

形態素解析なしで ReadMe 解析を用いた Web からのソフトウェア検索システムの提案

岩越守孝[†] 荒井裕樹[†] 佐治享[†] 中村真[†] 増田英孝[†] 中川裕志[‡]

東京電機大学工学部[†] 東京大学情報基盤センター[‡]

1 はじめに

P2P などの検索システムでは、ファイルの検索はファイル名を中心とした検索方法が主流である。しかしこれでは、ファイル名が明らかになっていないと検索ができない。そこで本研究では、ソフトウェアアーカイブに含まれている ReadMe ファイルの内容を参照してキーワード検索するシステムを提案する。ReadMe ファイルの内容を参照するために、あらかじめファイルの内容を適切に表すインデックスファイルを作成する。このファイルは形態素解析を用いずに単語の候補を取り出し、単語の重要度を計算することによって重要な単語だけを残してインデックス化する。

表 1: 圧縮形式

圧縮形式	個数	%
*.exe	63	18.9
*.lzh	189	56.5
*.zip	77	23.1
その他	5	1.5

表 2: ReadMe ファイルの種類

ファイルの種類	個数	%
readme.txt 等の一般的なタイプ	191	57.2
お読みください.txt 等の日本語名	14	4.2
ソフトウェア名.txt 等	39	11.7
その他の名前.txt	10	3.0
アスキーファイル (doc,html,htm,sjs,ja)	33	9.9
バイナリファイル (pdf,hlp,chm)	19	5.7
ReadMe ファイルなし	28	8.3

2 ソフトウェアアーカイブの収集

実際のソフトウェアアーカイブを調査するために、Web で配布されているプログラムを調査した。窓の杜 (<http://www.forest.impress.co.jp/>) および Vector (<http://www.vector.co.jp/>) で配布されているソフトウェアアーカイブを 334 個収集し、圧縮形式および ReadMe ファイルの形式を調査した。

収集したプログラムの圧縮形式は表 1 の通りである。プログラムの種類、対応 OS、概要等が書かれている ReadMe ファイルがどの形式で格納されているかを調査した結果を表 2 に示す。

3 圧縮ファイルからのテキスト抽出

ReadMe ファイルは圧縮されたアーカイブファイル中に存在するため、そのまま内容を参照することができない。前節の調査より、主な圧縮形式は ZIP 形式と LZH 形式である。これらの形式の圧縮ファイルの中のテキストファイルを展開することなく抽出するプログラムを作成した。

4 インデックスファイルの作成

ここでのインデックスファイルは、文書中のキーワードを重要度の高いものからリスト化したファイルと定義する。このキーワードの選出を形態素解析を用いずに行う。形態素解析を用いるためには、辞書とそれを扱うプログラムが必要になる。P2P システムのように、サーバが存在しない形態のシステムでは、各ノードが形態素解析器を持つことは現実的ではない。よって、インデックス作成のために異なる字種区切りを手がかりとする。単語の

Software Retrieval System from Web by Looking up Keywords in Readme Files without Morphological Analysis

Moritaka Iwakoshi[†], Hiroki Arai[†], Akira Saji[†], Makoto Nakamura[†], Hidetaka Masuda[†] and Hiroshi Nakagawa[‡]

School of Engineering, Tokyo Denki University[†], Information Technology Center, The University of Tokyo[‡]

分け方は、1.URL、メールアドレス、ファイル名は途中で字種が変わっても1つとして扱う、2.「カタカナと漢字」、「アルファベットと漢字」、「数字と漢字」の連続はまとめる、3. ひらがなと記号は削除する、とした。まず文書中の各文に対して、異なる字種間の区切りにタグを挿入する。そして、その区切り文字を用いてキーワードごとに分類する。すべての文書についてそれを行い、ファイルごとにキーワードの集合を作成する。

TF・IDF の計算とその問題点

キーワードの重み付けの手法にはTF・IDF(Term Frequency・Inverse Document Frequency)がある。しかし、クライアントサーバ型であれば、サーバに全文書が存在するが、P2P 型では1つのノードがすべての文書を保持しているわけではない。IDF は全文書数に依存するため、このような形態では、IDF を正しく計算することができない。1つのノードにおいて新たな README ファイルのインデックスファイルを作成する際に、その端末に登録してあるファイル数が少ないため、IDF があてにならない。そこで IDF 値の特に低かったもの(収集したファイル群のデータを用いて計算)の単語とその IDF 値のリストをあらかじめ作成し、プログラム内に組み込み、全 P2P ノードで共有することにした。そして、新たなインデックスファイルを作成する際の TF・IDF の計算に用いる IDF は、TF が高くても「IDF 値が特に低い単語リスト」に含まれている単語は削除する。値の低いリストにないものは、リストを作成した際の全ての単語の IDF の平均値を IDF として使用(これらはプログラム中に用意する)する。収集した 148 ファイルに対して IDF 値を計算した結果、キーワード数は 12641 個になり、IDF 値は最大 5.00、最低 0.44、平均 4.56 になった。表 3 に IDF 値の特に低い単語リストの一部を示す。

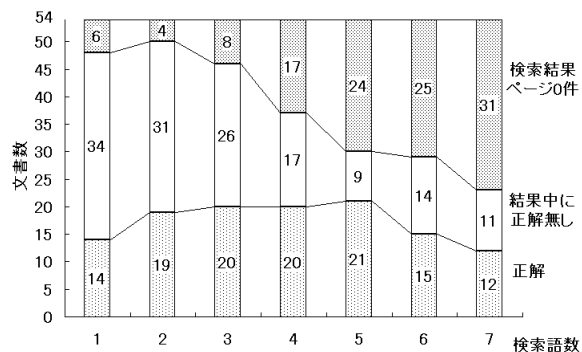
5 Google API を用いた検索能力の予備調査

前述で求められた上位の IDF のキーワードが本当に検索をする上で有効であるか、Web 検索によっ

表 3: IDF 値の低い単語の例

単語	IDF
windows	0.44
場合	0.45
ファイル	0.49
使用	0.61
削除	0.62
表示	0.65
設定	0.67
変更	0.70

図 1: Google API を用いた評価



て検索能力の予備調査を行った。そこで、Google API (<http://www.google.com/apis/>) を用いて検索プログラムを作成した。任意の 54 個のテキストファイルに対して、リストの上位の単語(第 1~7 位)を用いてキーワード検索し、検索結果の上位 10 位以内にそのソフトウェアと関連するページが見つかった場合に正解とする。結果を図 1 に示す。この結果では、上位 5 位までの語を使用すると最も正解率が高くなった。

6 おわりに

P2P システムを利用したキーワード検索を行うために現状のアーカイブを調査し、インデックスファイル作成とその予備調査を行った。今後は IDF 値の低いリストの最適化を行う必要がある。最終的な目標は P2P システムに組み込み、キーワードからファイル検索を可能にすることである。