデクス

キー： 文書 ID

値： その文書内で共起する単語のペアと関連度の組の集合．以下，このような単語のペアを共起ペアと呼ぶ．

また，ヘッドラインインデクスは，文書 ID をキーとし，検索結果として表示するヘッドライン（タイトルやファイル名）を値として持つテーブルである．

本検索システムは，以下の検索前処理によって，あらかじめこれらのインデクスを作成する．そして，検索キーが与えられたときには，以下の検索時処理によって，これらインデクスを用いた検索を行う．

検索前処理

手順 1 インデクサは，以下の処理を行う．

(a) それぞれの検索対象文書から単語を抽出する．

(b) 式 (4) によって単語の重要度を計算する．
(c) 単語，文書 ID，単語の重要度を転置インデクスに格納する．
(d) それぞれの検索対象文書から，タイトル，ファイル名等を抽出し，文書 ID とともにヘッドラインインデクスに格納する．

手順 2　関連度計算機は，以下の処理を行う．
(a) それぞれ検索対象文書から文を抽出する．
(b) 文から単語を抽出するか，文を係り受け解析し式 (12) によって，それぞれの共起ペアの関連度を計算する．
(c) 文書 ID，共起ペア，関連度を関連度インデクスに格納する．

検索時処理

手順 3　検索用インタフェースは，以下の処理を行う．
(a) ユーザの入力した文や文書を解析する．
(b) これらの文や文書に含まれる各単語の重要度および各単語間の関連度を式 (4)，式 (12) を用いて計算し，重要度のベクトルと関連度のベクトルを作成する．
(c) これらのベクトルを検索キー主題グラフとして検索サーバに送信する．

手順 4　検索サーバは，以下の処理を行う．
(a) 検索用インタフェースから検索キー主題グラフを受信する．
(b) 検索キー主題グラフに含まれる単語をキーとして，転置インデクスにアクセスし検索結果の候補となる文書 ID を絞り込む．
(c) それぞれの文書 ID に対応する重要度のベクトルを作成する．
(d) それぞれの文書 ID をキーとして関連度インデクスにアクセスし，関連度のベクトルを作成する．ここで関連度のベクトルの要素数は，その文書内の共起ペアの個数に等しい．
(e) このようにして得られた重要度のベクトルと関連度のベクトルを検索対象文書主題グラフとする．
(f) 式 (20) を用いて検索キー主題グラフと，それぞれの検索対象文書主題グラフとの類似度を計算する．
(g) 類似度の降順に文書 ID を並べ，上位 n 件のヘッドラインをヘッドラインインデクスから取得し検索結果とする．
(h) 検索結果を検索用インタフェースへ送信する．

手順 5　検索用インタフェースは検索サーバから受信した検索結果をユーザに提示する．

このように，転置インデクスを用いることによって，検索時に処理の対象となる文書を，検索キー主題グラフ内の単語を 1 つ以上含む文書だけに限定することができる．以下，このようにして絞り込まれた文書をヒットした文書，その件数をヒット件数と呼ぶ．類似度計算処理は，これらヒットした文書についてのみ逐次的に行われる．そのため，検索時間は，検索対象文書セット全体の大きさには依存せず，実際にヒットしたそれぞれの文書にかかる類似度計算時間の総和になる．それぞれの文書にかかる類似度計算時間は，関連度インデクスの値の大きさにほぼ比例する．ここで，関連度インデクスの値は，前述したように，それぞれの文書の中で実際に共起する単語のペアとその関連度の組の集合である．そのため，類似度計算時間は，文書に実際に含まれる共起ペアの個数に依存することになる．

3.2 検索システムの実装

インデクサには freeWAIS-sf version 2.2.10[☆1]付属のものを利用した．検索サーバは freeWAIS-sf のインデクサが作成する転置インデクス，ヘッドラインインデクスを用いる．形態素解析器には Juman version 3.5[☆2]を利用し，名詞と動詞（原型に変形）だけを単語として用いた．関連度計算機で係り受け解析を行う場合には，係り受け解析器として京都大学で開発された knp version2.0b6[☆3] を利用した．knp は形態素列を入力とし係り受け解析を行い，解析結果の係り受け木を出力する．関連度インデクスには，gdbm version 1.7.3[☆4]を用いた．検索サーバ，関連度計算機，検索用インタフェースは C++で実装した．

4. テキスト検索システムの評価

3 章で述べたテキスト検索システムを，検索精度，システム効率の両面から評価を行いグラフモデルの有効性を検証した．

4.1 検索精度の評価

検索精度の評価に，情報検索システム評価用テストコレクション（BMIR-J2）[☆5]を利用した．BMIR-J2

☆1 ftp://ls6-ftp.cs.uni-dortmund.de/pub/src/freeWAIS-sf/
☆2 http://pine.kuee.kyoto-u.ac.jp/nl-resoure/juman.html
☆3 http://pine.kuee.kyoto-u.ac.jp/nl-resoure/knp.html
☆4 http://www.gnu.org/directory/gdbm.html
☆5 BMIR-J2 は（社）情報処理学会・データベースシステム研究会が，新情報処理開発機構との共同作業により，毎日新聞 CD-ROM'94 データ版を基に構築した情報検索システム評価用テス

表 1 2つの検索要求文集合
Table 1 Two QuerySets.

	検索要求文の個数	平均単語数
QuerySet1	13	2.77
QuerySet2	14	3.07

は，新聞記事 5,080 件（平均約 600 文字），検索要求文 50 件（平均 8.6 文字），それぞれの検索要求文に対する正解の集合からなる[10]．50 件の検索要求文は，検索時に必要となる機能によって 5 種類に分類されていて，必要に応じて使用するものを選択できるようになっている．今回は，グラフモデルの性質から以下のものを取り除いた残りを評価に用いた．

- 数値レンジ機能☆1および知識処理機能☆2を必要とする検索要求文
- 形態素解析の結果 1 単語となる検索要求文

次に，これらの検索要求文を，BMIR-J2 の検索要求文の ID が奇数のものと偶数のものに分けて 2 つのセットを作成した（**表 1**）．

QuerySet1，QuerySet2 のそれぞれの検索要求文を検索用インタフェースの入力として用いた．検索要求文から検索キー主題グラフを作成する際には，検索対象文書と同様に式 (4) と式 (12) を用いた．また，検索精度の評価用プログラムには trec_eval☆3 を利用した．

4.1.1 グラフモデルの有効性の検証とパラメータ p の決定（実験 1）

式 (20) の定数 p を $p = 0.0$（関連度だけを考慮）から $p = 1.0$（重要度だけを考慮）まで 0.1 きざみで変化させ，QuerySet1 と QuerySet2 を用いて検索を行った．そして，再現率，0.0，0.1，0.2，...，1.0 の 11 点における Interpolated Recall-Precision Averages☆4の平均（以下，11pt-ave）を求めた．

検索要求文，検索対象文書の単語間の関連度は共起に基づく手法によって計算した．ベクトル空間モデル（$p = 1.0$）の検索精度を基準としたときの p 値の変化に対する検索精度の向上率を**図 5** に示す．

図 5 から QuerySet1，QuerySet2 のどちらを用いても $p = 0.6$〜0.9 辺りで，$p = 1.0$ と比べて検索精度が向上している．この結果から単語の重要度と単語間の関連度の両方を考慮することによって，検索精度を向上させることができると考えられる．

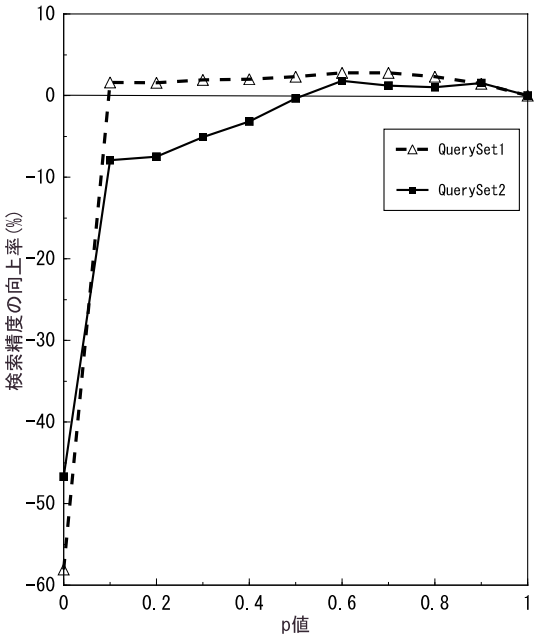

図 5 p を変化させたときの検索精度の変化
Fig. 5 11 point average of the interpolated recall-precision averages versus p.

以下の実験では，特定の評価データに依存した実験結果とならないように，p 値として QuerySet1 の検索精度を最大にする 0.7 を採用し，QuerySet2 を用いて評価を行うこととする．この条件での再現率–適合率の関係を**図 6** に示す．また，$p = 0.7$ での QuerySet1，QuerySet2 の Query ごとの検索精度を**図 7** に示す．

図 5 からは，11pt-ave では最大 2%程度の検索精度の向上が見られたものの，図 6 からは再現率によっては，グラフモデルとベクトル空間モデルの適合率が逆転しているところもある．また，図 7 からすべての Query に対して検索精度が向上しているのではなく，検索精度が低下しているものもある．そこで，全体の検索精度の向上がわずかであったこと，検索精度の低下する Query もあることに関して分析を行った．

BMIR-J2 では，それぞれの検索対象文書は，平均約 600 文字と比較的短い新聞記事であり，1 つの主題で記述された均質な文書と見なせる．このような均質な文書を対象とし，かつ検索要求文に含まれる単語数が平均 3.07 語と短かったためにグラフモデルの効果は僅かなものであったと考えられる．

検索精度が最も低下する Query は “教育産業”（ID114）であった（図 7）．この Query における不正解文書の 1 つは，毎日新聞の ID00442430 であり，

トコレクションである．CD-毎日新聞 94 版を使用．
☆1 正解の決定に数値の大小比較等が必要
☆2 正解の決定に世界知識を用いたキーワードの展開が必要
☆3 ftp://ftp.cs.cornell.edu/pub/smart/trec_eval.v3beta.shar
☆4 特定の再現率における，評価に用いたすべての Query の適合率の平均値．補完（Interpolated）とは，たとえば，再現率 0.1 の適合率の値を再現率 $\geq$0.1 における適合率の最大値で代用したものである．

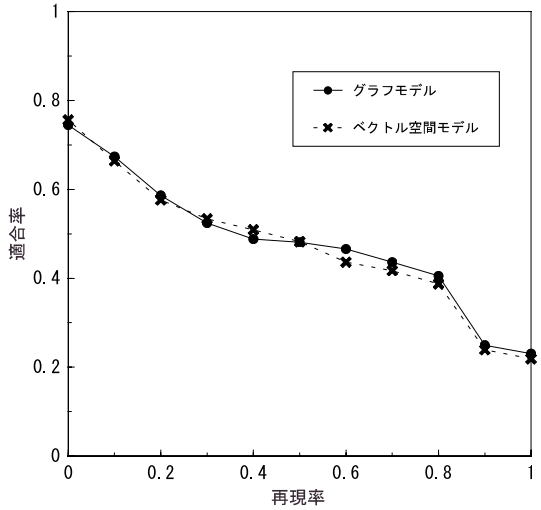

図 6 均質な文書に対するグラフモデルとベクトル空間モデルの比較
Fig. 6 Comparison of Graph model and the Vector space model when each document has only one topic.

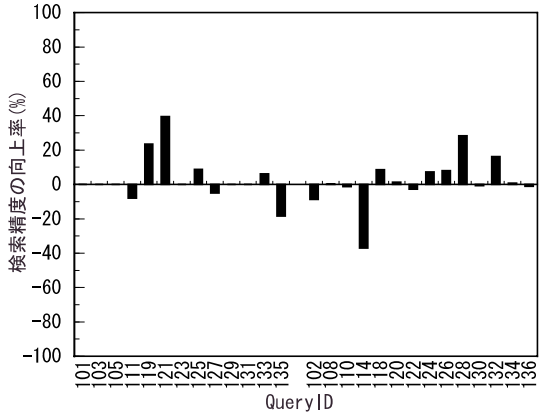

図 7 Query ごとの検索精度
Fig. 7 11 point average of the interpolated recall-precision average for each query.

以下の文を含んでいた．

> 二〇一〇年には百二十三兆円の市場と二百四十万人の雇用が見込めるなど産業面への効果のほか，通信を使った医療や教育など家庭生活にも多様なインパクトがあり，“真に豊かな社会が実現” できると指摘した．

この文書の主題は “マルチメディア” である．この文書の中で “産業” と “教育” はともにマルチメディアの影響分野であるという意味的な関係を持っている．しかし，このような意味的な関係は Query “教育産業” の中にある教育と産業の関係とは明らかに違う．このように，今回のグラフモデルの実現では，意味的な関係が違うものに対しても同一文内で共起しているという表層的な関係だけで高い関連度を与えてしまう場合があり，これが類似度計算の精度の低下につながってしまったと考えられる．

4.1.2 複数主題文書に対するグラフモデルの有効性の検証（実験 2）

BMIR-J2 の新聞記事のような均質な文書では，Web ページやシステムの紹介文書等の複数主題文書を必ずしも表現できていない．そこで，これらの複数主題文書を表現できるように，BMIR-J2 の新聞記事をランダムに n 個まとめて 1 つの文書とすることによって，新しい評価セットを人工的に作成した．この新しい評価セットの検索対象文書は，それぞれ新聞記事 n 個を含み，検索対象文書の総数は（5,080/n）件である．質問に対して，n 個のうち 1 つでも正解記事を含めば正解文書とした．また，複数の正解記事を含む文書を検索した場合でも正解数は 1 と数えた☆．n を 1，2，4，8，16 とした評価セットを作成し，$p = 0.7$ とし，QuerySet2 を用いて検索精度の評価を行った．

結合した文書数 n と，ベクトル空間モデルの 11pt-ave を基準とした場合のグラフモデルの検索精度の向上率との関係を**図 8** に示した．また，$n = 16$ のときの再現率–適合率の関係を**図 9** に示した．

図 8 を見ると文書が長くなるに従って，ベクトル空間モデルと比べた場合のグラフモデルの検索精度が向上している．特に $n = 8$，$n = 16$ のときには，約 10%も検索精度が向上している．また，図 9 を見るとグラフモデルは，ベクトル空間モデルと比べて，どの再現率においても検索精度が向上している．この結果から，複数主題文書を対象とした場合の検索精度の向上にグラフモデルが有効である．

4.1.3 単語間の関連度の計算方法の比較（実験 3）

主題グラフの作成および類似度計算のアルゴリズムから明らかなように，関連度の計算方法が検索精度に影響を及ぼすものと考えられる．そこで，共起に基づく手法と係り受けに基づく手法の 2 つの単語間の関連度の計算方法について，$n = 16$，$p = 0.7$，QuerySet2 を用いて検索精度の評価を行った．ここで，検索対象文書に共起に基づく手法を用いる場合には，検索要求文に対しても共起に基づく手法を用いた．同様に係り

☆ このような正解の定義以外に，それぞれの文書に含まれる正解記事数や正解記事の割合によって，正解文書に順位を付ける方法も考えられる．しかし，たとえば，ユーザが “形態素解析” について調べたい場合に，“形態素解析に関しても述べられている自然言語処理に関する本” と “形態素解析についての論文” を比較し，ユーザにとってどちらがより有用であるかや，検索要求により合致するのはどちらかの判断をつけることは難しい．このような理由から正解文書の順位づけは行わなかった．

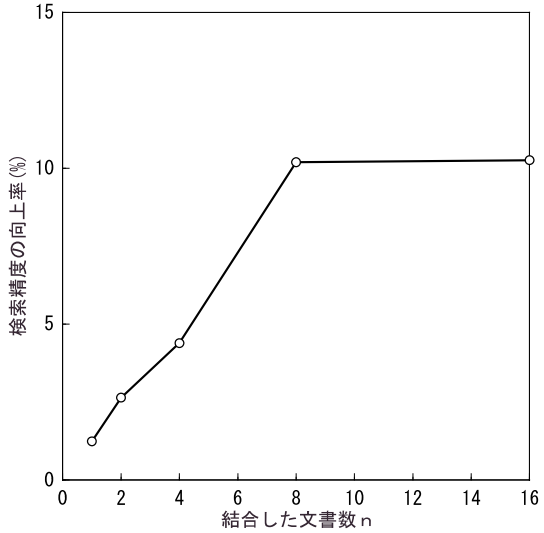

図 8 文書を複数結合したときの検索精度
Fig. 8 11 point average of the interpolated recall-precision average versus the number of documents combined.

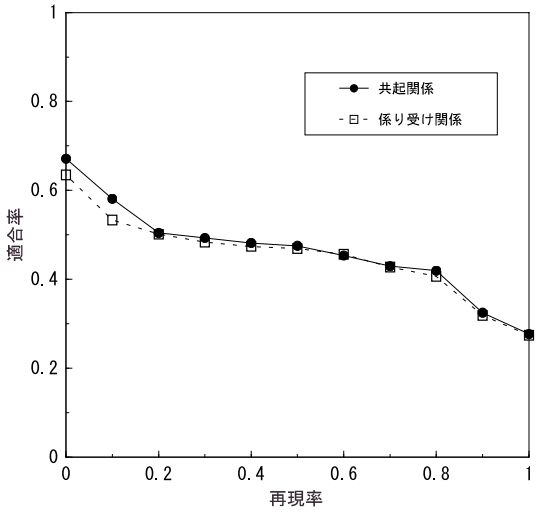

図 10 共起関係と係り受け関係を用いた場合の比較
Fig. 10 Comparison of the use of co-occurence and dependency.

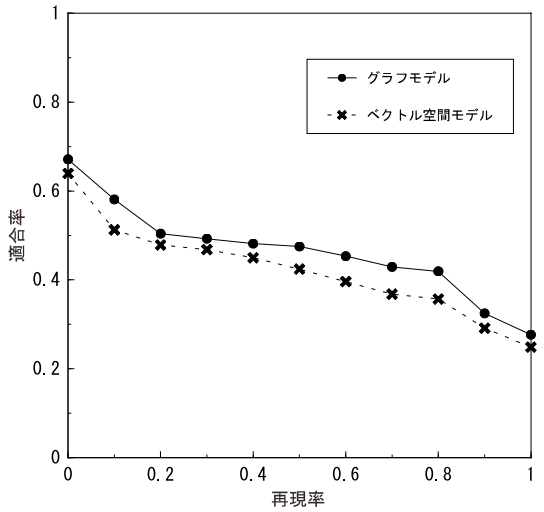

図 9 複数主題文書に対するグラフモデルとベクトル空間モデルの比較
Fig. 9 Comparison of Graph model and the Vector space model when each document has many topics.

受けに基づく手法を用いる場合には，検索要求文に対しても係り受けに基づく手法を用いた．**図 10** に再現率–適合率の関係を示す．

図 10 を見ると単語間の関連度を計算するのに共起関係を用いても係り受け関係を用いてもほとんど検索精度に差がないことが分かる．この結果から，ある文の中で単語どうしが直接の係り受け関係にあるかどうかは平均的な検索精度の向上に大きな影響を及ぼしていない．つまり，文で表現された検索キーと，検索対象文書との類似性判定に限れば，文内共起を用いて関連度を計算すれば十分であると考えられる．

4.2 システム効率の評価

ユーザの検索要求をオンラインで処理するために必要となる検索時間と，前処理に必要となるインデクス作成時間およびインデクスサイズの評価を示す．以下の評価では，測定環境として，Solaris2.6，Sun Ultra Enterprize 450 （UltraSPARC-II 296 MHz× 2，メモリ 512 Mb）を用いた．

4.2.1 検索速度の評価

BMIR-J2 の新聞記事 5,080 件を検索対象とし，50 件の検索要求について，ベクトル空間モデルとグラフモデルの検索時間を比較した．予備実験を行い，検索対象全体の文書量が増加した場合でもヒット件数が同じならば，検索時間にほとんど差がないことを確認した．**図 11** にヒット件数と検索時間の関係を示す☆．参考のため freeWAIS-sf version-2.2.10 の検索時間も測定した．

図 11 から，グラフモデルの検索時間は，freeWAIS-sf のそれよりも短い．また，検索対象全体の文書量には依存せず，ヒット件数に依存するものであるので，適用分野によっては実用的な範囲内であると考えられる．しかし，ベクトル空間モデルの検索時間よりもかなり長くなっている．グラフモデルの検索時間の増加の主な原因は，関連度インデクスにアクセスし関連度ベクトルを作成する処理である．関連度インデクスへ

☆ 検索時間として，検索キーを検索サーバが受信してから検索結果が作成されるまでの時間を測定した．

のアクセス時間は，そのインデクス構造から明らかなように共起ペアの個数に比例する．共起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 $m(m-1)/2$ 個となるので，このような最悪の場合を想定すると異なり語の個数が大きい長文の検索には適用が難しい．そこで，実際に適用可能な領域を明らかにするために文書内の異なり語の個数と共起ペアの個数に関して測定を行った．

BMIR-J2 の新聞記事 5,080 件とランダムに収集した Web ページ約 34,000 件を用いて異なり語の個数と共起ペアの個数の平均と標準偏差を求めた（**表 2**）．ここでそれぞれの文書からの単語の抽出には，本検索システムと同様に Juman version 3.5 を利用し，名詞と動詞（原型に変形）だけを単語として用いた．また，この新聞記事 5,080 件の異なり語の個数と共起ペアの個数の関係の分布を**図 12** に示した．比較のため，異なり語の個数 m と，そのすべての組合せの個数 $m(m-1)/2$ の関係を描いた．

表 2 から新聞記事の共起ペアの個数の平均と標準偏差は 1134.0，463.7 である．したがって，この程度の共起ペアの個数の平均と標準偏差を持つ文書セットを対象とする場合は，今回の測定と同程度の検索速度が期待できる．また，図 12 から，異なり語の個数 m に対して実際の共起ペアの個数は，そのすべての組合せである $m(m-1)/2$ よりもはるかに小さいことが分かる．そのため，検索対象がある程度長文となったとしても，今回の実験と似た性質を持つ文書セット（たとえば他の新聞記事等）に対しては，本手法を適用することが可能であると考えられる．

一方，Web ページの共起ペアの個数の平均は 1616.4 であり，今回の新聞記事と比べて多少多い程度である．

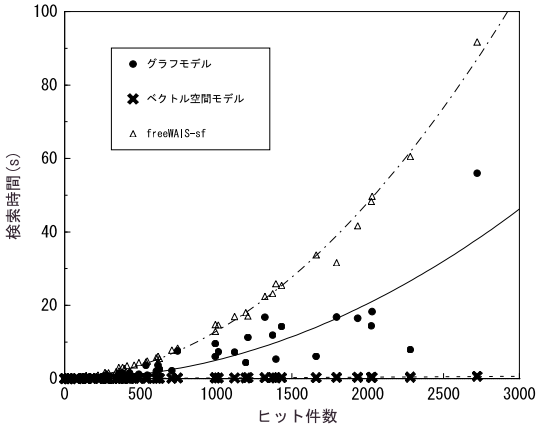

図 11 グラフモデルとベクトル空間モデルの検索時間の比較
Fig. 11 Comparison of Graph model and the Vector space model in search time.

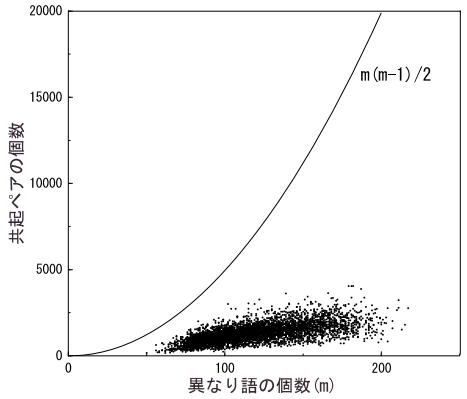

図 12 異なり語の個数と共起ペアの個数の関係
Fig. 12 The number of distinct words versus the number of co-occurence words.

表 2 文書に含まれる異なり語と共起ペアの個数の平均と標準偏差
Table 2 Average number and standard deviation of distinct words and co-occurence words.

	異なり語の個数		共起ペアの個数	
	平均	標準偏差	平均	標準偏差
新聞記事	110.5	28.3	1134.0	463.7
Web ページ	130.8	199.9	1616.4	10208.1

表 3 転置インデクスと関連度インデクスの作成時間とインデクスサイズ
Table 3 Time and space of inverted file and association index.

文書数	インデクス作成時間（秒）		インデクスサイズ（Mb）	
	転置インデクス	関連度インデクス	転置インデクス	関連度インデクス
100	3	42	0.6	2.5
500	8	206	1.8	12.0
1000	13	424	3.1	25.1
5080	66	2115	12.5	128.4

したがって，平均的には今回の測定の数倍程度の検索時間で検索を行うことができると予測できる．しかし，標準偏差の値が非常に大きいので検索時間にばらつきが大きくなり，ときどき極端に検索時間が長くなってしまうことが予想される．

4.2.2 インデクスサイズと作成時間の評価

グラフモデルでは，転置インデクスに加えて関連度インデクスを作成する必要がある．関連度を計算する際に，式 (12) の *idf* 値をあらかじめ転置インデクス作成時に計算しておけば関連度インデクスの作成は，文書単位に独立した処理となる．そのためインデクスサイズと作成時間は文書量の増加に対してほぼ線形に増加する．そこで，BMIR-J2 の新聞記事，100 件，500 件，1,000 件，5,080 件を対象として，転置インデクス，関連度インデクスそれぞれの作成時間とインデクスサイズを測定した（**表 3**）．ここで，転置インデクスの作成には freeWAIS-sf version 2.2.10 のインデクサを用いた．単語間の関連度は，共起に基づく手法を用いて計算した．

表 3 を見ると，グラフモデルで必要となる関連度インデクスの作成時間とサイズは，ともに文書量に対してほぼ線形のオーダではあるが，転置インデクスのそれと比べて大きいことが分かる．関連度インデクスの作成時間とサイズが大きいのは，そのインデクス構造から明らかなように，それぞれの文書の中の各共起ペアに対して関連度を計算する必要があるからである．したがって，長文や大規模の文書セットを対象とした場合でも，現実的な作成時間，インデクスサイズとするためには，高速なインデクスアルゴリズムやインデクスの圧縮手法等の検討が必要である．

5. 考　　察

4.1 節の検索精度の評価から，グラフモデルを用いたテキスト検索システムでは，複数主題文書を対象とした場合に検索精度が向上することが分かった．このことから，グラフモデルは，検索キーのような単一主題文書と，複数主題文書や文書集合との類似性判定が必要な場面に特に有効である．

- 3 章で述べたように，Web ページ，システムの紹介文書等は，複数主題文書と考えられるので，グラフモデルの適用による類似性判定の精度の向上が期待できる．
- 内容に基づくフィルタリングでは，ユーザプロファイルをユーザが高い評価を与えた文書集合で表すことがあり，文書のカテゴリ分けでは，分類カテゴリの特徴をあらかじめ教師データとして与えた文書集合で表すことがある．そのため，新聞記事等を対象とした情報フィルタリングや文書のカテゴリ分けでは，単一主題文書（新聞記事等）と，文書集合（ユーザプロファイルや分類カテゴリの特徴等）との類似性判定を行うことになるので，グラフモデルの適用による類似性判定の精度の向上が期待できる．

同様に，グラフモデルは文書クラスタリングにも有効に働くと考えられるが，文書集合（クラスタ）どうしの類似度を計算することになるため今回の評価とは異なった結果が得られる可能性もある．

今回の評価実験ではまったくランダムに文書をまとめたが，分類カテゴリの特徴やユーザプロファイルでは文書をまとめる何らかの基準を外部から与える場合が通常である．たとえば，このような基準として政治，経済といった分野がある．また，クラスタリングでは，主題間の類似性の高い文書をまとめることによって新たなクラスタを生成する．したがって，実際にこれらの領域にグラフモデルを適用する際には，このような文書をまとめる基準が，どの程度類似性判定の精度に影響するのかを検証する必要がある．

本稿では，重要度，関連度の計算に，それぞれ tf*idf 法と共起関係および係り受け関係を用い，主題グラフ間の類似度の計算に，重要度のベクトルと関連度のベクトルのそれぞれの内積の線形結合を用いた．しかし，これらの計算方法は容易に変更することができる．たとえば，文書からパッセージ（段落や意味的なまとまり）を抽出する TextTiling [4]，語彙的連鎖[16]といった手法や，格情報[17]，単語の近接関係等を関連度計算に利用することができる．また，ベクトル空間モデルの精度向上のために提案されている様々な重要度の計算方法やベクトル間の類似度計算尺度を利用することも可能である．このように重要度，関連度，グラフ間の類似度の計算方法を工夫することによって，本手法の類似性判定の精度をさらに向上させることができると考えられる．

4.2 節のシステム効率の評価から，単語間関連度の計算コストは大きいものの，主題グラフどうしの類似度計算自体にはそれほど多くの時間がかかっていない．そのため，前処理によって関連度インデクスを作成できるような状況であれば，テキスト検索のようなリアルタイムの類似度計算が必要な場面にも，ある程度対象文書を限定すればグラフモデルを適用することができる．ただし，現在のインデクス構造のままでは，Web ページのような共起ペア数のばらつきが大きい文書，非常に長い文書，大規模文書セットの検索には

適用が難しい．そのため，これらの文書を対象とする場合には，何らかの近似手法を用いる必要がある．たとえば，関連度の小さい共起ペアはインデクスに登録しないといった手法が有効である．また，検索時間はほぼヒットした各文書の類似度計算時間の和であることから，まず，転置インデクスに含まれる単語の重要度を用いてランキングを行い，検索結果の上位 n 件の文書についてだけ関連度を用いて再ランキングを行う手法も有効である．ただし，これらの近似手法の利用に際しては，ある閾値を設定した場合，どの程度，検索精度に影響を及ぼすのかの検証が必要である．

6. 結論および今後の課題

本稿では，文書間の類似性判定の新しいモデルとして，グラフモデルを提案した．グラフモデルは，従来のベクトル空間モデルと異なり，単語の重要度だけでなく，単語間の関連度も考慮した文書間の類似性判定ができる．このモデルに基づくテキスト検索システムを実現し評価を行った．検索精度の評価実験において，新聞記事 16 個をまとめることによって作成した複数主題文書の評価セットを対象とした場合に，グラフモデルの検索精度がベクトル空間モデルのそれと比べて約 10%向上した．この結果から，グラフモデルは，特に Web ページ等の複数主題文書を対象とした文書間の類似性判定に有効である．さらに，ユーザプロファイル，分類カテゴリの特徴，クラスタ等の文書集合を対象とした類似性判定にも，グラフモデルは有効であると考えられる．また，システム効率の評価実験から，単語間の関連度をあらかじめ計算することによって，グラフモデルをテキスト検索システム等のリアルタイムの類似性判定が必要なものにも適用可能であることを示した．

今回の評価実験では比較的短い検索キーとランダムに文書を結合することによって作成した複数主題文書間での類似性判定の精度についての評価を行った．しかし，あらかじめ何らかの基準で分類された文書集合を対象とした類似性判定や，クラスタリングの際に必要となる文書集合どうしの類似性判定に，グラフモデルがどの程度有効かの定量的な評価は今後の課題である．また，グラフモデルでは，単語の重要度，単語間の関連度，グラフ間の類似度の計算方法を容易に変更することができる．今後，これらの計算方法の違いが類似性判定の精度にどのように影響するのかの検証を行いたい．さらに，主題そのものを表現する手段として主題グラフを再考し，要約等の分野へ適用する方法を考案することも今後の課題である．

参 考 文 献

1) 長尾　真ほか：自然言語処理システムの動向に関する調査報告書，日本電子工業振興協会 (1998).
2) Salton, G., Wong, A. and Yang, C.S.: A Vector Space Model for Automatic Indexing, *Comm. ACM*, Vol.18, pp.613–620 (1975).
3) Salton, G., Allan, J. and Buckley, C.: Approaches to Passage Retrieval in Full Text Information Systems, *Proc. SIGIR'93*, pp.49–58 (1993).
4) Hearst, M.A. and Plaunt, C.: Sub-topic Structuring for Full-Length Document Access, *Proc. ACM-SIGIR'93*, pp.59–68 (1993).
5) 丸山　宏：グラフのマッチングを用いた意味解析，情報処理自然言語処理研究会，Vol.58, No.3 (1986).
6) Belkin, N.J., Oddy, R.N. and Brooks, H.M.: Ask for Information Retrieval: Part II. Results of a Design Study, *The Journal of Documentation*, Vol.38, No.3, pp.145–164 (1982).
7) Text REtrieval Conference (TREC): http://trec.nist.gov/.
8) 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（NTCIR）: http://research.nii.ac.jp/ntcir/.
9) IREX 実行委員会：IREX ワークショップ予稿集 (1999).
10) 木谷　強ほか：日本語情報検索システム評価用テストコレクション BMIR-J2，情報処理学会データベースシステム研究会，Vol.114, No.3, pp.15–22 (1998).
11) 長尾　真，佐藤理史，黒橋禎夫：自然言語処理，岩波書店 (1996).
12) Salton, G. and Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management*, Vol.24, pp.513–523 (1988).
13) Pfeifer, U.: *freeWAIS-sf 2.0* (1995). http://ls6-www.cs.uni-dortmund.de/ir/projects/freeWAIS-sf/
14) Church, K. and Hanks, P.: Word Association Norms, Mutual Information and Lexicography, *Computational Linguistics*, Vol.16, No.1, pp.22–29 (1990).
15) 五十嵐幸雄：「知的」検索技術が製品に，手間，洩れ，無駄を削減，日経エレクトロニクス，No.705, pp.63–70 (1997).
16) 望月　源，岩山　真，奥村　学：語彙的連鎖に基づくパッセージ検索，自然言語処理，Vol.6, No.3, pp.101–126 (1999).
17) 大森信行，岡村　潤，森　辰則，中川裕志：情報検索手法を利用した関連マニュアル群のハイパーテキスト化，情報処理学会論文誌，Vol.40, No.6,

pp.2776–2784 (1999).

(平成 13 年 6 月 25 日受付)
(平成 13 年 10 月 30 日採録)

（担当編集委員 河野 浩之）

富田 準二（正会員）

日本電信電話株式会社サイバースペース研究所所属．1997 年慶應義塾大学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．以来，情報検索，自然言語処理，XML 関連の研究開発に従事．日本ソフトウェア科学会会員．

竹野 浩（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主任研究員．1987 年大阪大学大学院基礎工学研究科博士前期課程修了後，日本電信電話株式会社に入社．以来，通信処理，情報検索の研究開発に従事．電子情報通信学会会員．

菊井玄一郎（正会員）

京都大学工学部電気工学第二専攻修士課程修了．ただちに NTT に入社，2001 年 4 月より（株）国際電気通信基礎技術研究所（ATR）に出向，現在に至る．自然言語処理，音声言語処理，特に自動翻訳，多言語情報検索等の研究開発に従事．ACL，人工知能学会，言語処理学会会員．

林 良彦（正会員）

日本電信電話株式会社サイバースペース研究所主幹研究員．1983 年早稲田大学大学院理工学研究科博士前期課程修了後，日本電信電話公社に入社．以来，自然言語処理，情報検索の研究開発に従事．博士（工学）．言語処理学会，人工知能学会，電子情報通信学会会員．

池田 哲夫（正会員）

1979 年東京大学理学部情報科学科卒業．1981 年東京大学大学院理学系研究科情報科学専攻修士課程修了．同年日本電信電話公社（現 NTT）電気通信研究所入所．現在，NTT サイバースペース研究所主任研究員．この間，プログラム言語の意味論の研究，データベース管理システムの研究開発などに従事．ACM，IEEE CS 各会員．工学博士．