検索キー・コンプリーションを実装した全文検索システムの開発

Fast Full-Text Search System Equipped with Key Completion Function

北 研二 [†] Kenji Kita　**幸山 秀雄** [‡] Hideo Kohyama　**山田 孝資** [§] Kôsuke Yamada

1. はじめに

Web 上に存在する膨大な情報の中から、ユーザの必要とする情報を入手することを支援するシステムとして、現在、Google を始めとしてさまざまなサーチエンジンが利用可能となっている。サーチエンジンを利用するためには、ユーザの情報要求 (検索意図) を具体的な検索キーという形で表現し、この検索キーをサーチエンジンに入力する必要がある。しかし、ユーザにとっては、自分の検索要求を検索キーとして具体化することは必ずしも容易ではなく、特に、初心者においては、どのような検索キーを与えれば適切な検索結果が得られるかを知ることはきわめて困難である。

我々は、単語や複合語などのごく一部が与えられたときに、与えられた文字列を自動補完し適切な検索キーをユーザに提示するという検索キー・コンプリーション機能を実装した全文検索システム FFTS (Fast Full Text Search) を開発した [1]。最近、Google も同様の機能を持った Google Suggest というエンジンを開発し、英語版で試験的にサービスを行っているが、我々の研究開発は Google Suggest に先行して行われており、日本語の検索キー・コンプリーションに関しても既に実装済みである。

2. FFTS の概要

サーチエンジンの性能は、検索速度、再現性 (検索漏れの少なさ)、適合性 (検索ノイズの少なさ) などの観点から評価することができる。全文検索システム FFTS は、以上の観点から見て高い能力を持っている。以下で、FFTS の特徴を簡単に説明する。

2.1 高い再現性および適合性

多くの検索エンジンでは、形態素解析処理により、入力された検索キーから単語を抽出し、抽出された単語に基づき検索 (多くの場合、単語の AND 検索) を行っている。検索キーを複数の単語に分解して検索すると、一見、再現性は高くなるが、半面、適合性を著しく損なう結果になってしまう。たとえば、ユーザが長い複合語を含んだページを検索したい場合にも、入力された複合語が複数の単語に分解されてしまうため、入力複合語をそのままの形で含んだ検索結果を得ることが困難になる。

一方、FFTS には単語という概念がなく、検索対象文書中のあらゆる文字列が検索可能となっている。入力された検索キーと完全に一致する文字列を含んだ文書はすべて検索可能であり、検索キーをそのままの形で含んでいない文書は一切検索しないため、再現性および適合性の観点からきわめて高い性能を持っている。

2.2 検索キー・コンプリーション機能

検索対象となる文書数が膨大になると、与えられた検索キーを含む文書数も増える。この結果、不要な検索結果が多く出力され、ユーザが必要とする情報を入手する妨げとなってしまう。このような検索結果の肥大化を防ぐには、必要な情報に関連した適切な検索キーをサーチエンジンに与えることがきわめて重要となる。しかし、適切な検索キーを考え出すことは必ずしも容易ではなく、ある程度のスキルが要求される。

上記のような問題に対処するために、FFTS では、入力されたごく一部の文字列を自動補完し、適切な検索キー候補をユーザに提示するという検索キー・コンプリーション機能を備えている (FFTS では「ヒント機能」と呼んでいる)。検索キー・コンプリーションでは、検索対象となる文書集合の中から、与えられた文字列と前方一致あるいは後方一致する適切な検索キーの候補を探してきて、これらの候補をユーザに提示する。これにより、ユーザが必要とする情報をすばやく入手することを支援する。

検索キー・コンプリーションの具体例を、図 1 および図 2 に示す。図 1 の例は、徳島大学のホームページを検索対象に、入力文字列「研究開発」を与えたときのコンプリーション結果を示している。また、図 2 の例では、某自治体のホームページを対象に、「ごみ」を与えたときのコンプリーション結果を示している。

[†]徳島大学高度情報化基盤センター & 徳島大学工学部
[‡]徳島大学大学院工学研究科 & ピーチツリー
[§]徳島大学工学部

図 1: 「研究開発」に対するコンプリーション結果

「ごみ」のように短いキーによる検索では、ユーザの意図しない検索結果が多数表示され、本来必要な情報がその中に埋もれてしまうという結果になりがちである。多数の検索結果を検討して、その中から必要な情報を探し出すのに大変な労力を要することになってしまう。これに対して、検索キー・コンプリーションは、検索を実行する前段階で検索キー自体を適切なものに洗練し、その後、検索を実行するため、効率的で無駄の少ない検索を実現できる。

また、日本語の場合には、複合名詞の末尾の名詞が、属性的な役割を持つことが多いので、与えられた文字列と後方一致する検索キー候補生成は時として非常に有用である。たとえば、自治体のホームページに対し、「相談」に対する検索キー・コンプリーションを適用すると、「教育相談」、「医療相談」、「育児相談」、「住宅改造相談」、「住宅相談」、「小児救急電話相談」、「消費生活相談」、「宅地建物取引相談」、「人権相談」、「無料法律相談」、「労働相談」などの検索キー候補が生成されるため、自治体でサービスを提供している相談窓口の一覧を容易に知ることができる。

2.3 同義語検索機能

FFTS は、同義語検索機能を備えており、異表記や簡略表現、あるいは等価表現による検索が可能である。同義語検索機能を用いるためには、同義語辞書ファイ

図 2: 「ごみ」に対するコンプリーション結果

ルにエディタ等で単語を追加するだけでよいので、きわめて容易である。

2.4 その他の特徴

- 表示設定機能により、検索結果のサマリー等の長さを自由に設定可能。
- 入力された検索キーのログ機能を備えており、ログを分析することで、ユーザのニーズを把握することが可能。
- HTML, 通常のテキスト、PDF 等の文書に対応。また、Linux, FreeBSD, Solaris, Windows 等の多数のプラットホームで動作可能。

3. 検索キー・コンプリーションの実現

検索キー・コンプリーションは、転置ファイル法 [2] を始めとする各種のインデキシング手法に実装することが可能である。いま、与えられた文字列 s に対するコンプリーションを求めることを考える。まず最初に、文字列 s を用いて検索を行い、文字列 s が出現する文書中の位置 i を同定する。次に、位置 i の前後の文字列を調べることにより、文字列 s を拡大し、s を部分文字列として含む検索キー候補を生成する。s を部分文字列として含む検索キー候補の中には、言語表現として不適切な (すなわち、単語や複合語ではない) 可能性があるが、文字や文字列の統計情報、あるいは字種

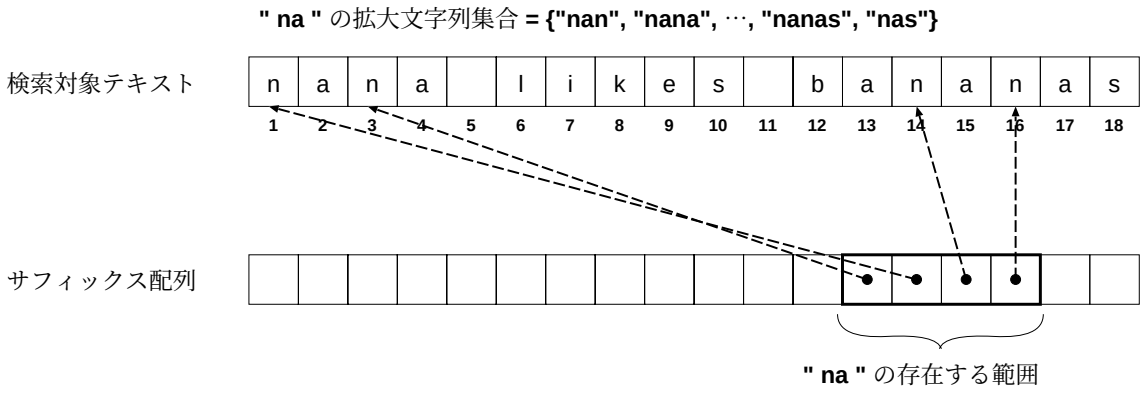

図 3: サフィックス配列を用いた検索キー・コンプリーションの例

情報を用いて、不適切なものを排除することにより、最適な検索キー候補を求める。

前述したように、検索キー・コンプリーションは各種のインデキシング手法上に実装可能であるが、例として、サフィックス配列 (suffix array) を用いた検索の場合について、以下で説明する。図 3 は、検索対象テキスト “nana likes bananas” およびこの文字列に対するサフィックス配列を図示したものである。いま、文字列 “na” と前方一致する拡大文字列を求めることを考える。まず、サフィックス配列を 2 分探索することにより、検索対象テキスト中で “na” の出現する範囲を同定する。文字列 “na” は検索対象テキストの 1 文字目、3 文字目、14 文字目、16 文字目を始点とする位置に見つかるので、各始点から始まる部分文字列 “nan”, “nana”, ⋯, “nanas”, “nas” が文字列 “na” の拡大文字列となる。仮に、頻度に基づき検索キー候補を生成するならば、この例の場合には、拡大文字列 “‘nan” および ‘nana” の頻度が他のものよりも高いので、これらを検索キー候補として提示することになる。

4. おわりに

本稿では、全文検索システム FFTS について述べた。FFTS は、高い再現性および適合性、検索キー・コンプリーション機能、同義語検索機能等を備えたきわめて高性能なシステムである。特に、検索キー・コンプリーション機能は、与えられた文字列から、より正確で検索ノイズの少ない検索キーを提示する手法であり、効率的な検索を行ううえできわめて有用である。

参考文献

[1] FFTS ホームページ:
http://www.peachtree.jp/ffts/index.html

[2] 北，津田，獅々堀：「情報検索アルゴリズム」，共立出版, 2002.