

## 情報検索における検索語の選別の効果

田中 路子<sup>†</sup> 梅村 恭司<sup>†</sup>

<sup>†</sup> 豊橋技術科学大学 情報工学系  
441-8580 豊橋市天伯町雲雀ヶ丘 1-1  
TEL: (+81)532-47-0111(ex.5430)  
[michiko@ss.ics.tut.ac.jp](mailto:michiko@ss.ics.tut.ac.jp), [umemura@tutics.tut.ac.jp](mailto:umemura@tutics.tut.ac.jp)

Bigram 方式の情報検索システムは、キーワードによる情報検索システムに比べ検索対象となる語が多数になる。そのため、検索時間が増加する傾向にある。本研究では、Bigram の出現頻度を基に Bigram の選別を行うことで、情報検索システムの高速化を検討する。Bigram の出現頻度として、一般にはスコアに貢献する  $df$ (document frequency) を用いるが、本手法は  $df$  と相関が高く、しかも  $df$  に比べ高速に求められる  $tf$ (term frequency) を用いる。実験により、 $tf$  による Bigram の選別が、 $df$  による選別と同程度の精度を持ち、情報検索システムの高速化が図られていることを示す。

## Selecting Index Term for Information Retrieval

Michiko Tanaka<sup>†</sup> Kyoji Umemura<sup>†</sup>

<sup>†</sup> Dept. of Information and Computer Sciences, Toyohashi University of Technology  
Tempaku, Toyohashi, Aichi, 441-8580, Japan,  
TEL: (+81)532-47-0111(ex.5430)  
[michiko@ss.ics.tut.ac.jp](mailto:michiko@ss.ics.tut.ac.jp), [umemura@tutics.tut.ac.jp](mailto:umemura@tutics.tut.ac.jp)

There are more indexed terms in bigram information retrieval systems than keyword based systems. Therefore, the computation time of bigram systems tends to be longer than that of keyword based systems. This paper proposes a method to improve the efficiency of bigram information retrieval system by selecting bigrams from their frequency. Though it is natural to use  $df$  (document frequency), we propose to use  $tf$ (term frequency), which is highly correlated with  $df$ , and which is more efficient to calculate. The experiment shows that  $tf$  is almost as suitable as  $df$  to select bigram and is more suitable to make system efficient.

## 1. はじめに

ネットワークの発展に伴い、情報検索システムの高速化の要求が高まっている。多くの情報検索システムは、辞書を用いて選出したキーワードを基に情報検索を行っている。これに対し、我々の情報検索システムはドキュメントの意味を考慮せず文字列の出現頻度を基に情報検索を行っており、検索要求のドキュメントに対し、辞書を用いずに単純に二文字の組に区切った文字列を基に情報検索を行っている。以後、この単純に区切った二文字の組を Bigram と呼ぶ。具体的な情報検索の方法は、検索要求のドキュメントを Bigram に分割し、それを検索語として用いている。そして、各検索語に対しスコア付けし、各検索語を含むドキュメントごとにスコアの累積値を求める。最後に、累積値をドキュメントのスコアとし、そのスコア順に表示することにより情報検索の結果とする。本手法の特徴は、形態素解析を行わないことであり、これにより辞書に登録されていない新しい技術用語などの未知の用語に対応することができる。但し、我々が検討している情報検索は、あるドキュメントが与えられた際に、類似するドキュメントを選出することを想定している。

このような Bigram 方式の情報検索システムにおいて、我々は Bigram の様々な統計量の組合せを検討し、精度向上を行った<sup>1)</sup>。その結果、情報検索コンテスト (NTCIR Workshop1)<sup>2)</sup>において好成績を残したシステム<sup>3)</sup>と同程度の精度であることを示した。しかし、Bigram 方式の情報検索はキーワードによる情報検索に比べて、検索語の数が多数になるため検索時間が増加する問題がある。

そこで本稿では、これまで我々が検討してきた Bigram 方式の情報検索システムにおいて、Bigram の選別を行って高速化を試みる。その選別方法は、出現頻度に基づいて Bigram をソートし、その頻度情報を利用して、適切な数の Bigram のみを選出するアルゴリズムである。

本稿は、はじめに情報検索システムの原理について簡単に述べ、次に Bigram の選別による高速化について述べる。最後に、情報検索システム

の検索精度と検索時間について評価する。

## 2. 情報検索システムの概要

情報検索システムの流れを図 1 に示す。まず、ドキュメント集合と検索要求それぞれに対し前処理 (Preprocessing) を行う。その後、検索語である Bigram の選別 (Selection of Bigram) を行い、次に Bigram ごとに重み付け (Weighting) をする。最後にドキュメントごとに重み付けを行い、重みの大きいものから検索結果としてドキュメントを出力する。

ここで、本システムの概要を述べるにあたり、必要な特徴量を定義する。

- tf(term frequency) : ドキュメント集合全体に含まれる対象とする Bigram の出現頻度
- dtf(document term frequency) : 一つのドキュメント中に含まれる対象とする Bigram の出現頻度
- df(document frequency) : ドキュメント集合全体において対象とする Bigram が出現したドキュメントの出現頻度
- Burstiness : 出現集中を示す特徴量で、 $tf/df$  により求められる<sup>4)</sup>.
- idf(inverse document frequency) : 一般にスコアを求めるために用いられ、対象とする Bigram に対し、その Bigram がドキュメントを特定するための情報量を示す。従って、 $idf = -\log(df/N)$  により求められる。ここで、N はドキュメント数である。

### 2.1 前処理

我々が検討している情報検索システムの前処理は二つの処理から構成される。

一方の前処理は、検索対象となるドキュメント集合に対して Suffix Array<sup>5)</sup> を用いることである (図 1-(1))。Suffix Array のデータ構造は、全ての部分文字列の出現位置を特定できる表を作成する。この表の作成には、データの大きさを  $n$  とした場合、およそ  $n \log(n)$  の計算量を要する。しかし、一度表を作成すると、ドキュメント中にある文字列の出現位置の特定は、 $\log(n)$  の計算量で求められる。そのため、このデータ構造をインデックスに用いることで、出現頻度の計算は  $\log(n)$  で行える。

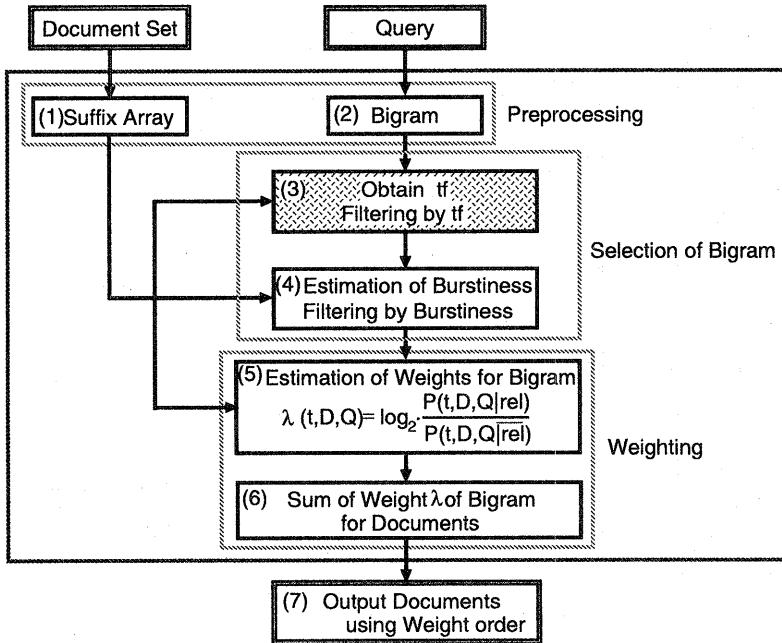


図1 情報検索システム

もう一方の前処理は、検索要求を Bigram に分割することである(図1-(2))。これは検索要求の文章を先頭から一文字ずつずらしながら、Bigram を作成する。この Bigram を情報検索の検索語として用いる。

## 2.2 Bigram 選出の原理

2.1 節で生成した Bigram 全てに対して、処理を行うことは時間の無駄になると考へ、関連がありそうな Bigram を Suffix Array を用いて選出する。選出は出現頻度 tf と出現集中 Burstiness を用いて二段階で行う。一つ目の tf による選別は、高速化のための選別であり、本稿で提案する手法である(図1-(3))。詳細は、3 節に記述する。

二つ目の Burstiness による選別は精度向上のための選別である。この選別は一つ目の tf による条件を満たす Bigram に対し、Burstiness の推定値を用いて行う。すなわち、Burstiness の推定値がある閾値よりも大きい Bigram を選出する(図1-(4))。この閾値は、NTCIR Workshop1 のデータより経験的に求めた(図2-(1))。この Burstiness による選別は、ある Bigram の出現が集中

するドキュメントは、その Bigram との関連が深いという考えに基づいている。

## 2.3 重み付けの原理

重み付けは、まず選出された Bigram ごとに重みを求めて、各ドキュメントに対し、この選出された Bigram の重みの累積値を計算する。この累積値がドキュメントの重みとなる。

### 2.3.1 Bigram の重み

選別された Bigram に対し重み付けを行う。Bigram に対する重み付けは確率モデルから実証的に求めている<sup>1)</sup>。重みは次式に示す log odds<sup>6)</sup> より推定する。

$$\log_2 O(D, Q) = \log_2 \frac{P(rel|D, Q)}{P(\bar{rel}|D, Q)} \quad (1)$$

上式の各記号は、D:選出された Bigram を含むドキュメント、Q:検索要求である。また、rel は D と Q が関連するという事象を表し、 $\bar{rel}$  は逆に、関連しないという事象を表している。式(1)はベイズの決定則を用いることで、次式のように表現できる。

$$\log_2 O(D, Q) = \log_2 \frac{P(D, Q|rel)}{P(D, Q|\bar{rel})} \cdot \frac{P(rel)}{P(\bar{rel})} \quad (2)$$

```

    idf = (log((float) total) - log((float) df)) / log 2; } (1)
    tf-threshold = 1.834407 + (-0.04802984) * idf;
    if((float) tf / ((float) df) > tf-threshold) {
        p0fit = -0.61384461 + idf * 0.01652994;
        p1fit = -0.80924267 + idf * 0.78607417;
        p2fit = -0.05118307 + idf * 0.78646177;
        p3fit = -0.23198404 + idf * 0.81838589;
        p4fit = -0.75251594 + idf * 0.83374752;
    }else{
        p0fit = p1fit = p2fit = p3fit = p4fit = 0.0;
    }
} 
```

図 2 精度向上のための閾値と学習から推定した重み

式(2)の  $P(\text{rel})/P(\overline{\text{rel}})$  は、ドキュメント  $D$  を引数に持たないため、ドキュメントのランク付けの効果に関して無視することができる。また、式(2)の  $\log_2 P(D, Q|\text{rel})/P(D, Q|\overline{\text{rel}})$  は次式のように log odds の和で分解して近似する。ここで、選出された Bigram を  $t$  とする。

$$\log_2 \frac{P(D, Q|\text{rel})}{P(D, Q|\overline{\text{rel}})} \approx \sum_{t \in T} \log_2 \frac{P(t, D, Q|\text{rel})}{P(t, D, Q|\overline{\text{rel}})} \quad (3)$$

このような Bigram ごとの log odds を、Bigram の重み入とする(図 1-(5))。すなわち、選出された各 Bigram に対し、その Bigram をどれだけ考慮するかを示す重み入を次式より求める。

$$\lambda(t, D, Q) = \log_2 \frac{P(t, D, Q|\text{rel})}{P(t, D, Q|\overline{\text{rel}})} \quad (4)$$

理論上は式(4)が重みになるのだが、実際に Bigram の重みを計算する際には、どのドキュメントが関連がある( $\text{rel}$ )か、関連がない( $\overline{\text{rel}}$ )かを特定することはできない。従って、式(4)の  $\lambda$  を直接計算することはできないが、 $\lambda$  をドキュメント中の出現頻度  $dtf$  やドキュメントを特定するための情報量  $idf$  などの観測できる特徴量の関数と考えて、重み入の推定値を得ることができる。

このような学習データから重みを推定した結果を図 2-(2)に示す。なお、図 2-(2)における  $p0fit$  は  $dtf=1$  の重みを表し、 $p1fit$  は  $dtf=2$  の重みを表す。以下同様に、 $p2fit$  は  $dtf=3$ 、 $p3fit$  は  $dtf=4$ 、 $p4fit$  は  $dtf=5$  の重みである。

### 2.3.2 ドキュメントの重み

2.3.1 節で求めた Bigram の重み入を基に、ドキュメントの重み付けを行う。各ドキュメントに関して、ドキュメントと検索要求の両方に

現し、しかも 2.2 節のアルゴリズムによって選別された全ての Bigram の重み入の累積値を求める。この累積値がドキュメントの重みになる。(図 1-(6))。このようにして求められたドキュメントごとの重みの大きいものから、最も検索要求に関連のあるドキュメントとして出力する(図 1-(7))。

### 3. Bigram の出現頻度を利用した高速化の原理

本稿で提案する高速化は Bigram の出現頻度を用いて検索語の選別を行うことである(図 1-(3))。選別の手法としては、Bigram を出現頻度  $tf$  の順に並べて、小さい方の  $x$  個を選出する。ここで、選別に用いる出現頻度として  $tf$  を用いる点が提案法の特徴である。なお、 $x$  はドキュメント中の考慮する Bigram の個数を示すパラメータ値である。この  $tf$  による選出は、出現回数が少ない文字列の方が情報検索に有利だという性質に基づいている。出現頻度を示す  $tf$  は、ドキュメント集合全体における Bigram の出現頻度を示し、Suffix Array のあらゆる文字列の位置情報を保持する性質より求めることができる。

選別を行う場合、スコアの値が大きいものを選別するというのが自然な方法である。一般に、スコアはドキュメントを特定するための情報量  $idf$  によって求められる。さらに、 $idf$  はドキュメントの出現頻度  $df$  から求める。すなわち、選別における Bigram の出現頻度としては、一般に  $df$  が用いられると考えるが、 $df$  は文章中の重複を判定する必要があるため、出現頻度の多いものに関しては時間がかかる。そこで、我々は  $df$  の代わりに  $tf$  を用いる高速化を提案する。 $tf$  は  $df$  と大きな相関があり、頻繁に出現する文字列でも高速に数え上げることができる。

### 4. 情報検索システムの評価

評価として、検索精度と検索時間について検討する。情報検索の精度においては、まず Bigram の選別を行う有用性を示し、次に、選別を行う出現頻度として  $tf$  を用いた選別と  $df$  を用いた選別の検索精度を比較する。また、検索時間においても同様に、 $tf$  を用いた選別と  $df$  を用いた選別を比較する。検索データは、33 万件のドキュメ

<検索課題 q=0031>

<タイトル>  
データ品質制御  
</タイトル>

<検索要求>  
マルチキャスト通信における関連する複数データの品質制御手法について論じたものはないか。  
</検索要求>

<検索要求説明>  
一度に複数の受信者にデータを送るマルチキャスト通信では異なる環境において複数の受信者が接続するので、従来の通信者制御や受信データの割合を考慮する送信者制御や転送データ制御等は役立たない。そこで、マルチキャスト環境におけるデータ転送データ割合制御手法が必要となる。マルチキャスト環境におけるデータ転送データ割合制御手法から、特にマルチメディアデータつまり、音声と動画のように相互に関連性のある複数データの取り扱いについて述べた論文が欲しい。相互に関連性のある複数データの取り扱いについて述べていないとしても、マルチキャスト環境におけるデータ転送データ割合について述べているものががあれば、部分的には要求を満たす。新しい研究を始めるにあたり最新の研究動向を知りたい。

</検索要求説明>

<概念>

- <J.CONCEPT>
  - a. マルチキャスト,
  - b. フロー制御,
  - c. マルチメディア
- </J.CONCEPT>
- <E.CONCEPT>
  - a. Multicast,
  - b. Rate Control, Flow Control,
  - c. Multimedia,
  - d. Quality Of Service
- </E.CONCEPT>
- <A.CONCEPT>
- d. QOS
- </A.CONCEPT>

</概念>

<分野>

- 1. 電子・情報・制御

</分野>

</検索課題>

図 3 情報検索における検索課題の例

表 1 情報検索コンテスト(短い検索文)との精度比較

System	11pt Average Precision
提案法	0.3254
BKJIBIDS	0.2715

ント集合に対し、図 3 に示すような検索課題 53 件からなる NTCIR Workshop1 のデータを用いた。なお、図 3 の検索課題の中で、本システムは検索要求の部分のみを用いた短い検索文による検索と、図 3 の検索要求説明の部分のみを用いた長い検索文による検索の二種類について評価を行う。

#### 4.1 検索精度に関する評価

情報検索に用いた Bigram の個数と、その際の検索精度の関係を図 4、図 5 に示す。図 4 は短い検索文に対する検索精度を示しており、図 5 は長い検索文に対する検索精度を示す。これらの図中に示す一点鎖線は、各々が用いた検索文の平均長を示す。これらの図より、検索精度が検索文の平均長よりも以前に飽和していることが分かる。このことは、検索文の一部分を用いて情報検索を行えば十分であることを示唆しており、

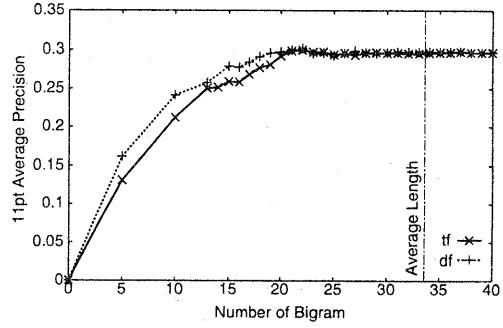


図 4 短い検索文に対する検索精度

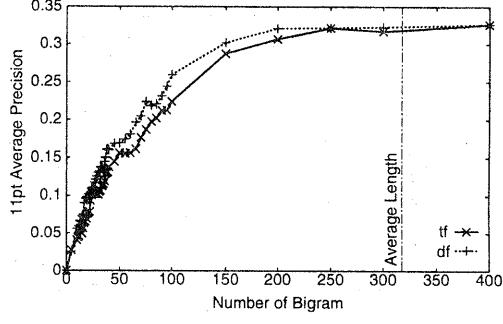


図 5 長い検索文に対する検索精度

検討する Bigram の個数を制限することにより、高速化が図れることを意味している。

次に、選別に用いる出現頻度に関しては、図 4、図 5 より、同じ個数の Bigram を用いた場合、検索精度が飽和するまでは tf に比べ、df の方が多少精度良く情報検索が行えている。しかし、精度が飽和してしまうと、tf も df も同等の精度で情報検索が行えることが分かる。

また、NTCIR Workshop1 において本手法と同様に Bigram 方式を用いた情報検索システムで、他のコンテストでも実績のある Berkley の情報検索システム BKJIBIDS<sup>3)</sup> の検索精度と本手法による検索精度を比較した結果を表 1 に示す。Berkley のシステムは、同条件における長い検索文に対する評価を行っていないので、表 1 では短い検索文に関する評価のみを示す。表 1 より、本手法は十分な精度を持つシステムであると言える。

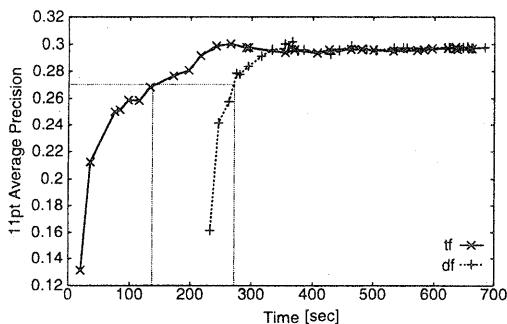


図 6 短い検索文に対する検索時間

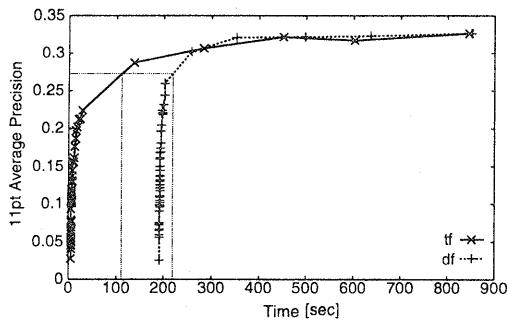


図 7 長い検索文に対する検索時間

#### 4.2 検索時間に関する評価

df と tf における検索時間と検索精度の関係を図 6、図 7 に示す。図 6 は短い検索文における比較を示し、図 7 に長い検索文における比較を示す。これらの図において、表 1 で示した Berkley の情報検索システム BKJIBIDS の検索精度を目標精度として考える。Berkley の情報検索システム BKJIBIDS の検索時間は公表されていないため、我々が実現したシステムにおける tf と df の選別について、53 件の検索文に対する検索時間を比較した。この目標精度を得るために要した時間は、短い検索文の場合は約 130 秒、また長い検索文の場合は約 110 秒、df に比べ tf を用いた方が高速に情報検索が行えていることが分かる。すなわち、検索要求の長短に関わらず、df と tf が同程度の検索精度を得るために要する時間は df に比べて tf を用いる方が高速に検索できることが分かる。

一般に精度の高いシステムにおいては、高速化を行うと精度が低下する場合が多い。しかし、本システムの提案する高速化の手法は、精度の高いシステムにおいても効果があることが図 6、図 7 より分かる。

#### 5. おわりに

本稿では、Bigram を用いた検索において検索時間が増加する問題に対し、出現頻度である tf を用いた Bigram の選別を行うことによる高速化手法を提案した。実験により df と tf において検索精度と検索時間の比較を行い、目標とする検索精度を得るために要する時間は、短い検索文において約 130 秒、df に比べ tf の方が速く情報検索が行えることを示した。このことより、tf による選別は df による選別と同等の精度で検索でき、しかも情報検索が高速に行えることを示した。

**謝辞** この研究は住友電気工業株式会社との共同研究の成果を利用して頂きました。深く感謝いたします。

#### 参考文献

- 1) Kyoji Umemura and Kenneth W. Church.: Empirical Term Weighting and Expansion Frequency, ACL EMNL 2000 Workshop, Hong Kong, pp. 117-123 (2000).
- 2) Noriko Kando, Kazuko Kuriyama, Toshihiko Nozue, Koji Eguchi, Hiroyuki Kato and Souichiro Hidaka.: Overview of IR Tasks at the First NTCIR Workshop, NTCIR Workshop, vol. 1, pp. 11-44 (1999).
- 3) Aitao Chen, Fredric C. Gey, Kazuaki Kishida, Hailing Jiang and Qun Liang.: Comparing Multiple Methods for Japanese and Japanese-English Text Retrieval, NTCIR Workshop, vol. 1, pp. 49-58 (1999).
- 4) Kenneth W. Church and William A. Gale.: Poisson mixtures, Journal of Natural Language, pp. 163-190 (1995).
- 5) Manber Udi and Gene Myers.: Suffix arrays : A new method for on-line string searches, SIAM Journal on Computing, pp. 935-948 (1993).
- 6) Christopher D. Manning and Hinrich Schütze.: *Foundations of Statistical Natural Language Processing*, THE MIT PRESS, p. 551 (1999).