情報学基礎　64-16
自然言語処理 145-16
（２００１．９．１１）

概念間の関連度に基づく情報ランク付けを用いた情報検索手法

藤井　啓彰，渡部　広一，河岡　司

同志社大学大学院　工学研究科　知識工学専攻
〒610-0394 京都府京田辺市多々羅戸谷 1-3

情報があふれる社会の中で必要な情報を得る情報検索が必要となってくる．情報検索で必要な情報をユーザーからのキーワードを使ったブーリアンモデルで絞り込むだけでは何千，何万と該当文書がヒットしたときにユーザーはどの文書から見て良いのか分からない．そこでユーザーからの検索質問文と文書の定量化を行いユーザーの要求と関連している順に情報をランク付けし，ユーザーに要求により相応しい情報を与えることが必要である．検索質問文と文書間の定量化についての計算方法としてベクトル空間モデルが存在するが，本研究では連想概念ベースを元にした概念間の関連度を利用して定量化する重み付き関連度計算が情報ランク付けに有効であることを示す．

The Method of Information Retrieval using The Ranking Based on The Degree of Association between Concepts

Hiroaki Fujii , Hirokazu Watabe and Tsukasa Kawaoka

Graduate School of Engineering , Doshisha University
Kyotanabe Kyoto , 610-0394

In this information-oriented society that we live in, a method of retrieving necessary information is needed. Obtaining necessary information, by information retrieval using the Boolean method, which uses keywords given by the user, narrows corresponding documents down to the thousands or ten thousands, which results in the user not knowing which document to look at first. Therefore, by transferring the user's retrieval query and documents into quantitative values and ranking the requested information in order of relation to the user's demand, it is needed to give the user information that fits the user's demand. As a calculation method to transfer the retrieval query and the documents into quantitative values, the vector space model exists, but this paper shows that the method of ranking information, using the concept-base and the degree of association, is effective.

1. はじめに

今日, コンピュータのパフォーマンスは指数関数的に延びていき, それに伴い蓄積できる情報量も同様に増えている．しかし IT の情報と言えば聞こえはよいが, 情報には良い情報と悪い情報が存在する. 情報化社会では情報を発信, 蓄積するだけでなく自分にとって良い情報を得ることが必要となってくる. そして膨大な情報の中から自分に必要な情報のみを取り出す検索という作業が生まれ, 様々な検索方式が考えられてきた. その代表的なモデルとしてブーリアンモデル[3]やベクトル空間法[3]などが存在する．Google や infoseek などの代表的な大規模 web 検索システムでは計算速度の面からブーリアンモデルが用いられている. 必要な文書群をユーザーからのキーワードで絞り込み, その上で文書群に対しユーザーの要求に相応しい順にランクを付ける. 情報にランクを付ける方法はベクトル空間法や確率モデル, Google の PageRank などが存在する.

本研究では情報検索のための情報ランク付けのための検索質問文と文書間の類似度計算

方法について, 重み付き関連度計算の有効性を示す. ここで一般に用いられる情報ランク付けのための計算方法としてベクトル空間法が存在するが, この計算方法だとキーワード「道路」と「信号機」は直交する単語として一致度は 0 になってしまう．一致させたいキーワードが「人」と「人間」なら同義・類義問題で辞書参照により解決できる問題だが,「道路」と「信号機」のように同義・類義, その他の論理関係でも無い関係だが関連はあるという単語までも直交して取り扱ってしまう.

情報ランク付けの計算方法ではないが「道路」と「信号機」,「机」と「勉強」等の関連語を考慮した索引手法として LSI(Latent Semantic Indexing)[3]が存在する．LSI とは索引語の共起情報を複数の索引語を同一シンボルに統合し, 同義・類義語問題や表記揺れ問題の解消がされると言われている索引手法である. また精度においても若干の向上があると言われている．しかし，大規模な文書-索引語行列を特異値分解するにはかなりの Computing Power が必要であり，しかも小規模の文書-索引語行列を特異値分解したならばデータの小ささ故に偏りが大きく悪影響を与え一般的な常識とは異なった索引語の縮体が行われてしまうことがあり, 検索するユーザーから見て常識的な感覚とは異なる結果になるだろう.

本研究では情報をランク付けする際に常識的な判断を実現するための連想概念ベースを用い，重み付き関連度計算の有効性を示す.

2. 情報検索の一般的な技術

情報検索でよく用いられる技術や概念について説明する.

2.1 文書-索引語行列

ある文書 doc_i を以下のように定義する.

$$doc_i = \{(d_1, w_1), (d_2, w_2), \cdots, (d_L, w_L)\}$$

doc_i は語 d_j とその重み w_j の対の集合で表現する. この語 d_j を情報検索では索引語という. 検索対象とする文書数が M で索引語の異なり数が N であった場合，M 個の文書を以下のようなマトリックスで表現できる.

$$A = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ w_{M1} & w_{M2} & \cdots & w_{MN} \end{bmatrix}$$

w_{xy} は文書 doc_x における索引語 d_y の重みを表す．M 個の文書を N 個の索引語で表現したマトリックスを文書-索引語行列という.

2.2 *tf・idf* による文書-索引語行列の重み

文書-索引語行列 A の要素 w_{xy} は文書 doc_x における索引語 d_y の重みを表すが，その重み w_{xy} は以下の計算式によって得られる.

$$w_{xy} = tf(d_y) \cdot idf(d_y)$$

$tf(d_y)$ は文書 doc_x における索引語 d_y の出現頻度である．$idf(d_y)$ は索引語 d_y が出現する文書数によって決まり, 以下の式によって定義される.

$$idf(d_y) = 1 + \log\left(\frac{M}{M_{d_y}}\right)$$

M_{d_y} は索引語 d_y が出現する文書数

頻度 $tf(d_y)$ と索引語の網羅性 $idf(d_y)$ によって計算されることから, この重み計算方式を一般に *tf・idf* と言う.

2.3 LSI(Latent Semantic Indexing)

LSI は統計的な処理によって語を抽象化する手法である．そして，この抽象化された語を索引語と考えて検索を行う. 具体的には以下のような方法で語を抽象化する．まず，文書索引語行列を特異値分解によって，以下のように 3 つの行列に分解する.

$$A = T_0 S_0 D_0^T$$

文書集合中の索引語の異なり数を N，総文書数を M とすると，行列 T_0 は M 行 m 列，S_0 は m 行 m 列，D_0 は N 行 m 列となる．また，T_0 と D_0 は正規直交行列，S_0 は対角行列となる．一般に $m \le \min(N, M)$ なので，S_0 の m 個の対角要素を抽象化された語として扱うことができる．また，一定のしきい値を設け，そのしきい値より小さい対角要素を 0 に置き換え

ることによって，さらにS_0の次元を下げることができる．

英語圏では同じ単語の繰り返しを防ぐために文章中に同義語を用いることが多いが, 日本語では同じ単語を繰り返すことに抵抗感はないので同義語の共起情報が得られず, LSI による索引手法はあまり効果的でないとも言われている[3].

2.4 ブーリアンモデル

ブーリアンモデルとは最も簡単な情報検索モデルである．ユーザーからのキーワードを AND や OR,NOT 等の論理演算子を用いて検索対象文書を絞り込む方法である. このモデルの長所は検索にかかる時間が少ないと言うことである. 計算速度において大規模な検索システムに使用される．しかし, ユーザーからのキーワードを満たす文書ならばどの文書も適合している結果と見なすので, 後に述べるベクトル空間モデルでユーザーからのキーワードと文書の間の類似度で検索結果をランク付けすることが多い.

2.5 ベクトル空間モデル

ユーザーからのキーワードと文書との間の類似度を計算することにより, ユーザーの要求により適合している結果をユーザーに返すモデルである．文書の定義を以下のようにする.

$$doc_i = \{(d_1, w_1), (d_2, w_2), \cdots, (d_L, w_L)\}$$

doc_iは語d_jとその重みw_jの対の集合であるが，計算の際には重みw_jが並んだベクトルと見なしている.

ユーザーからの要求を以下のように定義する.

$$query = \{(q_1, w_1), (q_2, w_2), \cdots, (q_M, w_M)\}$$

q_iは検索質問文中のキーワードであり，ユーザーからの要求はq_iとその重みw_iの対の集合で表すことができるが，$query$もdoc_iと同様に計算の際には重みw_jの並んだベクトルと見なしている．ある文書doc_iとユーザーからの要求$query$との類似度を計算するベクトル空間法$Vector(query, doc_i)$は以下の式で定義される.

$$\begin{aligned} & Vector(query, doc_i) \\ &= \cos\theta \\ &= \frac{query \cdot doc_i}{|query| \cdot |doc_i|} \end{aligned}$$

θは 2 つのベクトル$query$とdoc_iのなす角度で類似度はその 2 つのベクトルの余弦値$\cos\theta$である．類似度が高い順に文書をランク付け, ユーザーに結果を返すのがベクトル空間モデルである.

ベクトル空間モデルの利点はブーリアンモデルの欠点と表裏一体の関係になる. ブーリアンモデルには索引語の間の関係を論理演算子を用いて表現できるという利点があるが, ベクトル空間モデルではこの利点は失われている.

3. 重み付き関連度計算と連想概念ベース

この章では語の関連の深さを定量化する手法としての重み付き関連度計算と, 重み付き関連度計算を実現する連想概念ベースについて説明する.

3.1 連想概念ベース

連想概念ベースにおいて概念 A の定義は以下とする.

$$A = \{(a_1, w_1), (a_2, w_2), \cdots\cdot (a_N, w_N)\}$$

概念Aはその属性a_iと重みw_iの対の集合で表現される. 図１に連想概念ベースと概念の例を挙げる. 重みは 10,9,4,3,1 の５段階に付与されている. 連想概念ベースは辞書と新聞から構築され, シソーラスなど様々な知識ベースを用いて精錬されている.

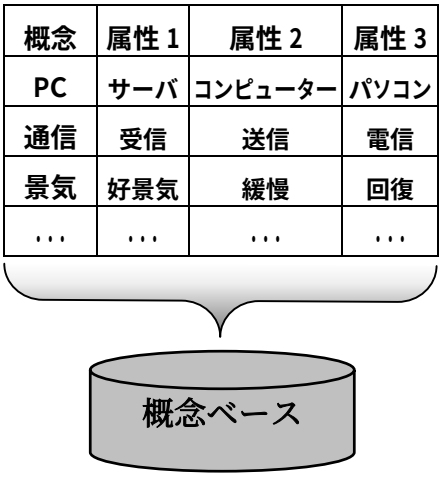

図１　　連想概念ベースとサンプル概念

3.2 連想概念ベースを利用した重み付き関連度計算

概念Aはその属性a_iと重みw_iの対の集合で表現されているが，ある一つの属性a_iもまた

概念と見なせる．よって概念a_iも以下のように定義できる．

$$a_i = \{(a_{i1}, w_{i1}), (a_{i2}, w_{i2}), \cdots\cdots (a_{iM}, w_{iM})\}$$

概念Aから見ればa_{ij}は属性の属性であり，a_{ij}をAの２次属性と定義し，a_iをAの１次属性と定義する．２次属性まで属性の連鎖的な取得によって概念Aは以下のようなマトリックス状になる．

$$A = \begin{matrix} (a_1 & a_2 & \cdots & a_N) \end{matrix} \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{N1} \\ a_{12} & a_{22} & \cdots & a_{N2} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ a_{1M} & a_{2M} & \cdots & a_{NM} \end{bmatrix}$$

概念Aと概念Bの関連の深さを定量化する重み付き関連度のアルゴリズム *ChainW*[2]は，１次属性集合の一致度を計算するアルゴリズム *MatchW* から計算される．１次属性集合の一致度計算アルゴリズム *MatchW* の計算は以下のように定義される．

$$\text{概念} A = \{(a_i, u_i) \mid i = 1 \sim L\}$$

$$\text{概念} B = \{(b_j, v_j) \mid j = 1 \sim M\}$$

$$MatchW(A,B) = (s_A / n_A + s_B / n_B)/2$$

$$s_A = \sum_{a_i = b_j} u_i \qquad s_B = \sum_{a_i = b_j} v_j$$

$$n_A = \sum_{i=1}^{L} u_i \qquad n_B = \sum_{j=1}^{M} v_j$$

重み付き関連度計算アルゴリズム *ChainW* は以下のようになる．

（１）１次属性数の少ない方の概念を概念Aとし($L \leqq M$)，概念Aの１次属性の並びを固定する．

$$A = \{(a_1, u_1), (a_2, u_2), \cdots\cdots (a_L, u_L)\}$$

（２）概念Bの各１次属性を対応する概念Aの各１次属性との重み付き一致度(*MatchW*)の合計が最大になるように並び替える．ただし，対応にあふれた概念Bの１次属性$(b_{xj}, j = L+1, \cdots, M)$は無視する．

$$B = \{(b_{x1}, v_{x1}), (b_{x2}, v_{x2}), \cdots\cdots (b_{xL}, v_{xL})\}$$

（３）概念Aと概念Bとの重み付き関連度*ChainW(A,B)*は以下の式で表す．

$$ChainW = (s_A / n_A + s_B / n_B)/2$$

$$s_A = \sum_{i=1}^{L} u_i MatchW(a_i, b_{xi})$$

$$s_B = \sum_{i=1}^{L} v_{xi} MatchW(a_i, b_{xi})$$

$$n_A = \sum_{i=1}^{L} u_i \qquad n_B = \sum_{i=1}^{M} v_j$$

4. 重み付き関連度計算を利用した情報のランク付け

文書検索においてユーザーが要求する文書をキーワードでブール代数を用いて絞り込み，その結果をユーザーに返すという検索モデルが多い．しかし，より良い結果をユーザーに返すためにはユーザーが要求する検索質問文の文書との関連の深さを定量化し，文書にランク付けを行い上位の文書をユーザーに返すのがよい．本研究ではユーザーの要求を概念とし，文書で表現している情報も概念として，この２つの概念間の関連の深さを重み付き関連度計算を利用して定量化する方法を提案する．

（１）文書，検索質問文を連想概念ベースに存在する語と存在しない語に分ける．あるi番目文書の索引語と重みの対の集合を表したのがdoc_i，その文書の索引語の中から連想概念ベースに存在する語と重みの対の集合を表したのがdoc_{iC}，連想概念ベースに存在しなかった語と重みの集合を表したのがdoc_{iNC}である．同様にある検索質問文*query*を$query_C$と$query_{NC}$に分ける．

$$doc_i = \{(d_1, w_1), (d_2, w_2), \cdots, (d_L, w_L)\}$$

$$query = \{(q_1, w_1), (q_2, w_2), \cdots, (q_M, w_M)\}$$

$$doc_i = doc_{iC} + doc_{iNC}$$

$$query = query_C + query_{NC}$$

（２）連想概念ベースに存在する語の集合を１次属性と見なし，重み付き関連度計算で２つの概念間の定量化をする．その時の値を*rel_value*とする．連想概念ベースに存在しない語の集合についてはベクトル空間法によりその余弦値を*vec_value*とする．

$$rel\_value = ChainW(doc_{iC}, query_C)$$

$$vec\_value = Vector(doc_{iNC}, query_{NC})$$

（３）*rel_value*と*vec_value*の２つの値を加

算した値を 2 つの概念間の関連の深さ *all_value* とする.

$$all\_value = rel\_value + vec\_value$$

図 2 に重み付き関連度計算とベクトル空間モデルを併用した情報のランク付けのフローチャートを示す.

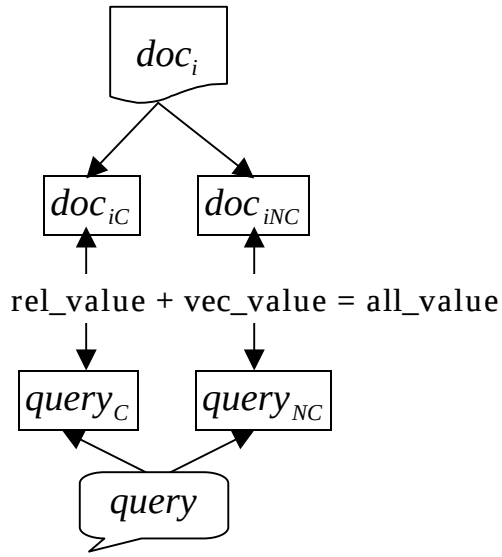

図 2　重み付き関連度計算とベクトル空間法を併用した情報のランク付け

5. 実験と評価

検索対象とする文書群は

http://yahoo.co.jp/News

から国内ニュースとスポーツニュースを 1/8 から 1/14 までの 7 日間分とした. 7 日間での文書数は 1721 であった. 人間の要求概念については 1/8 から 1/14 の 7 日間の出来事で強く記憶が残った事, 詳しく知りたい事をアンケートにより答えてもらった. それらの自然言語を茶筌[4]で形態素解析し, 名詞, 動詞, 形容詞を取り出し検索質問文とした. なお, 名詞の「人」,「誰」,「何」,「いつ」, 動詞の「する」,「ある」,「なる」と言った日常で頻繁に使用する語は取り除いた. これらの検索質問文リストは 60 セット用意した. 表 1 にアンケートの回答と検索質問文を載せる.

表 1　　検索質問文のリスト

自然言語での要求	検索質問文
大相撲初場所誰が上位で活躍している？	大相撲, 初場所, 上位 活躍
サルの遺伝子組み換えに成功した事件はどうなったか？	サル, 遺伝子, 組み替える, 成功, 事件

評価方法として, 1 つの検索質問文に対して最高の類似度/関連度を持つ文書を 1 つ選び出し, その 1 つの文書について相応しい（Rank A), 相応しくないが類似する（Rank B), 全く関連性はない（Rank C）の 3 つの評価を与える方法を採った. なお, 文書表現モデルは 2 種類用意し, 1 つは頻度重みそのままを文書表現とするもの, もう 1 つは *tf・idf* によって頻度重みを変更したものを用意した. 図 3 は文書を索引語とその頻度の対の集合で表したときのベクトル空間法と重み付き関連度計算の評価を示す. 図 4 は文書を索引語と *tf・idf* によって頻度を変更した時の結果である.

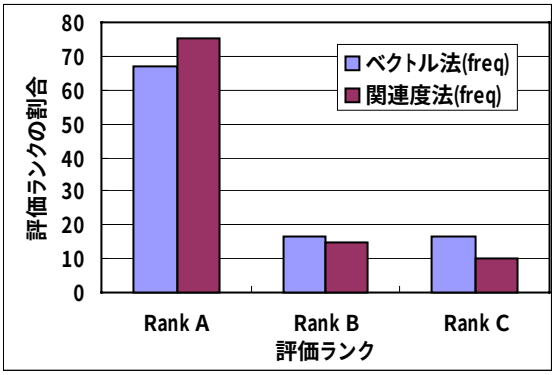

図 3 頻度重みを採用したときの評価結果

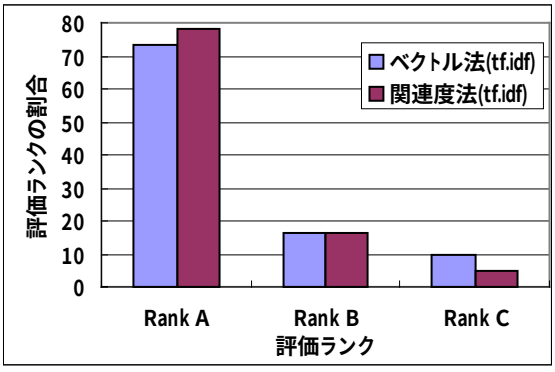

図 4 *tf・idf* 重みを採用したときの評価結果

文書表現が頻度重みを採用した物と *tf・idf* を採用した物との両方で重み付き関連度計算を利用した情報ランク付けの方が良い結果を返すことが分かる.

次に関連度計算方式で連想概念ベースの概念の属性数を変化させて実験を行った. 結果を図 5 に示す. なお, *tf・idf* によって重み計算した文書-索引語行列を使用した.

重み付き関連度計算に使用する概念の属性打ち切り個数を 5 個から 50 個で変化させてみたが評価は 5 個の場合で Rank C が多いが, ほとんど変化しないと言っていいだろう. 打ち切

り個数にこだわる必要はあまり無く, 計算コストから属性使用打ち切り個数を 15 個, 20 個にするのが良いだろう.

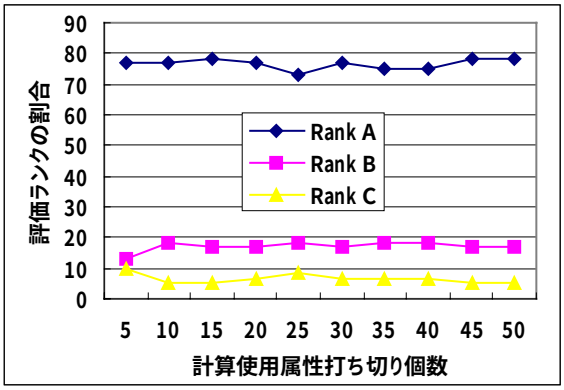

図 5　重み付き関連度計算方式で打ち切り個数を変化させた場合の評価結果

次に重み付き関連度計算が有効に作用しているのかを調べるために重み付き関連度で算出される *rel_value* と連想概念ベースに含まれない語からベクトル空間方式で算出される *vec_value* のスコア比を変化させて加算する実験を行った.

$all\_value = R \times rel\_value + V \times vec\_value$

とし, *R* と *V* の比を変化させて評価を行った. 図 6 に評価結果を示す. なお, 重み付き関連度計算に使用する属性使用打ち切り個数は 20 個である.

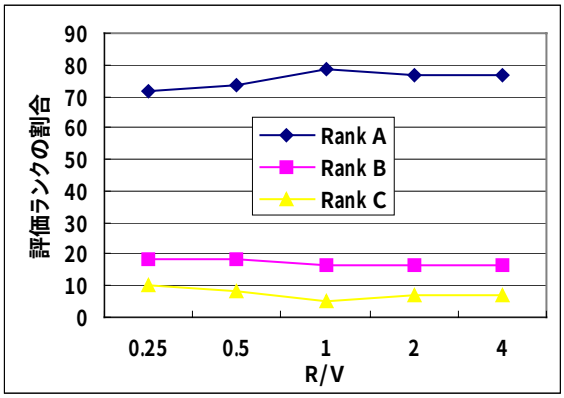

図 6 重み付き関連度計算の影響力

重み付き関連度計算から算出される *rel_value* と *vec_value* の比は 1:1 の時がもっとも良い結果となった. なお, V=1,R=0 の時の評価は Rank A が 18.33%, Rank B が 6.67%, Rank C が 6.67%であり, どの文書にも類似度が導けなかった例が 68.33%であった. これは大多数の索引語, 検索質問文中のキーワードが連想概念ベースにカバーされていることを示す. 逆に V=0,R=1 の時では Rank A が 63.33%,Rank B が 18.33%, Rank C が 18.33%となった. 重み付き関連度計算のみ使用したとき結果が悪くなったのは重み付き関連度計算 *ChainW* は連想概念ベースに含まれる語のみを対象とした計算方法であるために文書を特徴づける固有名詞などを扱っていないため十分に文書をランク付けできなかった思われる.

6. おわりに

連想概念ベースを用いて情報をランク付けする手法として重み付き関連度計算はベクトル空間法より良い結果を残した. 今回は検索質問文中のキーワードに重みを付与していないが適切な重みを付与する方法やその効果を調べる必要がある. 重み付き関連度計算の問題点として属性間に属性個数の差がある場合, 人間の感覚と違う結果となることがある. ある一方の概念の属性が 2 個で, もう一方の概念の属性の個数が 100 個だったとする. このうち 1 つの属性がヒットしたとき重み付き関連度計算では, 一方から見た場合の一致度ともう一方から見た場合の一致度の平均値を取るので 2 個しか属性を持たない概念にとってはたった一つの属性が一致しただけで 0.5 以上の関連度を取るのは感覚的におかしい. 今回の実験でも比較的文書長の小さい文書がヒットしやすかったように思える. 今後は属性の個数に依らない計算方法を考案したい.

謝辞

本研究は文部科学省からの補助を受けた同志社大学の学術フロンティアの研究の一環として行った.

参考文献

[1]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, No7, pp.1272-1283(1997)

[2]渡部　広一, 河岡　司, “常識判断のための概念間の関連度評価モデル”, 自然言語処理, Vol.8, No.2, pp39-54(2001)

[3] 徳永健伸, “言語と計算 5 情報検索と言語処理”, 東京大学出版会, (1999)

[4]http://chasen.aist-nara.ac.jp/