情 報 学 基 礎 45－2
（1997. 5. 15 ）

単語の文書頻度を利用した決定木学習アルゴリズムによるrelevance feedbackの高精度化

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

あらまし

relevance feedbackは、検索者にとって関心のある文書から抽出したキーワードを利用し、検索式を修正することで検索精度を向上させる手法である。その代表的なアルゴリズムであるRocchio feedback法は、種々の検索タスクで精度向上の効果が示されている。しかし、作成される質問ベクトルは論理積(AND)や否定(NOT)を完全には表現できず、検索者の意図を表現するには不十分である。ANDやNOTを用いて検索者の意図に適した検索式を作成する方法として、決定木学習アルゴリズムID3を利用する方法が提案されている。しかし、この方法も文書から検索語を選択する際に文書データベース内での単語の重要性を考慮しないため、検索者にとって重要でない語が検索語となる可能性があった。筆者は文書データベース全体の中で単語が登場する文書の数(単語の文書頻度)を利用することで、より重要な単語を検索語として選択するアルゴリズムを提案する。また提案手法の有効性を示すため、提案手法により作成した検索式をRocchio feedbackと融合し、これを情報検索システム評価用テストコレクションBMIR-J1により評価した結果を示す。

キーワード　情報検索、決定木学習、文書頻度、類似文書検索、適合帰還検索、関連フィードバック

Inductive Learning Using Document Frequency for Relevance Feedback

Hiroyuki Nakajima　　Tsuyoshi Kitani
Laboratory for Information Technology, NTT DATA

Abstract

Rocchio feedback is a method which modifies queries based on evaluated documents for relevance feedback. Although it is effective in identifying relevant documents, it doesn't utilize 'AND' and 'NOT' operators fully. Applying the ID3 inductive learning algorithm to this task, we can take advantage of queries including 'AND' and 'NOT'. But, it sometimes fails to select appropriate words for effecive queries, partly because it doesn't consider the importance of words. In this paper, we improve the inductive learning approach by using the document frequency of words to select query words. We apply generated queries by our method to the Rocchio feedback and empirically demonstrate the effectiveness of our method.

keywords　information retrieval, decision tree, document frequency, relevance feedback

1 はじめに

relevence feedback は、文書データベースの検索者が必要文書と判断した文書中の単語を用いて検索式を修正し、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検索者と協調して行なうものであり、relevance feedback は検索作業を補助する有効な手段と考えられている。

relevance feedback を実現する代表的なアルゴリズムに Rocchio feedback がある。Rocchio feedback は検索者が必要または不要の判定をした文書中の単語とその登場頻度を反映して検索式を変更する。Rocchio feedback は多くの検索タスクにおいて検索精度の向上に役立つと報告されているが、作成される質問ベクトルは論理積 (AND) や否定 (NOT) を完全には表現できない。

そこで、論理和以外にも AND や NOT を用いた、より複雑な検索式を検索者の指示した文書から推定することで、精度の高い relevance feedback を実現しようとする試みがある中でも Chen の提案する、決定木学習アルゴリズム ID3 を用いる手法は、検索者が必要または不要の判定をした文書数に比例する時間で処理が可能であること、コンパクトな検索式が得られるため検索処理が短時間で可能であるといった長所がある。しかし文書から検索語を選択する際に、文書データベース内での単語の重要性を考慮しないないため、検索者にとって重要でない語が検索語となる可能性があった。

検索語を選択する際に、Chen の手法では検索者の指示した必要文書と不要文書を区別する能力だけを用いているのに対し、筆者は文書データベース全体の中で単語が登場する文書数 (単語の文書頻度) を利用することで、単語の重要性を検索語の選択に反映させる手法を提案する。提案手法で選択される検索語は、必要文書と不要文書を区別できるだけでなく、文書データベース中で重要な単語となるため、Chen の手法と比較して検索者にとってより重要な語となる。

本報告では提案手法により作成した検索式を Rocchio feedback と融合する。これを情報検索システム評価用テストコレクション BMIR-J1 によって評価し、提案手法の有効性を示す。

2 既存手法による relevance feedback

2.1 Rocchio feedback

Rocchio feedback アルゴリズムはベクトル空間法 (Vector Space Model, VSM) と TF/IDF 法を用いた文書検索システムにおいて、relevance feedback を実現するアルゴリズムである [6, 2]。

ベクトル空間モデルは文書や文をベクトル空間上のベクトルとして表現する [4]。VSM における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、文書は文書中の単語の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。文書中に単語 t_i が出現する回数 (Term Frequency,TF) および単語 t_i が出現する文書数の逆数 (Inverted Document Frequency,IDF) を用いて単語 t_i の重み w_i を決定する [4]。TF/IDF 法は数多くのバリエーションが存在するが、ここでは最も標準的な式を示す [11, 2]。

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

ここで $f_{i,j}$ は文書 d_j 中に t_i が登場する回数、n_i は文書データベース中で t_i を含む文書数を表す。$|DB|$ 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式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。検索式のベクトルを v_q、提示した文書中から検索者が選んだ文書 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 は定数、$w_i < 0$ となる w_i は 0 とする)。検索式と文書の間の類似度は、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る。Rocchio feedback で作成されたベクトルと文書の間で類似度計算を行なう場合、個々の単語について独立に計算した類似度の和が文書との類似度になる。VSM による文書検索では、演算子 OR は類似度の和で表すと解釈されることが多く ([7] など)、Rocchio feedback の作成する検索式は単語を演算子 OR で結合していると考えることができる。しかし一般に検索者が検索式を作成する場合には、OR だけでなく AND や NOT といった演算子を用いており、演算子 OR だけでは検索者の意図を十分に反映できない。

2.2 決定木学習アルゴリズム ID3 による検索式作成

論理式を木構造で表現したものは決定木と呼ばれる。Chen は決定木学習アルゴリズムの一つである ID3 を用いることで、検索者が必要または不要を判定した文書から AND や NOT を含んだ検索式を作成し、relevance feedback を実現する方法を提案している [3]。

2.2.1 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最小の決定木を作成するアルゴリズムである[1][5]。アルゴリズムの概略を以下に示す。

1. 入力された正例と負例からなる集合を Set_0 とする。正例、負例はそれぞれ単語の集合を持つ。
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 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)。

$$I(t_j) = H - H(t_j) \tag{3}$$

正例と負例の 2 種類を分類する場合、

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

とすると、H と $H(t_j)$ は

$$H = h(p_i, n_i, s_i) \tag{4}$$

$$H(t_j) = \frac{s_i(t_j)}{s_i} h(p_i(t_j), n_i(t_j), s_i(t_j)) + \frac{s_i(\overline{t_j})}{s_i} h(p_i(\overline{t_j}), n_i(\overline{t_j}), s_i(\overline{t_j})) \tag{5}$$

で求めることができる。

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。

2.2.2 決定木学習を用いた relevance feedback

Chen は検索者が関心あるとした文書を正例、関心なしとした文書を負例として ID3 に与えて検索式を作成し、この検索式によって検索を行なうことで relevance feedback を実現した [3]。図 1に示すように、相互情報量の大きい単語で集合を分割することで決定木を作成し、正例を得るために用いた 単語を検索演算子 AND で結合したものを検索式として用いた。

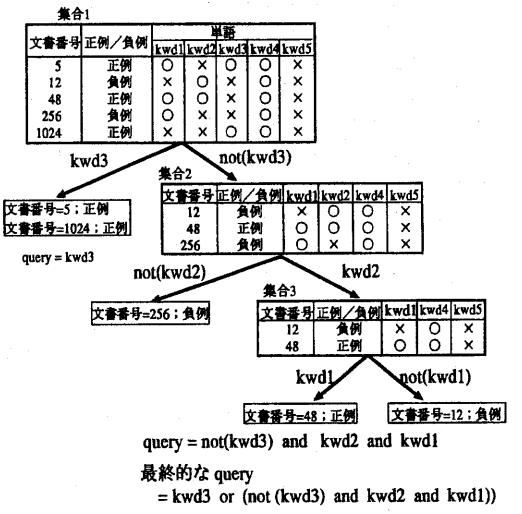

図 1: ID3 による検索式作成

Chen は ID3 だけでなく、ID5R を用いる手法も示しており、ID3 に比べ高い精度を得ている。ID5R は学習例が複数回に分けて与えられた場合、決定木をインクリメンタルに作成するよう ID3 を拡張した手法で、前回の学習で作成された決定木を保存したまま、決定木の葉をさらに分岐させる [8]。筆者が行なった予備実験では、質問文中の自立語のみを決定木の作成に用いて決定木を作成し、正例と負例の両方が含まれている葉について、文書のタイトル中の単語を候補として用いて決定木を作成した場合に良好な結果を得た。

3 単語の文書頻度を考慮した検索語の選択

正例と負例のみを ID3(ID5R) による決定木の作成に用いる Chen の手法では、集合分割に用いる単語を選択する際に文書データベース中での単語の重要性が反映されない。後述の実験において Chen の手法に作成した決定木には、

- 重要性の低い単語がたまたま正例と負例をよく区別できる (= $I(t_j)$ が大きい) 場合があり、検索者にとって重要でない語を選択してしまう
- 重要な検索語であっても、少数の正例、負例にしか含まれない場合は選択されない[2]

といった問題が見受けられた。

[1] 最小の決定木を作成するのは MDL 原理による。また最小決定木の作成は NP 完全問題であり、ID3 のアルゴリズムは近似解を求めるものである。

[2] 特にユーザの質問文中の語で TF/IDF 法による重みが大きいもので生じる

これらの問題を解決するため、データベース中のすべての文書を正例、負例、'不定'の文書(必要ないし不要の指示がされていない文書)の3つに分類する決定木を作成することにする。

正例と負例とを区別できる語であっても、文書データベースの多くの文書に登場する語の場合、検索語としての重要性は低いと考えられる。このような語では正例、負例とデータベース中の他の文書とをよく区別することができないため、文書を3つに分類することで検索語として選択されにくくなる。逆に少数の正例、負例にしか含まれない語でも、文書データベース中にあまり登場しない語であれば検索語として選択されやすくなり、Chenの手法で生じた問題が解決されると期待できる。

指定された文書、および文書データベース中の文書すべてを含む集合を Set_0 として、ID3での相互情報量 $I(t_j) = H - H(t_j)$ は集合 Set_i について、

$$\begin{aligned} u_i &= \mathrm{Set}_i\text{中の不定の文書の数} \\ s_i &= p_i + n_i + u_i \\ u_i(t_j) &= \mathrm{Set}_i\text{中の}t_j\text{を含む不定文書の数} \\ s_i(t_j) &= p_i(t_j) + n_i(t_j) + u_i(t_j) \\ u_i(\overline{t_j}) &= \mathrm{Set}_i\text{中の}t_j\text{を含まない不定文書の数} \\ s_i(\overline{t_j}) &= p_i(\overline{t_j}) + n_i(\overline{t_j}) + u_i(\overline{t_j}) \end{aligned}$$

$$\begin{aligned} &h(a,b,c,d) \\ &= -\{\frac{a}{d}\log_3(\frac{a}{d}) + \frac{b}{d}\log_3(\frac{b}{d}) + \frac{c}{d}\log_3(\frac{c}{d})\} \quad (6) \end{aligned}$$

とすると、H、$H(t_j)$ は

$$H = h(p_i, n_i, u_i, s_i) \quad (7)$$

$$\begin{aligned} H(t_j) &= \frac{s_i(t_j)}{s_i} h(p_i(t_j), n_i(t_j), u_i(t_j), s_i(t_j)) \\ &\quad + \frac{s_i(\overline{t_j})}{s_i} h(p_i(\overline{t_j}), n_i(\overline{t_j}), u_i(\overline{t_j}), s_i(\overline{t_j})) \quad (8) \end{aligned}$$

で求めることができる。

しかし上式を評価するためには検索者が必要または不要を判定した文書だけでなく、文書データベース全体をID3のアルゴリズムに従って分割していく必要がある。これは大規模な文書データベースを扱う場合、処理時間の点から現実的ではない。

文書データベース全体の中で単語 t_j が登場する文書数を $df(t_j)$ とする。ここでは不定文書中では単語間に相関がなく、いずれの分割対象の文書集合でも単語が登場する文書の比率が等しい、すなわち任意の t_j において、すべての分割対象の集合で

$$\frac{u_i(t_j)}{u_i} = \frac{u_0(t_j)}{u_0} = \frac{df(t_j) - p_0(t_j) - n_0(t_j)}{u_0} \quad (9)$$

がなり立つと仮定することで、文書データベース全体を分割することなく集合分割に用いる語 t_j を決定する。

準備

集合 Set_0 から Set_i を得るまでの分割に用いられる単語を $t_{i_1}, \cdots, t_{i_m}$、分割で生成される集合を $\mathrm{Set}_{i_0}, \cdots, \mathrm{Set}_{i_m}$ (ただし $\mathrm{Set}_{i_0} = \mathrm{Set}_0$、$\mathrm{Set}_{i_m} = \mathrm{Set}_i$) とする。なお、$0 < n \leq m$ である任意の整数 n について $\mathrm{Set}_{i_{n-1}}$ の分割に t_{i_n} が用いられて Set_{i_n} が得られるものとする。t_j についての関数 $f(t_j)$ を

$$f(t_j) = (I(t_j) - H)s_0 \quad (10)$$

とする。

関数 $f(t_j)$ の評価

式9より、集合 Set_{i_k} について

$$u_{i_k}(t_{i_{k+1}}) = \frac{u_0(t_{i_{k+1}})u_{i_k}}{u_0} \quad (11)$$

より

$$u_{i_{k+1}} = u_{i_k} df'(t_{i_{k+1}}) \quad (12)$$

となる。ただし $df'(t_{i_k})$ を

$$df'(t_{i_k}) = \begin{cases} \frac{u_0(t_{i_k})}{u_0} \\ (\mathrm{Set}_{i_k}\text{が}t_{i_k}\text{を含む場合}) \\ \frac{u_0 - u_0(t_{i_k})}{u_0} \\ (\mathrm{Set}_{i_k}\text{が}t_{i_k}\text{を含まない場合}) \end{cases} \quad (13)$$

とする。よって

$$u_i = u_0 \prod_{l=0}^{k} df'(t_{i_l}) \quad (14)$$

$$u_i(t_i) = u_0(t_i) \prod_{l=0}^{k} df'(t_{i_l}) \quad (15)$$

$$u_i(\overline{t_i}) = \{u_0 - u_0(t_i)\} \prod_{l=0}^{k} df'(t_{i_l}) \quad (16)$$

を得る。$f(t_j)$ に

$$\begin{aligned} s_i(t_j) &= u_i(t_j) + p_i(t_j) + n_i(t_j) \\ s_i(\overline{t_j}) &= u_i(\overline{t_j}) + p_i(\overline{t_j}) + n_i(\overline{t_j}) \\ u_0(t_j) &= df(t_j) - p_0(t_j) - n_0(t_j) \end{aligned}$$

および式15,16を代入することで $f(t_j)$ は文書データベース全体を分割することなく評価が可能となる。また H, s_0 は各集合において単語によらず一定であるから、$I(t_j)$ の大小比較は $f(t_j)$ の大小比較で代用できる。

4 実験方法

4.1 使用データ

実験に使用したデータは以下のとおりである。

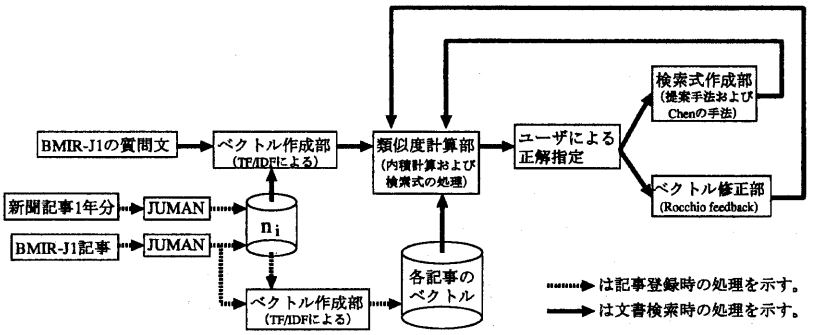

図 2: システム構成図

1. 情報検索テストセット BMIR-J1 [3]
   日本語の検索文 60 文と日本経済新聞 600 記事、および各検索文に適合する記事をランク (A,B,C の 3 段階) 付きで示している。本報告では各検索文に対して A,B のランクが付けられた記事を正解とし、奇数番の検索文をパラメータ決定用、偶数番の検索文を評価用として用いた。
2. 新聞記事 1 年分 (1993 年毎日新聞)
   上記テストセットの 600 記事と併せ、式 (1) で用いている単語の文書頻度 n_i の決定に利用した。
3. 上記の新聞記事と検索文から抽出したキーワード
   日本語形態素解析システム「茶筌 (ChaSen) 1.0 (JUMAN 2.0+)」 [9] を用い、長さ 4byte 以上の名詞、形容詞、動詞、未定義語 (数字を除く)) をキーワードとした。

4.2 実験手順

実験に用いたシステムの構成を図 2に示す。実験は検索者から指定される正例の文書数 (num_{rel}) を 3 に固定して行なった。

4.2.1 パラメータ決定

訓練用の検索文を用い、Rocchio feedback のパラメータ α、β、γ を以下の方法で決定した。

1. パラメータ決定用の検索文から自立語を取り出し、ベクトル v_q を作成する。
2. ベクトル v_q と BMIR-J1 の各新聞記事 600 件のベクトル間で類似度を計算する。
3. 得られた記事から最大の類似度を持つ正解記事 $num_{rel} = 3$件と、正解記事より上位にある不正解記事を取り出す
4. 定数 α は 16 に固定し、定数 β、γ を 2 から 128 の間で変化させて Rocchio feedback を実施する。
5. 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 (ただし Rocchio feedback に用いた文書は検索結果および正解から除外)。
6. すべての質問文について、各再現率での適合率の平均を求め、各再現率での適合率の和が最大になるように β、γ を決定する。

4.2.2 本実験

本実験では、評価用の検索文について以下の各方法で類似度を決定し、Rocchio feedback、Chen による手法、提案手法を比較した。正例には下記の **Query** で得られた類似度が最も高い正解記事 3 件を、負例には正例より大きな類似度を持つ不正解記事を用い、評価は評価用検索文 30 文のうち、負例を得た 22 文について行なった[4]。

Query 検索文中の自立語から作成したベクトル v_q により類似度を決定する。

Rocchio Rocchio feedback で作成したベクトル v により類似度を決定する。

Chen Chen の手法で作成した検索式に適合した文書について、**Rocchio** による類似度を 2 倍したものを最終的な類似度とする。

Proposal 提案手法で作成した検索式に適合した文書について、**Rocchio** による類似度を 2 倍したものを最終的な類似度とする。

本実験では検索式に適合した文書について、Rocchio feedback で得られた類似度を一定の割合[5]で上昇させ、検索精度の変化を調べることで、得られた検索式がユーザの検索意図を表しているか評価する。

[3] 株式会社日本経済新聞社の協力によって、社団法人情報処理学会・データベースシステム研究会・情報検索システム評価用データベース構築ワーキンググループが、1993 年 9 月 1 日から 12 月 31 日の日本経済新聞記事を基に構築した情報検索用データベース (テスト版) を利用

[4] $30 - 22 = 8$ 文は類似度が最も高い 3 文書がいずれも正解である。この 8 質問文は質問文中の自立語だけでも高い精度で検索でき、relevance feedback の必要性は低い。

[5] 2 倍とした (学習により決定した値ではない)。

5 実験結果

Query,Rocchio,Chen,Proposal の各方法による検索精度を図3に示す。Rocchio feedback と決定木学習アルゴリズムで作成した検索式を融合した Chen と Proposal は、いずれの再現率においても Rocchio と同等以上の適合率を示している。また Proposal は再現率 50% を除く全ての再現率において Chen の適合率を上回っており、Proposal による検索式により Chen に比べ高い適合率が得られることがわかる。

なお、BMIR-J1 の記事のうち Proposal による検索式に適合した記事は 1 質問当たり平均 23.2 記事、Chen による検索式に適合した記事は平均 42.5 記事 (学習例として与えた正解記事 3 件を除く) であったが、いずれも検索式に適合する文書だけでは Rocchio に比べ低い再現率となった。本実験の結果は得られた検索式を Rocchio feedback と融合して用いることで、relevance feedback の精度を向上させ得ることを示している。

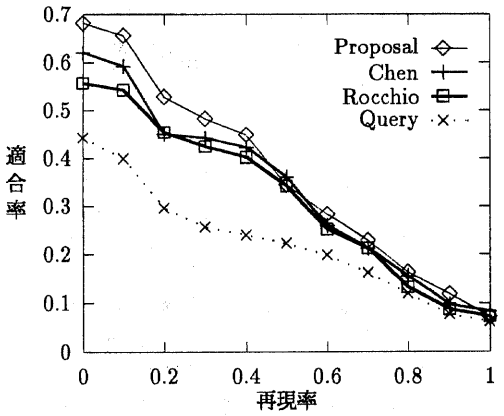

図 3: 従来手法と提案手法の比較

6 おわりに

relevance feedback における検索式の推定において、単語の文書頻度を検索語の選択に利用することで、より重要な語を検索語として選択するアルゴリズムを提案した。Rocchio feedback 法と融合した実験により、従来手法と比較して検索精度の向上に効果があることを示した。

提案手法では文書データベース全体の中での単語の出現文書数を用いたが、これは TF/IDF 法における IDF に相当する。TF に相当する各文書内での単語の登場回数を用いて、各文書における単語の重要性を利用することができれば、より精度の高い relevance feedback が可能になると考える。また、決定木学習の際に学習データをシソーラスによって一般化する手法が提案されており [1, 10]、その有効性を検証する予定である。

今回は検索者から指定される文書数を 3 として実験を行なったが、今後は学習例を増やして提案手法の有効性を検証する予定である。

利用したテストコレクションは対象文書数と質問数が少なく、統計的に有意な結果が得られたとは言えない。提案手法は言語依存性のある処理を行なっていないため、日本語以外の言語でも処理が可能である。英語のテストコレクションにより評価を実施することで、提案手法の有効性を検証する予定である。

参考文献

[1] Almuallim,H. Two methods for learning ALT-J/E translation rules from example and a semantic hierarchy. In *Proceeding of COLING94*, pp. 57–63, 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] Chen, H. Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms. *JASIS*, pp. 194–216, 1995.

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

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

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

[7] Ulrich Pfeifer,Tung Huynh. Freewais-sf, 9 月 1994 年. ftp://ls6-www.infomatik.uni-dortmund.de/pub/wais/freeWAIS-sf-1.0.tgz.

[8] Utgoff, P.E. Incremental induction of decision trees. *Machine Learning*, pp. 161–186, 1989.

[9] 松本 裕治 ほか. 茶筅 (chasen) 1.0 (juman 2.0+), 2 月 1997 年. http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html.

[10] 田中 英輝. シソーラスを利用した言語データ最適一般化アルゴリズム. 自然言語処理 Vol.108 No.14, 情報処理学会, 7 月 1995 年.

[11] 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.