卒業論文

# 著名人の発言に基づいた問題解決アドバイスシステムの設計と実装

氏名：藤原祥太
学籍番号：2260070107-3
指導教員：山崎勝弘
提出日：平成 23 年 2 月 17 日
立命館大学　理工学部　電子情報デザイン学科

**内容梗概**

本論文では、著名人の発言をデータベース化し、利用者の問題解決に役立つアドバイスを提供するシステムの設計と実装を行っている。

近年ではインターネットなどの検索技術の発展により、利用者が膨大な情報量の中から必要となる情報を容易に受信・閲覧できるようになっている。本研究では、現代社会に生きる人々の潜在的に抱える悩みの問題解決に対して、日本に数多く存在する偉業を成し遂げた著名人の発言が有効でないかと注目した。

そして取り上げた著名人の書籍を基に発言をまとめたデータベースから、利用者の悩みに対する回答として的確なアドバイスを提示することができるよう、情報検索やマイニングの基礎的な技術的方法論を記し、システムに実装する。加えて、利用者の問題解決に対して著名人の発言は役に立ったかどうかという実験を行いシステムを評価し、今後のシステムに対する改善点を述べる。

# 目次

１．はじめに ..... 1
２．問題解決アドバイスシステムの設計 ..... 3
2.1　システムの外部仕様 ..... 3
2.2　データベースの対象著名人 ..... 4
2.3　テキストマイニング ..... 4
2.4　形態素解析 ..... 5
2.5　検索手法とインデックス（索引）の作成 ..... 5
３．問題解決アドバイスシステムの実装 ..... 8
3.1　問題解決アドバイスシステムの内部仕様 ..... 8
3.2　データベースの内部仕様 ..... 9
3.3　データベースとインデックスの相関関係 ..... 10
3.4　表示・検索方法 ..... 11
４．実験と評価 ..... 15
4.1　結果表示例 ..... 15
4.2　利用者の評価 ..... 16
4.3　逐次検索と索引検索の照合回数の比較 ..... 19
５．考察 ..... 20
5.1　システム全体の評価 ..... 20
5.2　システムの改善点 ..... 20
5.3　並列化に向けて ..... 21
６．おわりに ..... 22
謝辞 ..... 23
参考文献 ..... 24

# 図目次

図 1. システムの外部仕様.....3
図 2. テキストマイニングの概念.....5
図 3. 逐次検索の概念.....6
図 4. インデックス(索引)の作成の流れ.....7
図 5. システムの内部仕様.....8
図 6. データベースとインデックスの相関関係.....10
図 7. 文書データに ID をつける流れ.....14
図 8. PAS の実行画面と検索結果.....15

# 表目次

表 1. データベースの内部仕様.....9
表 2. テキストから作成した索引ファイル.....13
表 3. データベース内のテキストの ID.....13
表 4. 実験結果.....16
表 5. 複数ヒットした結果の比較.....18
表 6. 照合回数の比較.....19

# 1. はじめに

現代の日本において、統合失調症や適応障害などの精神疾患は、特別なものではなく、誰にでも起こりうる疾患となった。平成８年には２１８万人だった我が国における精神疾患の患者数が、平成２０年には３２３万人へと急増し、うつ病など気分障害の患者数は１００万人を超えている。また、我が国の自殺率は先進７か国の中で最も高く、平成１０年以来１２年連続で３万人を超える深刻な事態となっている。１５歳から３４歳までの若い世代の死因で自殺がトップなのは日本のみである。これらは、現代のストレス社会が原因であると考えられる。精神疾患患者以外にも多くの人々が日々の生活に悩みや問題を抱えていることは容易に想像でき、彼らは自らが抱える問題を解決するきっかけを求めているのではないかと考えられる。このような社会的問題を抱えた我々に、何らかの変化をもたらしてくれる人々が存在する。例えば、松下電器工業株式会社創業者である松下幸之助氏やメジャーリーガーのイチロー選手だ。彼らの偉業は、日本全国に明るいニュースをもたらし、多くの人々に勇気を与えてくれる。そして、偉業を達成した著名人の発言は、人々の感動や共感を呼び、個人が抱える悩みの問題解決に役立つのではないかと私は考えた。

本研究は、近代の日本において偉業を成し遂げた数々の著名人の発言をデータベース化し、日々直面する問題を解決するために、利用者の抱える悩みに対して適切なアドバイスを提供することができる“問題解決アドバイスシステム（以下 Problem-solving Advice System：PAS と称す)”の開発を目的とする。具体的には、先述したイチロー選手や松下幸之助氏をはじめとし、スポーツ界・経済界に精通する著名人の発言に注目し、データベースを構築した。また、システム構成としては、著名人の発言を収録したデータベース、データベースからアドバイスを検索するための検索システム部、そしてユーザーにアドバイスを提供する結果表示部である。

また、設計にあたって、我が国の近年の WEB 技術の発展の背景を考えてみると、現代社会においては、計算機ハードウェアの高性能化、記憶媒体の大容量・低価格化により蓄積されるデータ量は日々増大の傾向に至る。それらによって、各個人が瞬時に多種多様な情報を受信・閲覧できる時代が到来し、情報社会が取り巻く環境の中で、我々の生活様式は変化しつつある。インターネットを利用することで、数々の著名人の発言やインタービューを受信・閲覧することは可能であるが、信頼性に欠ける情報や古い情報が多いことも事実であり、必ずしも利用者の期待に応えられるものではないと判断する。

この問題点から、インターネットから情報を抽出し有効に活用することは容易ではなく、情報飽和社会における膨大な情報から有益なアドバイスを取得する際には、著名人の発言をまとめた書籍が、信頼性という観点において、有益なものであると考える。以上の理由から、本システムでは、書籍に収録されている発言を実際に設計するシステムのデータベースに利用する。注目した著名人の書籍から発言を収録したデータベースを制作するにあ

たって、利用者の期待に応じるアドバイスを提供するために、現在の WEB 技術で広く用いられている検索技術や、データベース内の発言を解析するマイニング技術をシステムの実装に取り入れた。

本論文の構成は次の通りである。第 2 章では PAS の設計について、必要となるシステム構成やマイニング、検索手法の基礎的知識について述べ、第 3 章では PAS の実装に必要なデータベースの内部仕様や、検索アルゴリズムついて述べ実装を行う。第 4 章では、PAS の表示結果例や、実際に利用者に活用してもらう実験によってシステムの評価を行っている。また、全文検索の代表的な手法である逐次検索と索引検索の性能評価について述べる。第 5 章では、システムの改善点について述べ、データベースの処理や検索を高速で行う方法論について検討する。

# 2. 問題解決アドバイスシステムの設計

本章では、PAS の設計にあたり、システムの外部仕様と必要なテキストマイニングの知識や実装する検索手法について示す。

## 2.1 システムの外部仕様

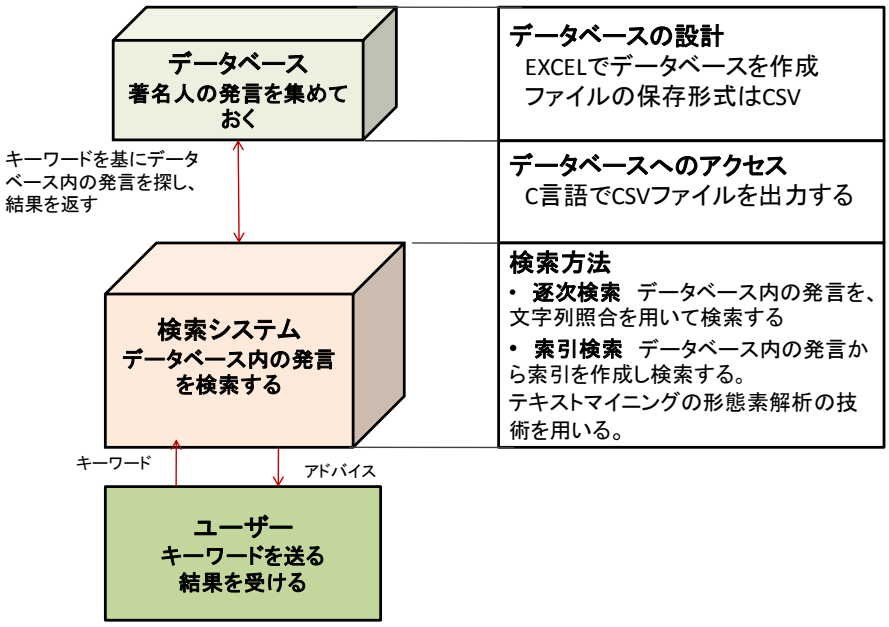


図 1. システムの外部仕様

図１にシステム構成を示す。システム構成における段階は大きく３段階に分けられる。

１．データベースの作成

２．データベースへのアクセス部分

３．検索システム部分

データベースの作成部分では著名人の発言を、エクセルを用いてデータベース化している。保存形式は、csv ファイル形式とした。データベースへのアクセス部分で、エクセルでデータベース化した発言集のファイルを読み込み、検索システム部分で入力したキーワードを文字列照合で探索できるようにする。検索システム部分では、利用者が、悩みとしているキーワードを入力し探索を行い、データベース内の発言をアドバイスとして結果に表示する。また、後述する逐次検索を行うための文字列照合のアルゴリズムと、索引検索を行うためのアルゴリズムについて実装を行った。

### 2.2　データベースの対象著名人

今回のアドバイスシステムで対象とする著名人は、アメリカ・メジャーリーグのシアトルマリナーズで日本人初の外野手としてプレイするイチロー選手、松下電器産業株式会社創業者（現・パナソニック株式会社）である松下幸之助氏、本田技研工業株式会社創業者である本田宗一郎氏、元サッカー日本代表であり、現役時代には日本人として欧州サッカーリーグで活躍した中田英寿氏である。

以上の４名を対象とする理由は以下の３点である。

１．日本全国の方がその存在と活躍を認識されている。

２．世間に多くの発言を発信し影響を与えている。

３．発言をまとめた書籍が出回っており、信頼性がある。

インターネットを用いることにより、上記 4 名の発言を閲覧することは可能であるが、個人運営の HP などが多く見られ､情報の信頼性という点において確かであると言い難い。よって、著名人の書籍[1]～[6]より発言をデータベースに収録する。

### 2.3　テキストマイニング

テキストマイニングとは、定式化されていない文書（以後、自然文書と称す）から一定の知見や発想を得るデータマイニング技術の一種である。例えば、アンケートの自由回答や掲示板への書き込みを解析することで、顧客や市場のニーズを抽出したり、自社製品への改善点を発見するなどの利用方法が挙げられる。

テキストマイニングを支える技術には

１．自然言語処理技術

２．データマイニング技術

３．可視化技術

の３点である。

自然言語処理技術では、文書から単語や語句などの要素を抽出する。また語句間の関係を１つの要素として抽出する場合もある。例えば「テキストマイニングの概念」という文書から「テキスト」、「マイニング」、「テキストマイニング」、「の」、「概念」という単語、語句を抽出する。

データマイニング技術では自然言語処理技術で抽出した要素間から、文書データ全体の傾向や特徴を発見する。また各要素間の関連性を分析することにより、新たな知見を発見する。

可視化技術では、ネットワーク図や散布図を用いて、人間にとって理解しやすい形で分析結果を提示する。本研究では、著名人の多くの発言を解析し、その関連性を発見し、利用者にとって有益となる発言を提供する。図２にテキストマイニングの流れを示す。

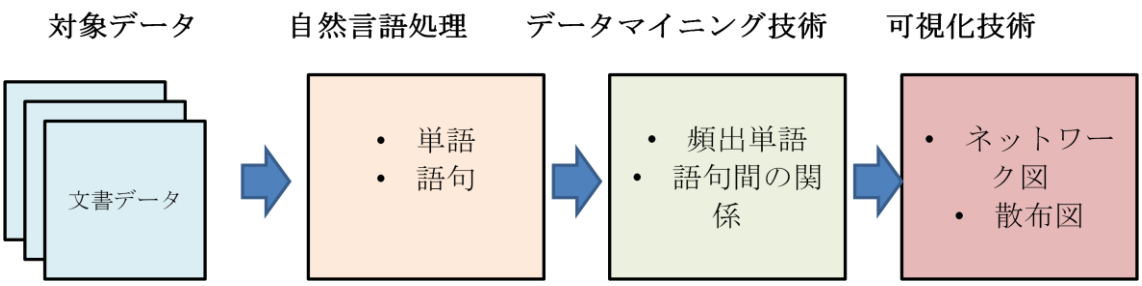


図 2. テキストマイニングの概念

### 2.4 形態素解析

多くの場合、検索語として用いられるものは単語である。したがって文書から検索語を抽出するには、まず文書を構成する文字列のどの部分が単語であるかを同定する必要がある。単語の同定は、元の文書が記述されている言語により処理が大きく異なる。英語などの文書は、単語が空白で区切られている場合、単語の同定は極めて容易である。しかし、日本語の場合、単語間に空白を入れて書く（分かち書きをする）習慣がないため、単語を同定することが極めて困難となる。そこで日本語の場合、文書から単語を正確に抽出するために形態素解析が必要となる。

形態素とは、それ以上に分割ができない語句の単位であり、形態素解析とは、文書を構成する文字列を単語に分割し、その語形変化を解析する処理である。辞書との照合に基づき形態素解析を行う手法や、大規模な文書データベースから文字や単語の連鎖確立を求め、これらの確率に基づき形態素解析を行う確率的手法が開発されている。現在の技術水準では、おおよそ９６～９８％の精度を達成している。

### 2.5 検索手法とインデックス（索引）の作成

主な検索手法は、全文検索と内容型検索に分けることができる。全文検索は、文書中から検索質問として与えられた文字列と一致する部分を探し出すことを目的とし、検索質問文字列を含む文書と、その文書中での出現位置が検索結果となる。

また全文検索は文書内の文字を１文字ずつ照合する逐次検索と、事前に文書から作成された索引を用いる索引検索の２種類に分類することができる。本研究で実装するシステムでは、利用者の要求に応える最適なアドバイスを提供するために、逐次検索と索引検索の両方の手法を採用する。

逐次検索では検索要求が起こるたびに、検索対象文書と検索単語との文字列照合を直接行うので、大規模な文書データベースに対しては時間的効率面から不向きである。しかしこの手法は、前処理が不要であり、内容が動的に変化する文書にも対応可能であるという利点がある。図３は逐次検索の流れを示している。

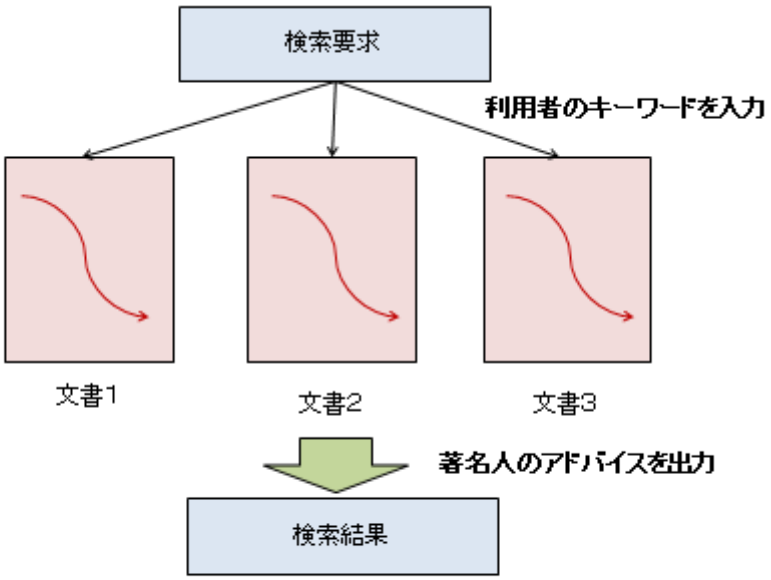


**図 3. 逐次検索の概念**

一方、索引検索は文書から索引を作成する前処理が必要となる。索引は、文書内のどの位置にどの単語が現れたかという情報が検索しやすい形で格納される。検索時には、文書に対して直接照合を行うことなく、索引に対してのみ検索を行うので、大規模な文書データベースに対しても高速な検索を行える利点がある。一般に扱う文書データベースが大規模になれば索引の記憶容量も増加し、索引の構築時間も増加する。しかしながら、インターネットの検索のようなリアルタイム性が要求される場合には、前処理によるデメリットよりも高速な検索機能のメリットが尊重されている。

索引検索の流れについて以下の図４に示す。

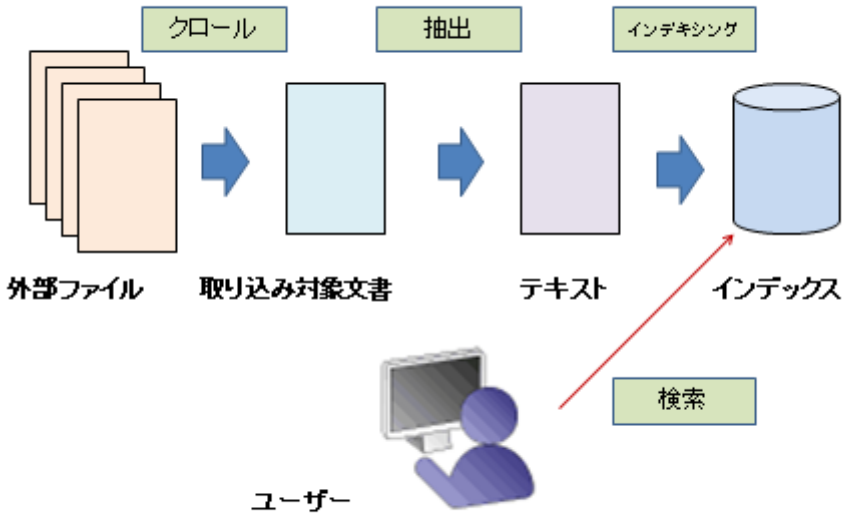


図 4. インデックス(索引)の作成の流れ

索引検索では、「クロール」「抽出」「インデキシング」と呼ばれる 3 つのステップで構成されるフローに基づき、検索対象文書のインデックスが作成され、検索が可能となる。具体的な処理の流れとしては、まずクロール処理が外部ファイルからインデックスに取り込むべき文書を判断し、その文書を抽出処理に引き継ぐ。

次に抽出処理は、一太郎や Microsoft Word/Excel など、ファイル形式ごとに提供される抽出プログラムを使用して、インデックスに取り込むべき情報を抽出する。本研究では、データベースのファイル形式を csv ファイルとして扱っている。また、インデックスに取り組む情報を抽出する際、テキスト以外にも、スタイルや制作者など、文書に格納されているさまざまな情報を利用する。

最後のインデキシング処理は、抽出された情報をインデックスに登録する仕組みである。インデックスに登録されることで、その文書は検索対象となり、検索が可能となる。

# 3. 問題解決アドバイスシステムの実装

本章では、第２章で述べた PAS の外部仕様や、形態素解析などの検索手法を基に、実際にシステムとして構築する。またシステムの内部仕様を示し、その後、データベースの構成や索引を作成する方法についても示す。

## 3.1　問題解決アドバイスシステムの内部仕様

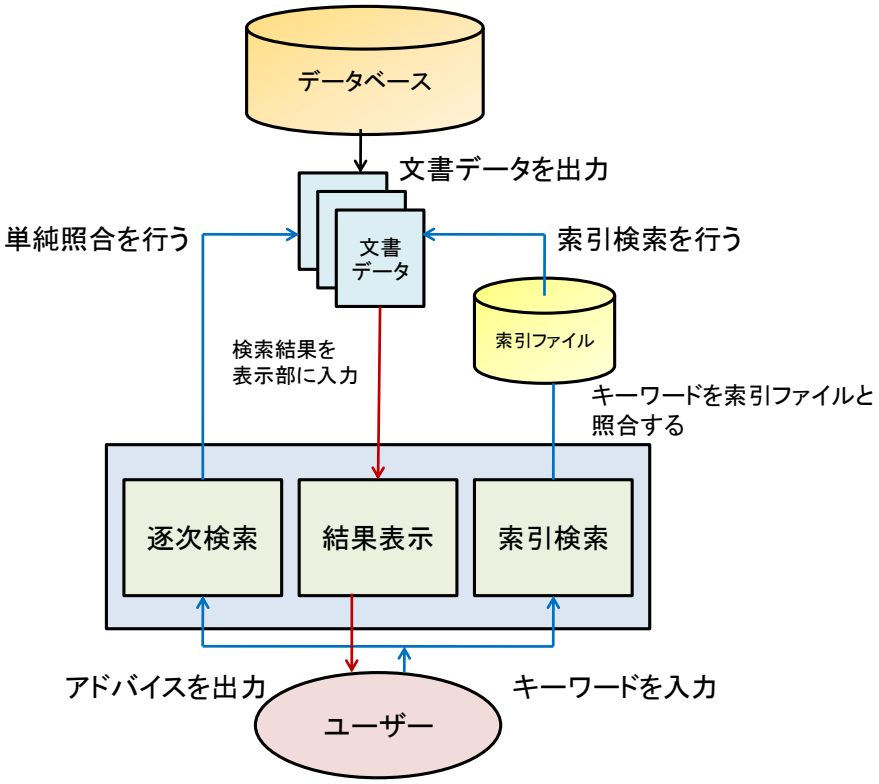


図 5. システムの内部仕様

図５は PAS の内部仕様図を示している。ユーザーが入力したキーワードの流れが青い矢印であり、データベースから返ってくるアドバイスの流れが赤い矢印である。ユーザーが悩みとして入力したキーワードが“逐次検索”と“索引検索”を行う部分にそれぞれ流れ、逐次検索は文書データに対して、単純照合を行う。また索引検索は、索引ファイルに一度入力されその後、文書データに対して検索を行う。二つの検索手法により、得た結果を結果表示部に入力し、ユーザーにアドバイスとして出力する。

### 3.2　データベースの内部仕様

データベースの内部仕様は、主に 7 項目に分類される。発言の番号を示す“No.”、発言した人物の情報を示す“Person”、アドバイスとなる発言内容を格納する“Comment”、著名人の発言した書籍の情報となる“Source”、著名人が発言をした日付を示す“Date”発言内容を予め決めておいたカテゴリに分類するための“Category”、そして利用者の検索キーワードに発言を該当させるための情報を予め予想し記憶させておく“Keyword”である。

以上の項目に対しエクセルを用いて、データベースに構築し、保存形式を csv ファイル形式として保存した。csv ファイル形式で保存した理由として、csv ファイル形式は、データをカンマ(″,″)で区切って並べたファイル形式であり、主に表計算ソフトやデータベースソフトがデータを保存するときに使う形式であるが汎用性が高いという利点のためである。よって、多くの電子手帳やワープロソフトなどでも利用できるため、異なる種類のアプリケーションソフト間のデータ交換に使われることも多い（本研究では、cygwin を使用して結果表示を行う)。また、実体はテキストファイルであるため、テキストエディタやワープロなどで開いて直接編集することも可能である点も保存形式として採用した理由である。

表 1. データベースの内部仕様

| 項目 | 項目内容 | 具体例･現状 | 文字列サイズ |
|---|---|---|---|
| No. | 発言につける番号 | 1,2,3,4… | int |
| Person | 発言した人物 | イチロー | char[200] |
| Comment | 発言内容 | 収録数175 | char[1000] |
| Source | 情報元 | 書籍名など | char[500] |
| Date | 人物が発言した日時 | 年月日 | char[200] |
| Category | 発言を大まかに分類 | スランプ、努力 | char[500] |
| Keyword | 発言内容からユーザーが入力しそうなキーワードを推定してつける | 単一の単語<br>例)理想、評価 | char[500] |

本研究で実装したデータベースの発言数は、４人の著名人を合わせて３０５件収録してある。またデータベースは著名人の発言を収録しているため、文字列のサイズを各々の項目ごとに設定する必要がある。

利用者の検索要求に対応するアドバイスが適切に出力されるように、第４章で行う実験によって検索結果が 0 と出力された際に、データベースの内容を確認し、利用者の検索要求を Keyword に情報として記録する。このことによって、データベース内の情報を増やしシステムの改善を図る。

### 3.3　データベースとインデックスの相関関係

索引検索を検索システムに実装する際に、索引を作成する必要があることは前章で述べた。この項では、具体的なツールや手法を述べる。

索引を作成するにあたり、まずデータベースに入力した文書ファイルに対して、形態素解析を行う必要がある。本研究で使用したツールは、“MeCab”を使用した。

“MeCab”はオープンソースの形態素解析エンジンで、奈良先端科学技術大学院大学出身、現 Google ソフトウェアエンジニアで Google 日本語入力開発者の一人である工藤拓氏によって開発されている。名称は開発者の好物「和布蕪（めかぶ）」から取られた。

以下の図 6 はデータベースとインデックスの相関関係図である。まずデータベースから文書ファイルを出力し、MeCab に入力する。そして形態素解析を行い、文書ファイルの文書を単語と語句に分解し、単語・語句ファイルとして出力する。最後に出力した単語・語句ファイルのそれぞれの単語・語句に対して、ID（数字）をつける。以上の手順を C 言語のプログラミングによって実装を行った。

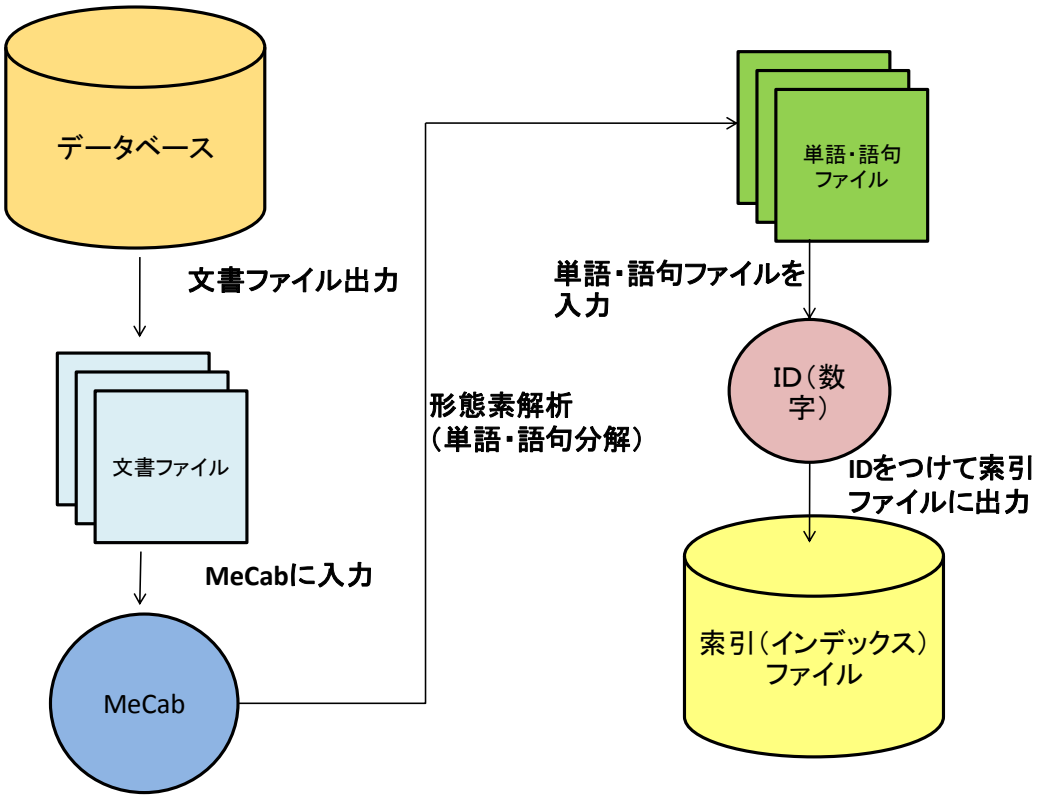


図 6. データベースとインデックスの相関関係

### 3.4　表示・検索方法

逐次検索は文字列照合の単純照合を用いて、データベース内のテキストに対してユーザーの入力した検索ワードを照合し結果に表示する。以下は単純照合のアルゴリズムである。

アルゴリズム１：単純照合による文字列照合

入力：テキスト text、キーワード key
出力：テキストストリング中のキーワード位置
Metohd

```
begin
    for i:=1 to m-n+1 do
    begin
        for j:=1 to n do
        if text i+j-1≠key[i] then
            next;
        print i;
        1:
    end
end
```

上のアルゴリズムでは、まず外側の for ループで、データベース内のテキストの文字列照合開始位置(変数 i で表す)を左側から、順次、1 文字ずつ右にシフトしている。テキスト中にキーワードが存在する可能性があるのは、テキストとキーワードの右側がそろう位置までである。最終のテキスト中の文字列照合開始位置は m-n+1 となる。内側の for ループと if 文では、テキスト中の i 文字目からキーワードとの照合を行う。キーワード中の全ての文字との照合に成功した場合は。この文字列照合の解として i を print 文を用いて出力する。もし途中でテキストとキーワードで文字が異なる場合は、その時点でキーワードが出現する可能性はなくなる。それ以上、文字列照合を続けても無意味なので、next 文により内側の for ループから抜け出す[9]。

索引検索は、索引ファイルを用いて利用者が入力した検索単語に対する文書や位置情報を索引ファイルから取得し、各情報間の照合を行うことにより、目的とする文書や文書内での位置を特定する。検索方法は、索引ファイルに格納されている位置情報１(ID)を、利用者が入力したキーワードと参照し、その後、索引ファイルの位置情報１(ID)とデータベース内のテキストの位置情報２(ID)を参照することにより結果を得る。

アルゴリズム２：索引検索による文字列照合

入力：検索質問　key
出力：key が存在する文書とその出現位置

手順１：{検索単語に対応する索引ファイルの情報を取得}
利用者の検索単語に対して、索引ファイル内を検索し、同じ検索単語に付与される位置情報１(ID)を取得する。
手順２：{位置情報の照合}
索引ファイルの位置情報１(ID)を用いてデータベース内のテキストの位置情報２(ID)を求める。
手順３：{求めた位置情報から出現文書と出現位置を特定}
手順２で求めたテキスト内の位置情報２(ID)を含む文書を検索結果として出力する。

---

実際に作成したデータベース内のテキストを用いて例を示す。
例・データベースにイチロー選手の
１.第三者の評価を意識した生き方はしたくない。
２.自分が納得した生き方をしたい。
という二つの発言を入力してある。
この発言に対して、形態素解析を行った後、生成された単語・語句ファイルに ID(数字)をつけて索引ファイルを作成する。重複している単語・語句の出現回数を数えておく。

表 2. テキストから作成した索引ファイル

| ID | 語句・単語 | 出現回数 |
|---|---|---|
| 1 | 第三者 | 1 |
| 2 | の | 1 |
| 3 | 評価 | 1 |
| 4 | を | 2 |
| 5 | 意識 | 1 |
| 6 | した | 3 |
| **7** | **生き方** | **2** |
| 8 | は | 1 |
| 9 | く | 1 |
| 10 | ない | 2 |
| 11 | 。 | 1 |
| 12 | 自分 | 1 |
| 13 | が | 1 |
| 14 | 納得 | 1 |
| 15 | い | 2 |

またデータベース内のテキストに対しても、形態素解析を行い ID(数字)を付与する。

表 3. データベース内のテキストの ID

| ID | 1 | 2 | 3 | 4 | 5 | 6 | **7** | 8 | 6 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 単語・語句 | 第三者 | の | 評価 | を | 意識 | した | **生き方** | は | した | く | ない |
| ID | 11 | 12 | 13 | 14 | 6 | **7** | 4 | 6 | 15 | 11 | |
| 単語・語句 | 。 | 自分 | が | 納得 | した | **生き方** | を | した | い | 。 | |

逐次検索であれば、入力した検索単語に対してテキストの左側から順に探索を行い、テキストと検索単語が一致すれば結果を出力するので、全ての照合が終わるまで時間がかかる。一方、索引検索は、索引ファイルとデータベース内のテキストに付与された ID の照合を行うだけなので検索にかかる時間が、逐次検索と比べて短くなる。

上の２つの表で、“生き方”と検索した場内には、数字の 7 番目の情報を含む文章を結果として出力を行う。逐次検索であれば照合を順次、２度行う必要があるが、索引検索であれば、同じ ID の照合を行うので、検索する照合の回数を減らすことができる。

小規模のデータベースであれば、メリットを十分に感じることはできないが、データベースの情報量が増えれば、前処理を行う手間があるが、検索をより高速に行うことができる。

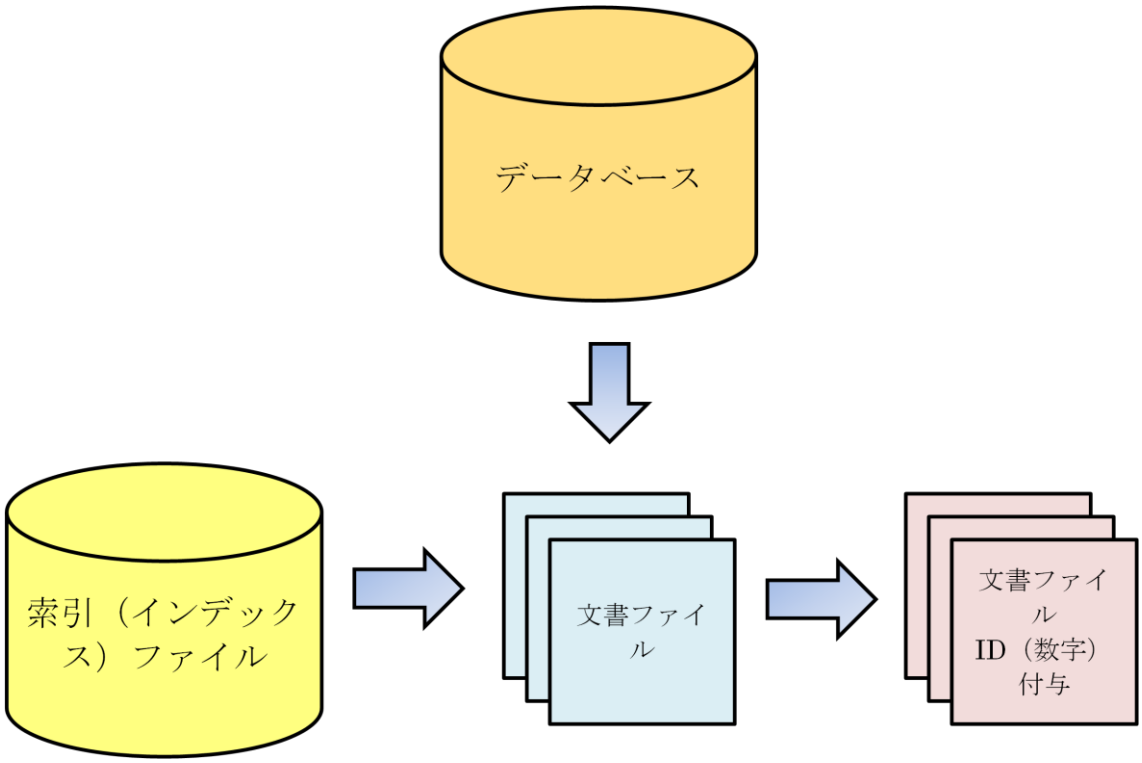


**図 7. 文書データに ID をつける流れ**

図 7 は文書ファイルに ID を付与する流れを示している。本研究では、予めデータベースの文書ファイルを、形態素解析を行うことにより索引ファイルを作成する。索引ファイルは、解析した単語・語句に対して ID を付与してある。その索引ファイルの ID を元に、データベース内の文書ファイル含まれている同じ語句・単語を照合して、同じである単語・語句に索引ファイルと同じ ID を付与していく。

索引検索では、新たに生成された ID を付与された文書ファイルの ID と索引ファイルに付与されている ID の情報を照合することで、利用者の検索単語に対する検索を実現させている。

# 4. 実験と評価

本章では、第３章で実装した PAS の結果表示例を示した後、実際に利用者に PAS を利用してもらい評価を行う。また逐次検索と索引検索の照合回数を比較し、それぞれの性能評価を行う。

## 4.1 結果表示例


図 8. PAS の実行画面と検索結果

図８は PAS の起動画面と検索を行った結果を示している。PAS を起動させると、逐次検索と索引検索のどちらを行うかを選択し、その後に利用者の悩みとなるキーワードを入力し結果を表示させる。また逐次検索と索引検索の照合回数も表示し、後述の照合回数の比較を行えるように実装した。

### 4.2　利用者の評価

以下の表４に利用者に入力してもらったキーワードと、結果のヒット数と出力されたアドバイスが役に立っているかを５段階で評価してもらった点数を示す。１点が最低点であり、５点が最高点である。

**表 4. 実験結果**

| 検索ワード | ヒット数 | 著名人 | No | 評価 |
|---|---|---|---|---|
| 成長 | 2 | イチロー | 41 | 4 |
| | | 松下幸之助 | 160 | 2 |
| スランプ | 0 | 該当なし | | |
| 最悪 | 0 | 該当なし | | |
| 目標 | 1 | イチロー | 32 | 3 |
| 生き甲斐 | 1 | イチロー | 28 | 5 |
| 期待 | 3 | イチロー | 29 | 1 |
| | | イチロー | 111 | 1 |
| | | イチロー | 120 | 4 |
| 理想 | 2 | イチロー | 49 | 4 |
| | | 松下幸之助 | 144 | 4 |
| 悩み | 1 | 松下幸之助 | 144 | 3 |
| 楽しみ | 2 | イチロー | 122 | 3 |
| | | 松下幸之助 | 140 | 4 |
| 怒り | 0 | 該当なし | | |
| 友情 | 1 | 本田宗一郎 | 217 | 3 |
| 勝つ | 3 | 中田英寿 | 244 | 4 |
| | | 中田英寿 | 270 | 3 |
| | | 中田英寿 | 275 | 4 |

PAS のアドバイスの有効性は、（評価の平均点）＝（評価の合計）÷（ヒット数）で求める。表４のアドバイスの平均点は、３．２５であった。計算ではヒットしなかったキーワードに関しては、評価の対象としていない。

以下に表４のアドバイスとして出力された著名人の発言の一例を示す。

１．キーワード　「**成長**」　著名人　「イチロー」

「初心を忘れないことっていうのは大事ですが、初心でプレイしていてはいけないんです。**成長**した自分がそこにいて、その気持ちでプレイしなくてはいけない。」

２．キーワード　「**友情**」　著名人　「本田宗一郎」

「人に**友情**を求めるなら、まず彼の秘密を守ることだ。」

３．キーワード　「**勝つ**」　著名人　「中田英寿」

「(**勝つ**ことへのこだわりについて）勝たなきゃ、未来がない。」

次に、表４のアドバイス以外に出力された著名人の発言の一例を示す。

４．キーワード　「**成功**」　著名人　「松下幸之助」

「**成功**するところまで続ければ、それは**成功**になる。」

５．キーワード　「**失敗**」　著名人　「本田宗一郎」

「日本人は、**失敗**ということを恐れすぎているようである。どだい、**失敗**を恐れて何もしないなんて人間は、最低なのである。」

６．キーワード　「**一人前**」　著名人　「松下幸之助」

「もし一緒に歩いている友人が、途中で何かにつまずいて転んだというようなことがあったら、誰もがその友人が起き上がることに手を貸す。そういう行動がごく自然にとれるのが**一人前**の大人というものである。たとえば適切ではないかもしれないが、わがコクと欧米との経済関係についても、現に欧米が困っているわけだから、そのことに対して友好国としての好意をもって、できる限りの協力をしていくということが、日本としてのとるべき道だと思う。」

７．キーワード　「**先輩後輩**」　著名人　「中田英寿」

「(チームや代表の**先輩後輩**関係について問われて）年齢や経験を問題にするなんてナンセンス」

次に、入力したキーワードの検索結果で、複数ヒットしたアドバイスについて、比較を行う。表５に、「**他人**」というキーワードを入力したときの、複数ヒットしたアドバイスの一覧を示す。評価の“A”と“B”の項目は、実験に協力してくれた人たちを表す。以上の二者に、複数ヒットしたアドバイスについて各々５段階で評価した点数も示し評価を行ってもらった。

**表 5. 複数ヒットした結果の比較**

| 著名人 | コメント | 評価 | |
|---|---|---|---|
| | | A | B |
| イチロー | 人々は僕のことを探検家とか開拓者と言います。でもそれは他人の意見。僕はそのためにアメリカに来たのではなく、野球をしに来たのです。 | 2 | 1 |
| 本田宗一郎 | 家族の人間を可愛がる人は、他人にはきつい人だ。 | 1 | 1 |
| 松下幸之助 | 人の歩む道も国の歩む道も結局同じことではなかろうか。ボンヤリしていては、道はひらけぬ。他人任せでは道はひらけぬ。つまりは、われ他人とともに懸命に考えて、わが道をひらく如くに国の道もひらかねばならない。そうしなければならないのが民主主義なのである。おたがいに三省したい。 | 4 | 4 |
| 中田英寿 | 俺は自分の真実を他人に分かってもらおうとは思わない。全ての人に自分を理解してもらうなんて不可能なことでしょう。それは、おれがサッカー選手だから起こる特別なことじゃなく、普通の学校や会社でも同じだと思うから、仕方がないことだと思うし、諦めているところもある。自分や自分の守るべきものは、俺にしかわからないよね。 | 3 | 5 |
| イチロー | 人と付き合うと言っても、ほとんどの人は他人ですよね。ゆとりを持って接することができたら、世界が全然変わってくると思うんですよ。 | 3 | 5 |

「他人」と入力した際、ヒット数は「５件」であった。収録した著名人全員が、発言を行っている。評価の点数は、A、B それぞれで違っている。点数の低いアドバイスの評価は、A、B ともに類似している。ただし A が３点と評価したアドバイスについて B では、５点と高評価を得ている。この結果から、人によって評価が異なるという法則を発見することができた。

### 4.3　逐次検索と索引検索の照合回数の比較

表 6. 照合回数の比較

| | 照合回数 | |
|---|---|---|
| キーワード | 逐次検索 | 索引検索 |
| 成長 | 50,364 | 17,196 |
| 目標 | 50,345 | 17,717 |
| 生き甲斐 | 49,352 | 17,230 |
| 期待 | 49,650 | 16,442 |
| 理想 | 49,948 | 16,498 |
| 悩み | 49,866 | 17,088 |
| 楽しみ | 49,396 | 17,214 |
| 成功 | 49,650 | 16,310 |
| 失敗 | 49,821 | 15,871 |
| 一人前 | 49,464 | 19,170 |
| 先輩後輩 | 49,238 | 検索不可 |

索引検索で検索が不可能とならなかったデータの、逐次検索の照合回数の合計と、索引検索の照合回数の合計から比較した。その結果、索引検索の方が逐次検索よりも約３倍、照合回数が少ないという数値を得た。よって索引検索は、索引の作成などの前処理の手間はかかるが、逐次検索よりも照合回数が少なくなるというメリットを証明することができた。

また表６の「先輩後輩」というキーワードに対して索引検索が不可能となった原因について考えると、逐次検索は入力されたキーワードを文字列として、文書データに対して文字列照合を行うので検索が結果として返ってくる。一方、索引検索は、「先輩」、「後輩」という単語に対して、それぞれ ID を付与している。その結果、「先輩後輩」という単語としては ID を付与していないので、索引検索で検索が不可能となった。

改善するためには、単語・語句ファイルを作成する際に、「先輩後輩」という語句も形態素解析し、ID を付与するか、「先輩」、「後輩」という２つのキーワードに対して検索が行えるようにシステムを実装すれば可能となる。

# 5. 考察

本章では、本研究で実装した PAS の評価と改善案について述べ、また PAS の並列化について検討を行う。

## 5.1 システム全体の評価

今回、実験を行い出力されたアドバイスについて、評価は十人十色であるが、私見としては、著名人のアドバイスを利用者自身がどのように解釈するかで、価値は変わると考える。アドバイスが、利用者が望んでいたものと違ったとしても、それをどのような視点で捉えるかが重要ではないかと考える。一見的外れのように思えても、著名人の考え方を知ることで、問題解決に役立つと実感した。PAS を作成し、著名人の発言の全てが必ずしも高評価を得るわけではないが、一定の価値あるアドバイスの提供につながっていると考える。

またシステム構成としては、全文検索の基本的手法である逐次検索と索引検索について、逐次検索は、アルゴリズムを実現し結果を得ることができた。また、索引検索においては、データマイニング技術の一種であるテキストマイニングを行い、形態素解析により、索引検索のアルゴリズムを実現することができた。

さらに、逐次検索と索引検索の照合回数の比較において、索引検索が逐次検索よりも照合回数が約３倍少なくて済むという結果を得ることができた。ただし、逐次検索では、逐次検索と比べてヒット数が異なるという事態が発生した。この問題について次項で改善点を挙げ、改善法を述べる。

## 5.2 システムの改善点

PAS の実験を行い判明した問題点は、利用者の悩みに対して適切なアドバイスを提供できたかどうかという点である。実験の結果、利用者の評価の平均点は、３．２５であった。今後は、さらに発言をデータベースに収録し、多種多様な悩みとなるキーワードにアドバイスとして対応させたい。

そのために、検索の精度向上を行う必要がある。PAS は検索のキーワードに１語にしか対応できておらず、第４章で判明した索引検索の「先輩後輩」というキーワードに対しても検索が実行できるように、今後は２語、３語と複数入力が行えるようにし、検索手法の基礎技術である AND 検索や OR 検索にも対応させたい。さらに、「先輩後輩」という単語に形態素解析を行い、ID を付与する必要がある。

また、「の」や「が」といった助詞に対しても ID を付与している。PAS において当初は、助詞に関しては ID を付与する対象外として実装を行っていたが、助詞を対象から省いて名

詞のみに行うと、微妙に文書データの ID が、索引ファイルの ID とのズレが生じて索引検索を正確に実行できない事態に陥ったので、今回は形態素解析を行った全ての語句・単語に対しても ID を付与した。さらに実験を通して、逐次検索と索引検索でヒット数が異なるキーワードも何点か見受けた。これは文書データに索引ファイルの ID を用いた ID の付与が正確にできていない部分があることを示している。今後は名詞のみに正確に ID を付与することができるよう改善していく必要がある。

データベースの“Keyword”には、著名人の発言から利用者が入力しそうなキーワードを予め推定し入力しておくとしたが、実際には著名人の発言がどのような悩みに対して有効であるかということを独自で判断することは極めて難しく、“Category”に関してもどの分野に分類できるのかということの判断も同様に困難を極めた。以上の作業は、完璧にデータベースに実装できたわけではないので、今後も改善すべき点である。

以上の点を改善できれば、利用者の悩みに対して適切なアドバイスを提供することができ、PAS の有効性はさらに向上できると考える。

### 5.3 並列化に向けて

我が研究室では、並列計算の研究を行っている。現在は PC クラスタを用いた OpenMP や PVM といった並列処理言語で処理を行う方法と、GPU を用いた並列処理言語 CUDA で処理を行う方法について研究を行っている。

PAS において、上記の並列計算を行えるようシステムに実装することで高速化の有効性のメリットを十分に与えることができる。具体的には、索引ファイルを作成するために形態素解析を行う処理がある。文書データに形態素解析を行い、ID を付与する処理は高速化を行う必要があると考える。索引検索を行う際に、索引ファイルの ID を用いて文書データのテキストに対して ID を付与する処理も時間を有している。PAS 起動時に、この前処理を行っているが、cygwin の画面上に長々と処理を実行している。また起動時に毎回この処理を行っているのでシステムの内部について仕様変更を行うか、並列計算を行えるように実装し高速化を行う必要があると考える。

# 6. おわりに

本研究では、人々が日々直面する様々な問題を解決するために、近現代において偉業を達成した著名人の発言をアドバイスとして提供する問題解決アドバイスシステムの設計と実装を行った。

実装したシステムについて、研究室の学生に実際に利用してもらい、悩みとなるキーワードに対して検索されたアドバイスについて評価を行ってもらった。そして、著名人の発言によるアドバイスの有効性は、３．２５という結果であった。この結果を受けて、今後もデータベースに数々の著名人の発言を収録し、データベースの情報をより強化する課題が発見できた。

また、設計と実装を行うために、WEB 検索技術で広く用いられている全文検索の逐次検索と索引検索の基本的手法をシステムに採用し、実際に逐次検索と索引検索の照合回数の比較を行った。照合回数の比較により、PAS においては、索引検索は逐次検索よりも約３倍照合回数が少なくて済むということがわかった。

さらに、利用者の要求に適切なアドバイスを提供するために、AND 検索や OR 検索といった技術を取り入れる必要があると実感した。

本研究を通して、現代の WEB 検索技術の検索手法や、テキストマイニングの基本的方法論について学習することができた。

# 謝辞

本研究の機会を与えて下さり、研究のご指導を頂いただけでなく、進路の相談にも快く受けて頂きました山崎勝弘教授に深く感謝の意を表します。また本研究に関して、データマイニングや検索技術の基本的方法論をご指導頂きました情報理工学部 情報システム学科 コンピューター―システム研究室の小柳滋教授と龍田賢治氏に深く感謝いたします。最後に本システム実装に向けて多大なご指導をして頂いた PISHVA JOHN CYRUS P 氏と、様々な助言を頂きました高性能計算研究室の皆様に深く感謝いたします。

# 参考文献


[1] イチロー，糸井重里：イチローに糸井重里が聞く，朝日新聞出版，2010.

[2] イチロー，デイビット・シールズ，永井淳：イチローUSA 語録，集英社新書，2001.

[3] 松下幸之助：松下幸之助経営語録，PHP 文庫，1993.

[4] 本田宗一郎，梶原一明：本田宗一郎　男の幸福論，PHP 文庫，1988.

[5] 別冊宝島編集部：人生の指針が見つかる「座右の銘」1300，宝島社，2010.

[6] 中田英寿，島田雅彦、小松成美：中田語録，文藝春秋，1998.

[7] 小泉修：ファイル編成から SQL まで図解でわかるデータベースのすべて，日本実業出版社，2003.

[8] 情報処理学会：情報処理 1　2005Vol.46No.1 通巻 476 号　特集　最新！データマイニング手法，2005.

[9] 北研二，津田和彦ほか：情報検索アルゴリズム，共立出版，2002.

[10] 山本毅雄，橋爪宏達ほか：全文検索　技術と応用，丸善株式会社，1998.

[11] 辻本篤志：テキストマイニングを用いた新聞記事検索システムの提案，立命館大学情報理工学部情報システム学科卒業論文，2008.

[12] 南扶友子：文字列照合の OpenMP による並列化，立命館大学理工学部電子情報デザイン学科卒業論文，2009.

[13] MeCab 公式サイト
< http://mecab.sourceforge.net/>

[14] Think IT　第２回　高度な日本語検索を実現する技術
< http://thinkit.co.jp/article/788/1>

[15] 厚生労働省　自殺・うつ病等対策プロジェクトチームとりまとめについて
< http://www.mhlw.go.jp/seisaku/2010/07/03.html>