自然言語処理 123－8
（1998. 1. 19）

シソーラスと決定木学習アルゴリズムによるRocchio feedbackの高精度化

中島 浩之　　木谷 強
NTTデータ通信株式会社 情報科学研究所
email: nakajima@lit.rd.nttdata.co.jp

あらまし

Rocchio feedbackは検索要求文と文書をベクトルで表現するベクトル空間法において、検索者によるfeedbackを用いて検索要求文から作成したベクトルを修正する手法である。多くの研究者により文書検索の精度を向上させる効果が報告されている手法であるが、ベクトルの修正はベクトル間の加減算によってのみ行なわれるため、検索語間の共起関係を捉えることができなかった。本稿では検索者が関心を持つ文書を正例、持たない文書を負例として決定木学習アルゴリズムID3に与えることで、必要な文書と不要な文書を区別する観点で重要な共起関係を検索式の形式で獲得し、得られた検索式に適合した文書についてRocchio feedbackの検索結果を修正する手法を提案する。また学習例数が不十分な場合に正確な学習ができないというID3の欠陥を補うため、文書データベース中の大部分の文書はユーザにとって関心がない文書であることに着目し、仮想的に負例を増加させる手法を示す。さらに、シソーラスを利用することで単語の概念を扱うBit-per-categoryエンコーディングをID3の前処理として用いることで、概念レベルでの共起も扱う。実験を行なった結果、提案手法が検索精度向上に有効であることがわかった。

キーワード　文書検索、シソーラス、決定木学習、類似文書検索、概念共起

Improving Rocchio feedback through learning co-occurence of concepts

Hiroyuki Nakajima　　Tsuyosi Kitani
Laboratory for Information Technology, NTT Data Corporation

Abstract

Rocchio feedback is a method which modifies queries based on evaluated sample documents. Although it is known to be effective in selecting relevant ones, it doesn't capture any co-occurence of words. We apply the ID3 inductive learning algorithm to this task, and capture co-occurence of words which can distinguish relevant sample documents from irrelevant ones. The ID3 requires a lot of samples for learning proper co-occurence from noisy data, but in this case, enough samples aren't always given. We notice that almost all of documents in database are irrelevant, and propose a method which adds non-sample documents as provisional irrelevant samples to increase sample documents. Furthermore, we introduce Almuallim's pre-processing approach (Bit-per-category encoding) which looks up thesaurus for learning co-occurence of concepts. We apply extracted co-occurence of words and concepts to the Rocchio feedback and evaluate the results using an information retrieval system collection.

keywords　information retrieval, thesaurus, inductive learning, relevance feedback, co-occurence of concepts

1 はじめに

文書検索データベースで必要な文書を検索するためには適切な検索式を作成する必要がある。しかし検索式の作成は検索業務を専門とするサーチャーですら試行錯誤を必要とする困難な作業である [16]。relevance feedback は検索者が呈示する必要文書のサンプルを利用して検索式を修正し、新たな検索式を作成する手法である。これは試行錯誤による検索式の作成をシステムが検索者と協調して行なうものであり、文書検索精度を向上させる有効な手段と考えられている [10]。

relevance feedback を実現する代表的なアルゴリズムである Rocchio feedback [8] は、検索要求文および検索対象の文書をベクトルとして表現するベクトル空間法 [3] (Vector Space Model、VSM) において、検索者によるフィードバックを用いて文書検索の精度を向上させる手法である。検索者は検索要求文に対する検索結果の一部をサンプルとして必要か不要か判断し、システムにフィードバックする。システムは判断されたサンプル文書を用いて検索語を増加させて再度検索を行なうと共に、検索要求文を表すベクトルを修正し、検索された文書が持つベクトルとの内積をスコアとして付与することで、検索結果をスコアの高い順に順位付けして呈示する。

Rocchio feedback は文書検索の精度を向上させる有効な手段として知られており、多くの研究者からその有効性が報告されている。しかしベクトルの修正はベクトル間の加減算によってのみ行なわれるため、検索語間の関係は考慮されない。そのため、複数の検索語が一つの文書中に現れる (共起する) ことで初めて具体的な事柄を指す場合 (例えば検索要求文「マシンがクラッシュした事例」) でも、一部の検索語 (例えば「マシン」) のみが数多く登場する文書があれば高いスコアが与えられてしまうことがあった。これは Rocchio feedback は検索語間の共起をスコア計算に反映していないためであった。

複数の検索語が共起している文書は論理積 (AND) を用いた検索式により検索可能である。そのため検索語間の重要な共起がわかれば、それら検索語を AND で結合した検索式にヒットする文書のスコアを上昇させることで、検索精度の向上が期待できる。しかし一般にはどの検索語の組合せが重要な共起であるかわからないという問題がある。例えば検索要求文中の全ての検索語を AND で結合した式を作成し、全ての検索語が共起している文書を検索する場合、全ての検索語がユーザの必要とする文書に含まれるとは限らないため、必要な文書にヒットしない恐れがある。

本稿では決定木学習アルゴリズム ID3[6, 7] を用い、必要文書と不要文書を区別できる最小限の検索語の組合せを獲得することで、上記の問題の解決を試みる。ID3 は相互情報量を基準として (近似的に) 最小の決定木を作成する手法であり、より少数の検索語の組合せで、より多くのサンプル中の必要文書と不要文書を区別する検索式を得ることができる。これは

- より少ない検索語を使うため、必要文書に含まれない検索語を使うリスクが減少する
- 必要文書と不要文書を区別する検索式を作成するので、不要文書にも登場する共起は排除される

ことから、本問題に適した手法である。しかし ID3 は以下の 2 つの問題点が知られている。

- 適当な決定木が得るためには十分な数の学習例を与える必要がある
- 単語を用いて検索式を作成する場合、検索式が指し示す内容と近い内容の文書であっても、使われている単語が異なればマッチしない (検索洩れをもたらす)

relevance feedback において与えられる学習例はユーザが必要ないし不要を判断した文書であり、常に十分な数の学習例が与えられるとは期待できない。筆者は文書データベース中の大部分の文書がユーザにとって関心がない文書であることに着目し、ユーザによってフィードバックされなかった文書を仮想的に不要文書として扱うことで、学習例数を増加させる手法を提案する。

検索式作成に用いる単語については、シソーラスを背景知識として利用することで検索語を概念的に抽象化する手法が既に提案されている [4, 7, 14]。本稿では提案されている手法の一つである Bit-per-category エンコーディング [14] を用い、文書検索における当該手法の有効性を検証する。

最後に提案手法により作成した検索式を Rocchio feedback によるスコア付与と融合し、英語対象の文書検索テストコレクションを用いて評価した結果を示す。

2 既存技術

2.1 Rocchio feedback

Rocchio feedback アルゴリズムはベクトル空間法 (VSM) と TF/IDF 法を用いた文書検索システムにおいて、relevance feedback を実現する。

VSM は文書や検索要求文をベクトル空間上のベクトルとして表現する [3]。このベクトル空間は扱う単語の種類と等しい数の次元を持ち、文書は文書中で単語が持つ重要性を示す'重み'を要素としたベクトルによって表される。

TF/IDF 法は文書データベース中の多くの文書に登場する語は重要でなく、特定の文書内に多く登場する語は重要とすることで単語の'重み'を決定する手法である [3, 15, 2]。文書 d_j 中の単語 t_i の重み $w_{i,j}$ は、文書 d_j 中に単語 t_i が出現する回数 $f_{i,j}$ (Term Frequency,TF) および単語 t_i が出現する文書データベース中の文書数 n_i の逆数 (Inverted Document Frequency, IDF) を用いて以下の式により計算される。

$$w_{i,j} = \frac{(\log(f_{i,j}) + 1.0) * log(\frac{|DB|}{n_i})}{\sqrt{\sum_{k=1}^{N}[(\log(f_{k,j}) + 1.0) * log(\frac{|DB|}{n_k})]^2}} \quad (1)$$

なお $|DB|$ は文書データベース中の文書総数である。

Rocchio feedback は検索者が必要または不要の判定をした文書のベクトルを用いて検索要求文のベクトルを修正することで、検索者の意図を検索式に反映する。検索要求文のベクトルを v_q、提示した文書中から検索者が選んだ必要文書 num_{rel} 件の持つベクトルの和を v_{rel}、検索者が選ばなかった文書 (不要文書) のうち、選んだ文書より上位にある文書 num_{nonrel} 件の持つベクトルの和を v_{nonrel} としたとき、新たなベクトルは

$$v = \alpha v_q + \frac{\beta v_{rel}}{num_{rel}} - \frac{\gamma v_{nonrel}}{num_{nonrel}} \quad (2)$$

となる (α, β, γ は [1] より各々 8,16,4 とした)。Rocchio feedback によるベクトルの修正は、ベクトル間の加減算によってのみ行なわれ、各検索語間の共起は考慮されない。

検索要求文に対する文書のスコアは検索式のベクトルと文書のベクトルとの内積によって計算され、検索システムはスコアの高い文書を優先的にユーザに呈示する。

2.2 決定木学習アルゴリズム ID3

ID3 は相互情報量を尺度として用いることで (近似的に) 最小の決定木を作成するアルゴリズムである [7]。決定木は検索式を木構造で表現したものと考えることができ、ID3 により得られた決定木は容易に検索式に変換することができる。

ID3 のアルゴリズムを以下に示す。

1. 入力された正例と負例からなる集合を Set_0 とする。ここで正例、負例はそれぞれ必要文書、不要文書から抽出した自立語の集合とする。
2. 集合 Set_0 に"未分割"の印をつける。
3. "未分割"の印がついた集合のうち任意の集合 Set_i 中の正例、負例に含まれる各単語 $t_j (1 \leq j \leq N)$ について、以下の式によって相互情報量 $I(t_j)$ を計算する ("未分割"の集合がなければ終了)。

$$I(t_j) = H - H(t_j) \quad (3)$$

ここで

$$\begin{aligned}
p_i &= Set_i\text{中の正例の数} \\
n_i &= Set_i\text{中の負例の数} \\
s_i &= p_i + n_i \\
p_i(t_j) &= Set_i\text{中で}t_j\text{を含む正例の数} \\
n_i(t_j) &= Set_i\text{中で}t_j\text{を含む負例の数} \\
s_i(t_j) &= p_i(t_j) + n_i(t_j) \\
p_i(\overline{t_j}) &= Set_i\text{中で}t_j\text{を含まない正例の数} \\
n_i(\overline{t_j}) &= Set_i\text{中で}t_j\text{を含まない負例の数} \\
s_i(\overline{t_j}) &= p_i(\overline{t_j}) + n_i(\overline{t_j}) \\
h(a,b,c) &= -\{\frac{a}{c}\log_2(\frac{a}{c}) + \frac{b}{c}\log_2(\frac{b}{c})\}
\end{aligned}$$

とし、H と $H(t_j)$ は

$$H = h(p_i, n_i, s_i) \quad (4)$$

$$\begin{aligned}
H(t_j) = {} & \frac{s_i(t_j)}{s_i} h(p_i(t_j), n_i(t_j), s_i(t_j)) \\
& + \frac{s_i(\overline{t_j})}{s_i} h(p_i(\overline{t_j}), n_i(\overline{t_j}), s_i(\overline{t_j})) \quad (5)
\end{aligned}$$

とする。

4. 単語 $t_j (1 \leq j \leq N)$ から $I(t_k)$ を最大にする t_k を選ぶ (複数ある場合は任意の一つ)。$I(t_k) > 0$ の場合、t_k を持つ文書の番号からなる集合を $Set_{i'}$、持たない文書の番号からなる集合を $Set_{i''}$ とし、それぞれに"未分割"の印をつける。i', i'' は既に集合 $Set_{i'}$、$Set_{i''}$ が存在しなければ任意の数でよい。$I(t_k) = 0$ の場合は分割しない。
5. 集合 Set_i から"未分割"の印を除き、3へ戻る。

上記アルゴリズムで作成した決定木で、正例を得るパスで用いた単語を演算子 AND で結合したものを検索式とする。なお本報告では検索要求文中の検索語のみを決定木の作成に用いた。

2.3 Bit-per-category エンコーディング

検索要求文中の検索語のみを用いて検索式を作成する場合、語の言い替えには対応できないため、極めて限られた範囲の文書しか検索することができず、結果として検索洩れが生じる。このような検索洩れを低減する手段として、シソーラス辞書を参照することで概念的に近い単語を検索語として追加 (シソーラス展開) する方法が挙げられるが、不適当な語をシソーラ

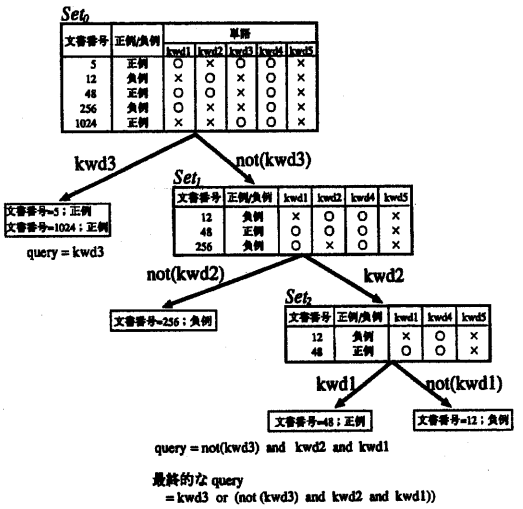

図 1: ID3 による検索式作成

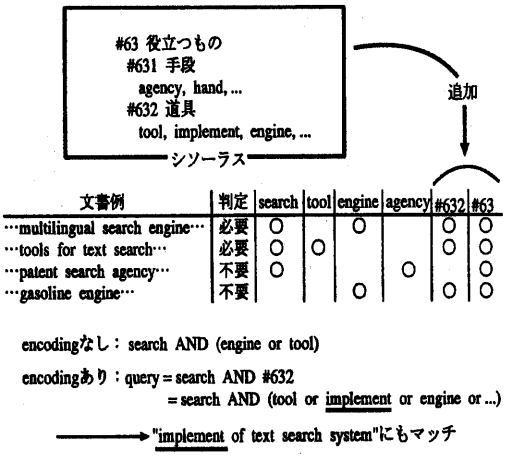

図 2: Bit-per-category エンコーディング

ス展開した場合、ノイズの混入をもたらすことが知られている。

Almuallim らは ID3 による決定木学習の過程でシソーラスを利用する Bit-per-category エンコーディング [14] を提案している。これは ID3 に学習例を与える際に例中の単語の上位概念を追加する手法で、単語を概念に置き換えることで、単語のみを用いるよりも簡素な検索式が得られる場合や、より正確にサンプル文書中の正例と負例を分割できる場合に概念を用いた検索式を作成する。例として、図 2で検索エンジンに興味を持つ検索者が、検索式 'search OR engine' により検索した結果に対し、興味を持つ文書を選択した例を示す (図中の # で始まる数字はシソーラスの概念分類番号を表す)。Bit-per-category エンコーディングを用いない場合、検索者の選択に対して ID3 は検索式 'search AND (engine OR tool) [1] ' を得る。

Bit-per-category エンコーディングは ID3 に文書を与える際に各文書中の単語のシソーラス上での上位概念語を新たな単語として加える (図 2の '#63'、'#632')。ID3 はより少ない単語を用いて決定木を作成しようとするため、検索語として 'engine' の上位概念である '#632' が選択される。また '#632' より上位の概念である '#63' では必要文書と不要文書を区別できないため '#63' は選択されない。得られた検索式で '#632' をシソーラスにおける下位概念語に展開することで、'implement of text search system' など検索者の興味に近い内容の文書を検索することができる。

[1]NOT は省略。

今回の実験ではシソーラスとして Roget シソーラス [9] を用い、サンプル文書中に同じ分類の語が複数含まれている場合に限り、その分類を加えた。なお実装上の都合から最下層の分類 (1044 分類) のみを用いた。

3 負例の追加

Rocchio feedback はサンプルとして与えられる文書数が多いほど検索精度の向上に効果を発揮する [2] が、常に多くの文書が検索者からフィードバックされるとは限らない。

Buckley らは最初の検索要求文により検索された文書のうち、検索者が必要・不要のチェックをしていない文書すべてを不要文書と仮定し、フィードバックされる不要文書の数を増加させるよう改良した手法 (modified Rocchio feedback) を提案している [1]。検索者が必要・不要のチェックをしていない文書から検索者に必要な文書を選択することが relevance feedback の目的であり、それら全てを不要文書とするのは正しい仮定ではない。しかし一般に検索対象のデータベースにおいて必要文書が占める割合は極めて小さいと考えられ、必要文書を不要文書として扱うことによる悪影響より、不要文書を増加させることによる効果の方が大きいと報告されている。

ID3 も十分な数の学習例が与えられた場合には適切な決定木を作成することができるが、学習例が不足すると適切な決定木の作成を行なえないことが多い。

本稿では modified Rocchio feedback と同様に、検索者が必要／不要のチェックをしていない文書すべてを不要文書と仮定し、検索者によりフィードバック

される不要文書に加えることでID3に与える学習例を増加させる。フィードバックされなかった文書をすべて不要文書とするため、全ての必要文書と不要文書を判別する決定木を作成すると、サンプル中の必要文書のみを得る検索式が作成される。ここでは決定木の深さがある程度深くなったところで文書集合の分割を停止させ、その段階で正例を得るパスを検索式作成に用いる。具体的には、サンプル中の正例と負例が区別された段階で集合の分割を停止して必要文書を得るパスとして扱うもの(以降 **Add1**)と、必要文書に登場する検索要求文中の検索語の全てを用いて決定木を作成し、必要文書を含む集合に至るパスを検索式作成に用いるもの(以降 **Add2**)の2種類について比較する(図3)。

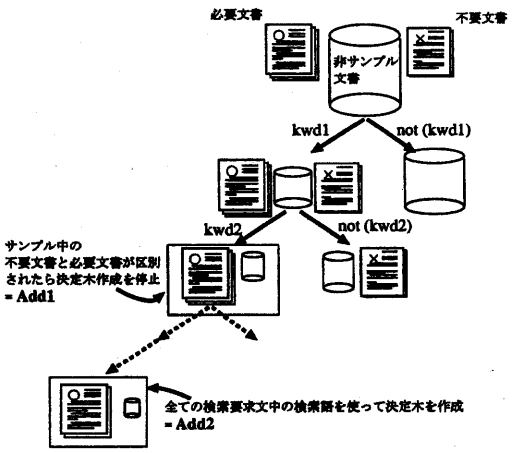

図 3: 決定木の枝刈り

4 実験

4.1 使用データ

検索精度の評価には、英文を対象とした文書検索テストセット NPL を用いた(表1)。テストセットは文書の集合と検索要求文からなり、各質問文に対して関連する文書(正解)が与えられている。テストセットからは FreeWAIS-sf [11] の不要語辞書に登場する語は除去し、また残った単語は Porter の stemming アルゴリズム [5] により語幹のみを取り出して利用した。

表 1: テストセット

文書数	文書総量(MB)	質問数	平均質問語数	平均正解数
11429	3.1	93	6.7	22.4

4.2 実験手順

実験に用いたシステムの構成を図4に示す。
実験手順を以下に示す。

1. 検索要求文から式(1)を用いて v_q を作成、各文書のベクトルとの内積を計算して各文書のスコアとする(通常の検索)。
2. スコア上位 $n(10, 30, 50)$ 件をサンプル文書とし、テストセットの正解を用いて正解(= 必要文書)と不正解(= 不要文書)を判定する。
3. Rocchio feedback により、各文書のスコアを再計算する。
4. 検索要求文と必要文書および不要文書を用いて検索式を作成する。作成方法を以下に挙げる。なお検索式作成には検索要求文中の検索語のみを用いた。
   - 検索要求文中の全ての検索語を AND で結合したもの(**Query_AND**)
   - 検索要求文中の全ての検索語を OR で結合したもの(**Query_OR**)
   - サンプル文書のみを学習例として ID3 により検索式を作成するもの(**ID3**)
   - サンプル文書のみを学習例として ID3 により検索式を作成するもの、ただし必要文書に登場する検索語のみ決定木作成に利用(**ID3+**)
   - サンプル文書以外の文書を不要文書として加える 3章の手法を用いて ID3 により検索式を作成するもの(**Add1**、**Add2**)

またシソーラスを決定木学習と併せて利用する方法として

- ID3 による決定木学習の際に Bit-per-category エンコーディングを利用(**Bit**)
- 決定木学習の結果に対し、シソーラスに登場するすべての語をシソーラス展開して利用[2](**Th1**)
- 決定木作成の際に、質問文中の検索語だけでなく検索語とシソーラスで同じ分類に属する語も利用(**Th2**)

[2]複数の分類に属する場合には展開していない。

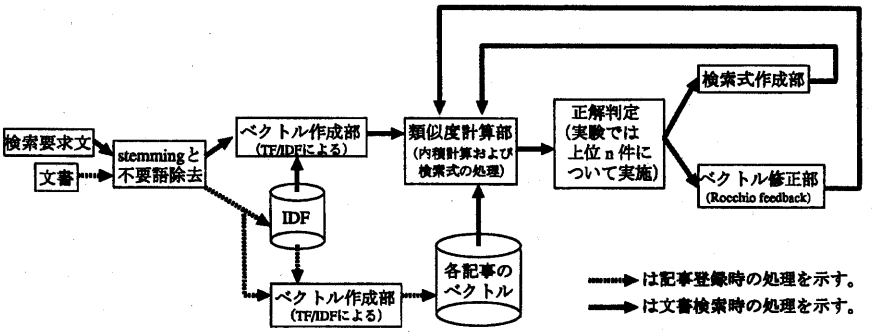

図 4: システム構成図

を用いる。

5. 得られた検索式に適合する文書について、3. で得た Rocchio feedback によるスコアを上昇させることで、共起を含む文書のスコアに反映させる[3]。

6. スコアの高い順に文書をソートし、検索結果が元の検索文に対する正解記事であれば正解として、再現率 $0, 10, 20, \cdots, 100\%$ を満たす時の適合率を求める。但し 2 で取り出したサンプル文書は評価対象から除く。

5 実験結果

Rocchio feedback、および提案手法による検索精度を示す (表 2)。表中の数字は再現率 $0, 10, \cdots, 100\%$ での適合率を平均したものであり、n は前章の実験手順 2 でのサンプル文書数である。

従来手法については、'Query' は検索要求文から作成したベクトルを用いた場合の検索精度 (前章の実験手順 1 に相当)、'Rocchio' は Rocchio feedback による検索精度を示す (実験手順 3 に相当)。'mod_Rocchio' は modified Rocchio feedback による検索精度を示す。

表 2: 適合率平均 (%)

手法	$n = 10$	$n = 30$	$n = 50$
Query	13.4	9.2	8.0
Rocchio	17.9	15.9	16.1
mod_Rocchio	18.4	15.2	15.7
Query_AND	18.7	16.5	16.4
Query_OR	18.2	16.0	16.2
ID3	18.8	19.8	17.7
ID3+	19.5	19.8	19.0
Add1	21.0	20.0	19.5
Add2	20.1	16.8	16.8

表 2から Query に比べ Rocchio が優れていることが確認できる。Rocchio と mod_Rocchio では、$n = 10$ では mod_Rocchio の方が優れているが、n が大きくなると逆転している。

決定木学習アルゴリズムを用いた ID3、ID3+、Add1、Add2 の 4 手法とも Rocchio、mod_Rocchio に比べ優れた精度を示しており、また全ての検索語を AND 結合した Query_AND 及び全ての検索語を OR 結合した Query_OR より優れた結果を示していることから、決定木学習によって適切な検索語を選択して AND 結合を作成した効果がわかる。

4 手法の中では 3章の提案手法を用いた Add1 が最良の結果を示している。Add1 と ID3+ を比較すると、$n = 30, 50$ では大差がないが、$n = 10$ では Add1 の方が優れた結果を示している。サンプル文書数 n が十分大きければサンプルのみを用いる ID3+ でも適当な検索式を学習可能であるが、サンプル数が少ない場合には仮想的な学習例を加えることでサンプルの不

[3] ここでは一律に類似度を 2 倍にした。この他にも定数を加える、検索式中の検索語の重みを加算する、検索式の重みを [11] にある手法などで評価して加える、などの手法を挙げることができ、適切な反映方法は今後の検討課題である。

足を補う Add1 が効果を発揮していると考えることができる。$n = 10$ における Add1 と ID3+ の精度を適合率－再現率グラフで比較したものを図5に示す。

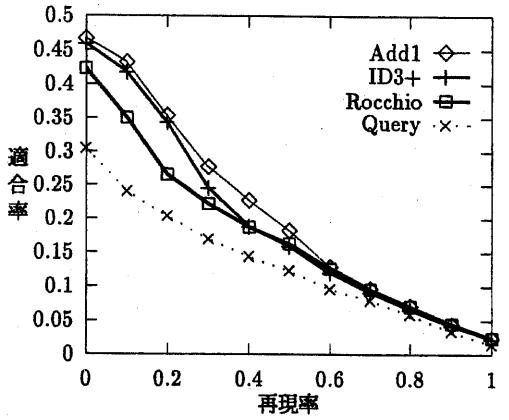

図 5: 疑似的に学習例を増加させる効果

なお、サンプル文書のみを学習例として用いる ID3+ と ID3 では、ID3+ の方がより優れた結果を示している。テストセットの検索要求文の中には検索語としての重要性が低い語があり、すべての検索語を決定木作成に用いる ID3 に比べ、必要文書のみに登場する検索語だけを用いる ID3+ は重要でない語を用いる可能性が小さいと考えられる。このことが ID3 より ID3+ の方が優れた検索精度を示す原因と考えられる。

決定木の作成をサンプル中の必要文書と不要文書が区別された段階で中止する Add1 に対し、必要文書中のすべての検索語を使って決定木を作成する Add2 は検索式に含まれる検索語が多いため Add1 に比べ少ない文書にヒットし、またヒットする文書は多くの検索語を含むことになる。多くの検索語を含む文書は Rocchio feedback によって高いスコアが与えられるため、サンプル文書数 n が大きい場合、その多くがサンプル中に登場する。そのため実験手順5でスコアを上昇させても、サンプル中の文書は精度評価に用いていないため効果が少ない。

次にシソーラスを用いた場合の結果を表3に示す。検索語を単純にシソーラス展開する Th1 および検索要求文中の語と同じ分類に属する語を決定木作成に用いる Th2 はシソーラスを用いない場合に比べ劣る結果を示している。一方 Bit-per-category エンコーディングを用いた Bit は ID3、ID3+、Add1 共に効果を示している。

Th2 がシソーラスを使わない場合に比べ劣ることから、決定木作成に用いる単語の種類を増加させるだけでは効果が得られないことがわかる。また Th1 の結果から、検索語を無条件にシソーラス展開すると逆効果であるものの、シソーラス展開する語を選択する Bit が効果を示していることから、展開する語によっては効果が得られることがわかる。

$n = 50$ における Add1,Bit と Add1 の精度を適合率－再現率グラフで比較したものを図6に示す。

表 3: シソーラスを用いた場合の適合率平均 (%)

手法	$n = 10$	$n = 30$	$n = 50$
ID3	18.8	19.8	17.7
ID3,Bit	19.7	20.9	17.9
ID3,Th1	18.5	18.9	16.5
ID3,Th2	17.3	17.5	17.0
ID3+	19.5	19.8	19.0
ID3+,Bit	20.2	20.0	19.2
ID3+,Th1	19.3	18.8	17.6
ID3+,Th2	18.1	17.8	16.5
Add1	21.0	20.0	19.5
Add1,Bit	21.6	20.5	20.2
Add1,Th1	20.9	18.8	18.0
Add1,Th2	19.5	19.2	18.7

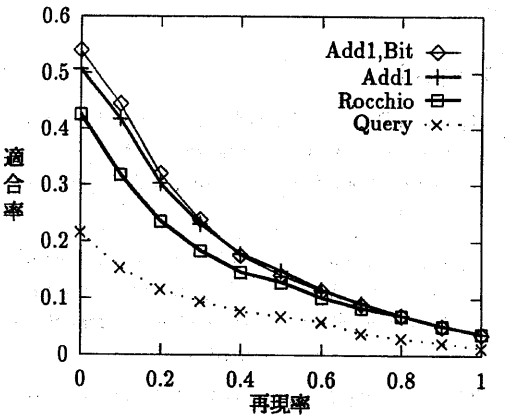

図 6: bit-per-category エンコーディングの効果

6 おわりに

決定木学習の手法を用いることで検索語間の共起関係を抽出し、Rocchio feedback の精度向上を図る手法を提案し、実験により効果を示した。検索者から

のフィードバック文書数が少ない場合の対策として、疑似的に負例を増加させ、学習例数の不足を補う方法を用いることで、より優れた効果が得られることを示した。またシソーラスにより単語を概念レベルに抽象化する Bit-per-category エンコーディングを用い、シソーラスが精度向上に役立つことを示した。

今回用いたテストコレクションの中には検索要求文中の検索語を含んでいても、文書中の主題となっていないために正解文書とならない場合が見られた。また決定木作成に用いる検索語を制限することで検索精度が向上する場合が見られた (5章で述べた ID3 と ID3+ の比較を参照)。提案手法では決定木作成に用いる検索語を選択する際に各文書内での単語の登場回数、出現位置などの情報は用いていないが、これらの情報は文書検索、および文書からのキーワード抽出に有効であり [12, 13]、これらの情報を用いて検索語を制限することで relevance feedback の精度向上にも効果が期待できる。

Buckley らは各単語の重みを別々に上下させることで学習例のランキングが変化するかテストし、ランキングの精度を向上させる単語について重みを上昇させる方法 (Dynamic Feedback Optimization,DFO) が Rocchio feedback の精度向上に効果があると報告している [2]。本稿の提案手法で得られた検索式は Rocchio feedback によるスコアの変更に用いるだけで、各単語の重みの修正には用いていないが、DFO を拡張して単語の組合せの重要度を変更する方法を検討中である。

参考文献

[1] Chris Buckley, Gerard Salton, and James Allan. Automatic routing and ad-hoc retrieval using SMART:TREC2. In *TREC-2*, pp. 45–55, 1994.

[2] Chris Buckley, Gerard Salton, and James Allan. The effect of adding relevance information in a relevance feedback environment. In *SIGIR*, pp. 292–300, 1994.

[3] Gerard Salton and Michael J.McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Advanced Computer Science Series. McGraw-Hill Publishing Company, 1983.

[4] Haussler, D. Quantifying inductive bias: AI learning algorithms and valiant's learning framework. *Artificial Intelligence*, Vol. Vol.26,No.2, pp. 177–211, 1988.

[5] Porter, M. F. An algorithm for suffix stripping. *Journal of the Society for Information Science*, No. Vol.14 No.3, pp. 130–137, 1980.

[6] Quinlan, J. R. Induction of decision trees. *Machine Learning*, pp. 81–106, 1986.

[7] Quinlan, J. R. *C4.5: Programs for machine learning.* Morgan Kaufman, 1993.

[8] Rocchio, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System*, pp. 313–323. Prentice-Hall, 1971.

[9] Roget, P.M. *Thesaurus of English words and phrases.* 1852.

[10] Amanda Spink. Term relevance feedback and query expansion: relation to design. In *SIGIR*, pp. 81–90, 1994.

[11] Ulrich Pfeifer and Tung Huynh. Freewais-sf, 1994.
ftp://ls6-www.infomatik.uni-dortmund.de/pub/wais/freeWAIS-sf-1.0.tgz.

[12] 原 正巳、中島 浩之、木谷強. 単語共起と語の部分一致を利用したキーワード抽出法の検討. 自然言語処理研究会資料 NL-106, 情報処理学会, 1995 年.

[13] 高木 徹、木谷強. 単語出現共起関係を用いた文書重要度付与の検討. 情報学基礎研究会資料 FI-41-8, 情報処理学会, 1996 年.

[14] フセイン・アルモアリム、秋葉 泰弘、金田重郎. 木構造属性を許容する決定木学習. 人工知能学会誌 Vol.12 No.3, 人工知能学会, 1997 年.

[15] 海野敏. 出現頻度情報に基づく単語重みづけの原理. *Library and Information Science*, pp. 67–87, 1988.

[16] 三輪眞木子. データベースサーチャの視点. 情報処理学会誌 Vol.33 No.10, 情報処理学会, 1992 年.