大規模日本語ウェブ文書を対象とした開放型検索エンジン基盤の構築

新里圭司 † 柴田知秀 ‡ 河原大輔 ‡‡ 黒橋禎夫 †
† 京都大学 大学院 情報学研究科
‡ 東京大学 大学院情報理工学研究科 ‡‡ 情報通信研究機構
{shinzato, shibata, kuro}@nlp.kuee.kyoto-u.ac.jp dk@nict.go.jp

1 はじめに

近年，World Wide Web (WWW) 上には膨大な量のウェブ文書が氾濫しており，その中から自身の必要とする情報を探し出すためには，検索エンジンの利用が必要不可欠である．しかしながら，既存の検索エンジンは，ユーザより与えられた数個の検索クエリをもとに，それらを含む文書へのリンクをリスト形式でユーザへと提示するだけであるため，

- 検索の対象となる文書集合の特性を考慮した適切な検索クエリを入力する必要がある
- 大量の検索結果の中から自分の欲する情報を探し出す必要がある

という問題がある．そのため，ユーザに対して適切な検索語の想起を促す想起支援システムや検索結果を分類してユーザへ提示するクラスタリングシステムなどの新しい検索サービスの実現は，今後さらに増大するウェブ文書の中から，情報を効率的に取得するためには必須となる．

上述した新しい型の検索サービスの実現のためには，その基盤となる検索エンジンが必要となる．現在，いくつかの商用検索エンジンで，その検索結果を得るための API が提供されているが，

1. 検索結果のランキング尺度が公開されていないため，検索の際に何が行われているのか API 利用者には把握できない
2. インデックスの更新が頻繁に行われるため，API を利用したシステムの再現性の確保が難しい
3. API 利用回数や取得可能な文書数に制限がある

などの問題があり，新しい検索サービスを研究・開発するための基盤として，既存の検索エンジンを用いる際の障壁となっている．

そこで本研究では，上記の問題点を解決し，新しい型の検索サービスの研究・開発を支援するため，大規模日本語ウェブ文書群（現在は，約 5,000 万ページ [1]）を検索対象とする検索エンジン基盤の構築を目的とする．

本稿では，現在我々が開発している開放型検索エンジン基盤 TSUBAKI [2]について述べる．

2 開放型検索エンジン基盤 TSUBAKI

図 1 は，開放型検索エンジン基盤 TSUBAKI を用いて「子供の体力低下」を検索した結果である．TSUBAKI の特徴としては，

- 無制限に利用可能な API および透明性・再現性のある検索結果
- 高度ウェブ処理用フォーマットによる大規模ウェブ文書の管理
- 構造的言語処理を用いたインデキシング

が挙げられる．以下では各特徴について述べる．

2.1 無制限に利用可能な API および透明性・再現性のある検索結果

TSUBAKI API [3]では，既存の検索エンジン API のような利用回数制限などを設けていない．さらに，取得可能な文書数についても制限を設けていないため，API を用いることで，検索結果に含まれる全ての文書を得ることが可能である．これにより，従来の検索エンジン API を用いた場合と比べ，より現実に近い量のウェブ文書を扱った大規模なシステムの構築を可能にしている．

[1] これらは，NTCIR5-WEB タスクにて使用された 1 億ページ [1, 3] から日本語を含んでいると判定されたページである．
[2] http://tsubaki.ixnlp.nii.ac.jp/index.cgi
[3] http://tsubaki.ixnlp.nii.ac.jp/api.cgi

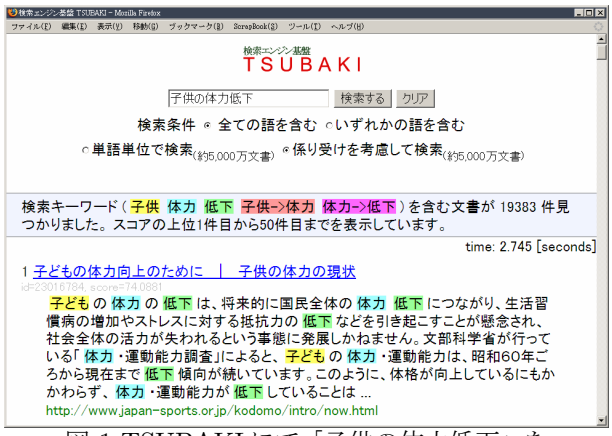

図 1 TSUBAKI にて「子供の体力低下」を検索した結果

また，TSUBAKI ではランキング尺度を公開するだけでなく，そのソースコードもオープンソース化することで，その透明性を確保している．さらに，検索対象となる文書集合を静的なウェブアーカイブ (ある時点でのスナップショット) にすることで，TSUBAKI を利用したシステムに対し再現性のある検索結果を提供している．これにより，API を用いたシステムの管理等が既存の API を用いた場合に比べ容易になることが期待できる．

今後は，検索対象となるウェブ文書を増やす予定である．その場合，API 利用者はスナップショットを指定して問い合わせることで，システムの再現性を保つことになる．

2.2 ウェブ標準フォーマット

TSUBAKI では，ウェブ標準フォーマット化してウェブ文書を管理しており，API ユーザは，オリジナルのウェブ文書に加え標準フォーマット化されたウェブ文書についても簡単に取得可能である．

ウェブ標準フォーマットとは，解析済みウェブ文書の流通を目的としたウェブ文書を管理するためのフォーマットである．図 2 に標準フォーマット化されたウェブ文書の一部を示す．標準フォーマットには，ウェブ文書を用いた研究を行う上で必要となるであろう，文書のタイトルやアンカー情報などのメタ情報に加え，ウェブ文書から抽出された日本語文，さらには，タイトルやアンカーテキスト，抽出された日本語文の形態素／構文解析結果が埋め込まれている．そのため，ユーザは，API を利用して標準フォーマット化されたウェブ文書を得ることで，ウェブ文書からの文抽出や形態素／構文解析などの前処理を行うことなく，即座に自身のシステムの処理を開始することができる．このことは，先程例として挙げたクラスタリングシステムのような動的に検索結果を処理するシステムにおいて重

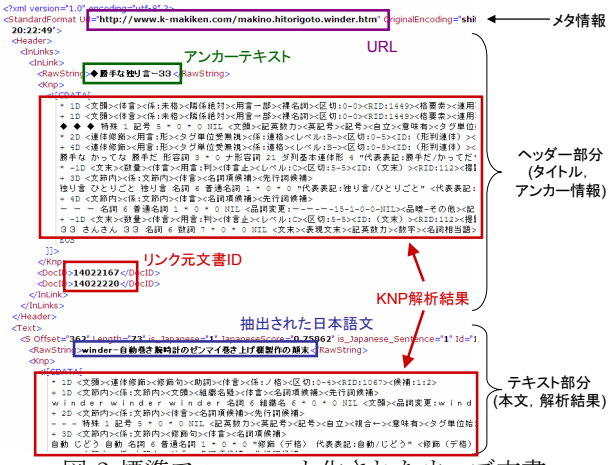

図 2 標準フォーマット化されたウェブ文書の例

要である．

2.3 構造的言語処理によるインデキシング

TSUBAKI では，

- 単語 (代表表記) インデックス
- 係り受けインデックス
- 文字トライグラムインデックス

の 3 種類のインデックスを用いて，単語のみを用いた検索，係り受けを考慮した検索，フレーズ検索をユーザへ提供している．本節では，各インデックスについて述べる．

2.3.1 単語 (代表表記) インデックス

TSUBAKI では，インデックスの単位の一つに「単語」を用いている．しかし,「子供」と「子ども」や「イチゴ」と「苺」のように，書き手によって表記のされ方が異なる単語は数多く存在するため，単に単語をインデックスとして用いたのでは，検索に「漏れ」が生じる．そこで，TSUBAKI では，単語を代表的な表記に統一してからインデキシングすることで，表記揺れによる検索漏れの問題に対応している．単語の代表的な表記としては，JUMAN [4]で形態素解析した際に得られる「代表表記」を用いている．これにより,「こども」や「子ども」は「子供」に集約されてからインデキシングされることになる．

2.3.2 係り受けインデックス

本当の意味で「情報を検索する」ためには，既存の検索エンジンのような単語の出現頻度やリンク構造を手がかりとする検索手法では不十分であり，計算機による「意味」の取り扱いが重要になる．TSUBAKI で

[4]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html

表 1 文字トライグラムが及ぼす擬似フレーズ検索への影響

検索語	擬似フレーズ検索でのヒット件数	実際に検索語を含んでいた文書数
京都大学大学院情報学研究科	773	664 (85.79%)
大画面薄型テレビ	230	224 (96.97%)
ラスト・サムライ	2,983	2,928 (98.12%)
２０００年問題	12,569	12,472 (99.22%)

は,「意味」を考慮した検索の第一歩として係り受け関係をインデックスの単位として用いている．これにより，意味を考慮した検索クエリと文書のマッチングが可能となる．例えば，次の検索クエリ

- 影響を与えたゲーム
- ゲームを与えた影響

は，単語のみをインデックスの単位とする検索エンジンでは，共に { ゲーム，与える，影響 } を含む文書を検索することになり，両者を異なる検索クエリとして区別することはできない．しかし，係り受け関係をインデックスの単位とすることで，先程の検索クエリは,

- ゲームを与えた影響 → { ゲーム, 与える, 影響, ゲーム→与える, 与える→影響 }
- 影響を与えたゲーム → { ゲーム, 与える, 影響, 影響→与える, 与える→ゲーム }

で表現されることになり，両者が異なる意味を持つ検索クエリであることを考慮した検索が可能になる．

TSUBAKI では，KNP [5]の解析結果より係り受けインデックスを作成している．

2.3.3 文字トライグラムインデックス

TSUBAKI では文字トライグラムを用いた擬似フレーズ検索を提供している．文字トライグラムにより，どの程度正確にフレーズ検索を行えているか複数のキーワードについて調査した結果を表 1 に示す．表より，全てのキーワードについて，85%以上の精度で，キーワードを含む文書を検索できていることがわかる．

3 検索アルゴリズム

本節では，TSUBAKI の検索アルゴリズムについて述べる．図 3 に「子どもの体力低下」について係り受けを考慮して検索する際の流れを示す.

TSUBAKI では，検索クエリ Q が与えられると，それを単語，係り受け，文字トライグラムと適切なインデックスの単位に変換する (ステップ 1，2).

[5]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html

1. CGI サーバーで検索クエリ，検索条件 (全ての語を含むか否か，係り受けを考慮するか否か) を受け付ける．

   例: Q=子どもの体力低下

2. 検索クエリ Q を検索条件に応じた適切なインデックスの単位（単語，係り受け，文字トライグラム）に分解し，Q' を生成する．

   例: Q'={ 子供，体力，低下，子供→体力，体力→低下 }

3. インデックスに分解された検索クエリ Q' を，31 台の検索サーバーに対して送信する．
4. 各検索サーバーにおいて，Q' に対する文書 d のスコア $score_{rank}(Q', d)$ を求め，文書 ID とスコアの組を返信する．
5. 各検索サーバーより返信された文書 ID とスコアの組をマージし，スコアに基づきソートする．
6. スコアの上位 M 件を検索結果として表示する．

図 3 検索の流れ

次いで，変換された検索クエリ Q' を，検索専用サーバーに対して送信し，並列に検索を行う (ステップ 3, 4)．TSUBAKI では 5,000 万文書を検索対象としているが，検索は 31 台の検索サーバーで並列に行われるため，各サーバーは高々200 万文書分のインデックスから，検索クエリ Q' に適合する文書を探すことになる．各検索サーバーは検索クエリ Q' に対する文書 d のスコアを以下に示す *OKAPI BM25*[2] に従い求める．

$$score_{rank}(Q', d) = \sum_{q \in Q'} BM25(d, q)$$

$$BM25(d, q) = w \times \frac{(k_1 + 1)fq}{K + fq} \times \frac{(k_3 + 1)qfq}{k_3 + qfq}$$

$$w = \log \frac{N - n + 0.5}{n + 0.5}, K = k_1((1 - b) + b\frac{l}{l_{ave}})$$

ここで，fq は表現 q の文書 d 中での出現頻度，qfq は q の検索クエリ中での出現頻度，N は検索対象としている全文書数 (TSUBAKI では 5.1×10^7)，n は q を含む文書数，l は文書 d の文書長 (TSUBAKI では文書中の自立語数)，l_{ave} は平均文書長を表す．また，k_1,k_3,b は *BM25* のパラメータであり，TSUBAKI では，$k_1 = 2$, $k_3 = 0$, $b = 0.75$ を用いている．

「子供→体力」のような係り受け関係について *BM25* を求めると，係り受け関係は単語に比べ出現頻度が相対的に低いため，高い値を得ることになる．そのため，係り受け関係について特別なボーナスを与えることなく，Q' 中の係り受け関係を含んでいる文書に対して自然な形で高いスコアを与えることができている．

全検索サーバーから文書 ID とスコアの組が返信されると，CGI サーバーではそれらをマージし、スコアに従って降順にソートする (ステップ 5)．そして，スコアの上位 M 件についてスニペッツを生成し，文書のタイトル，文書へのリンクと共に検索結果として表

表 2 TSUBAKI API で指定可能なリクエストパラメーター一覧

パラメータ	値	説明
query	string	検索クエリ (utf8) を URL エンコードした文字列．検索結果を得る場合は必須．
start	integer	取得したい検索結果の先頭位置
results	integer	取得したい検索結果の数
logical_operator	AND/OR	検索時の論理条件．デフォルトは AND．
dpnd	0/1	係り受けを考慮した検索を行うかどうかの指定．1 の時に係り受けを考慮して検索する．デフォルトは 1．
verbose	0/1	ヒット件数だけを得たい場合は 0，検索結果を得たい場合は 1．デフォルトは 1．
id	string	個別の文書を取得する際の文書 ID．オリジナルのウェブ文書，または標準フォーマット形式の文書を得る際は必須．
format	html/xml	オリジナルのウェブ文書，または標準フォーマット形式のウェブ文書のどちらを取得するかを指定．id を指定した際は必須．

示する (ステップ 6)．TSUBAKI では，$M = 50$ としている．

4 TSUBAKI API

TSUBAKI API では表 2 にあるリクエストパラメータを提供している．以下に，API にアクセスするための，リクエスト URL の例を示す[6]．

例 1: 「京都」について検索した結果の上位 20 件を取得したい場合

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=%E4%BA%AC%E9%83%BD&starts=1&results=20

例 2: 「京都」のヒット件数だけを知りたい場合

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=%E4%BA%AC%E9%83%BD&verbose=0

例 1 の場合について，TSUBAKI API より返される検索結果を図 4 に示す．検索結果には，検索語，ヒット件数，検索結果の最初の位置，取得した文書数などのメタ情報に加え，検索語を含むウェブ文書の ID，スコア，タイトル，URL，キャッシュへの URL などの情報が含まれている．標準フォーマット化された文書もしくはオリジナルウェブ文書を取得したい場合は，図 4 に示した検索結果から文書 ID を抽出し，再度 API に問い合わせる．以下に，文書 ID が 01234567 である標準フォーマット化されたウェブ文書を取得する場合のリクエスト URL を示す．

[6]紙面の都合上，“query=” の後に改行を挿入しているが，実際は不要なので注意．

```
<?xml version="1.0" encoding="utf-8"?>
<ResultSet time="2007-02-01 04:55:01" query="京都"
totalResultsAvailable="1390900" totalResultsReturned="20"
firstResultPosition="1" rankingMethod="OKAPI"
logicalCond="AND">
<Result Id="09405221" Score="10.46451">
  <Title>京都府ホテル</Title>
  <Url>http://okasoft.ddo.jp/pasokon/z_kyouto.html</Url>
  <Cache>
    <Url>http://tsubaki.ixnlp.nii.ac.jp/index.cgi?URL=
    INDEX_NTCIR2/09/h0940/09405221.html&KEYS=%B5%FE%C5%D4</Url>
    <Size>3316</Size>
  </Cache>
</Result>
<Result Id="37461679" Score="10.43856">
  <Title>ＪＴＢおすすめ・京都の宿！国内旅行／激安旅行／トリップサイト！</Title>
  <Url>http://www.tripsite.jp/jtb/kinki/kyoto1.html</Url>
 ... 中略...
  </Cache>
</Result>
</ResultSet>
```

図 4 TSUBAKI API より返される検索結果の例

例 3: 標準フォーマット化されたウェブ文書 (ID=01234567) を取得したい場合

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?id=01234567&format=xml

5 おわりに

本稿では，現在開発を進めている開放型検索エンジン基盤 TSUBAKI について述べた．その特徴として，

- 無制限に利用可能な API および透明性・再現性のある検索結果
- 高度ウェブ処理用フォーマットによるウェブ文書の管理
- 構造的言語処理によるインデキシング

が挙げられる．現在は，日本語ウェブ文書約 5,000 万件を対象とした検索が可能であり，API を介して誰でも自由に検索結果を取得できる．

今後は，(1) 検索対象となるウェブ文書数の増強，(2) 単語と単語の距離を考慮した近接インデックスの作成，(3)「メタボリックシンドローム」と「インスリン抵抗性症候群」ような同義語 (表現) への対応，を予定している．

参考文献

[1] Keizo Oyama, Masao Takaku, Haruko Ishikawa, Akiko Aizawa, and Hayato Yamana. Overview of the ntcir-5 web navigational retrieval subtask 2 (navi-2). In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pp. 423–442, 2005.

[2] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC. In *Text REtrieval Conference*, pp. 21–30, 1992.

[3] Masao Takaku, Keizo Oyama, Akiko Aizawa, Haruko Ishikawa, Kengo Minamide, Shin Kato, Hayato Yamana, and Junya Hayashi. Building a terabyte-scale web data collection ”nw1000g-04” in the ntcir-5 web task. In *NII Technical Report, No.NII-2006-012E*, 2006.