Webコンテンツのセマンティックデータ生成
Semantic Data Generation for Web Content

情報工学専攻　牛久　陽介
Yousuke Ushiku

要約: 本研究では, 検索エンジンで適当な 1 単語の検索に対する返却値であるページのリストを, 自然言語処理, テキストマイニング等を用いて, 各ページの内容によるクラスタリングを行う. その結果をセマンティックタグ等の付与を踏まえて RDF で表現することにより, 複数ページに関するセマンティクスを計算機で簡単に扱うことを可能にする. 更に人間にとってもセマンティクスに基づくユーザビリティが向上するようなシステムを提案した.

キーワード: クラスタリング, セマンティックウェブ, RDF

1　背景

今日, インターネットは多くの人に活用され, 生活の一部として欠かせない. しかしその情報の総量は急速に増え続けていてユーザには扱いきれないため, 検索会社に選び出す作業を任せている. 検索エンジンは膨大な量のページのリストを返すので, ユーザがその全容を把握するのは難しい. どこかに存在する自分が求める情報が見つからないことや, 検索会社のバイアスが入る可能性がある. 最近, セマンティックウェブという概念が再度勃興しており, それらへの解決策として注目を集めている. 従来, 計算機は自然言語を完全に理解することはできないが, 予め意味を定めた XML 等を文書に付加することにより, 計算機が自律的に情報を扱えるようになる. しかし現在のところ既存の情報にセマンティクスを付与するシステムは十分ではない.

2　関連研究

検索結果のクラスタリングに関する研究は, 本研究の課題であるウェブページの内容に着目してクラスタリングを行うコンテンツマイニングとウェブページのリンク情報に基づくストラクチャマイニングの二つがある. コンテンツマイニングの研究として, 成田らは, 検索サイトからの返却値に含まれる各ページについてのサマリを利用し, TFIDF 法や独自に定義した尺度を用いて文章中から特徴語を割り出し, 非排他的クラスタリングを行った [1]. この研究において生成されたクラスタやラベルについては未評価であり, 村松らがこのクラスタ間の関連性を取得する方法について研究を行った [2]. 本研究では, 検索エンジンに依存することを避けるため, 検索エンジンが提供する各ページのサマリを用いず, ページ自体を解析して排他的クラスタリングを行う.

2.1　RDF

ウェブ上のリソースに関する情報を明瞭かつ論理的に表現するデータモデルであり, それを記述するための言語体系である. 現在 RDF は, ある Web サイトについての更新情報を記述するフォーマットである RSS (RDF Site Summary) として最も良く用いられている.

2.1.1　RDF の基本的な考え方

RDF 関連仕様書の一つ. “Concepts and Abstract Syntax” は, RDF 設計の目標として次の 6 つを上げている.

1. シンプルで柔軟なデータモデル.
2. 論理学的な裏付けのあるセマンティクス表現と, 証明可能な推論.
3. URI に基づく拡張可能な語彙の利用.
4. XML による交換構文の採用.
5. XML スキーマデータ型によるデータの精密な型付け.
6. 誰もがどんなリソースについても記述できること.

RDF を利用する際はこれらに留意しなければならない.

3　自然言語処理

人間が日常的に使っている自然言語をコンピュータに処理させる一連の技術であり, 人工知能と言語学の一分野である. 自然言語処理には形態素解析と構文解析, 文脈解析, 意味解析の手順で表層的な観点から文章を解析する.

日本語は英語などと違い, 単語同士の間に空白を含めないため区切りが不明瞭である. そのため, 前処理として辞書と統計・確率を用いて分かち書きと呼ばれる文章を単語に切り分ける手法か, 文書から作成できる任意の n 文字以下の連続した領域を単語として扱う手法が良く用いられる. 今回用いるライブラリは前者の手法を取っている.

4 提案手法

4.1 形態素解析

本研究では, オープンソース形態素解析エンジン Mecab と, その内部辞書に複合語情報を持つ NAIST-jdic[§] を利用した. その際に, 抽出された名詞を一律で全角に変換して, 日本語が含まれない場合, 解析ミスとして捨てる. *TF*, *IDF* を算出する際の形態素解析に利用する.

4.2 Wikipedia データの解析

Wikipedia 日本語版の提供するデータ[¶] を利用して, 大域的な単語の重要度を評価する際に指標である *IDF* を計算する. 今回は 2011 年 9 月 7 日分のダンプデータである pages-articles.xml を利用した. 様々な情報を含んだ XML を エントリと本文に分解する Yoichiro Hasebe 氏の wp2txt というライブラリ[†] を利用し, 各エントリを 1 ページとして形態素解析を行い, 抽出された単語の出現するエントリ数を算出しデータベースに格納する. 該当日のデータの時点でエントリは約 145 万あり, この値を全ドキュメント数とした.

4.3 検索結果を取得

Yahoo!Japan デベロッパーネットワークが提供する検索 API[‖] を用いて, 検索結果上位 100 件の URL を得る. オプションで, html や PDF, msword 等, 検索される対象を絞り込むことができるので, 今回は html を指定した.

4.4 各ウェブページの取得

検索結果に含まれる URL のコンテンツをそれぞれ取得する. その際に, meta タグで指定されている文字コードを読み込んで, UTF-8 に変換し, 一律で扱えるようにする. 今回は javascript の解析等行っていないので, そのページにアクセスして一番最初に見える画面の情報のみを解析対象とする.

4.5 TF, IDF, TFIDF 算出

各ウェブページからタグを除去し, タイトルと本文を形態素解析をして抽出された全ての単語の出現回数をカウントする. その単語の出現回数がその文書の中での重要度の指標となる *TF* である. まず単語毎に Wikipedia のデータから算出した値が存在するかデータベースに問い合わせる. その単語のデータがデータベースに入っていなかった場合は, その都度 Wikipedia のデータを走査して出現エントリ数をカウントし, その値をデータベースに入れる. ここでもし出現エントリ数が 0 であれば, 形態素解析のミスとして, その単語は無視する.

単語 t がある文書に出現する回数を *D(t)*, 出現する文書数を *DF(t)* とし, 全文書数を N とすると, *IDF* は以下の式で計算できる.*TFIDF* は *TF* と *IDF* を掛け合わせた値である. 以下の式で算出される.

$$TF = D(t),\ \ IDF = \log \frac{N}{DF(t)},\ \ TFIDF = TF{*}IDF$$

4.6 クラスタリング・RDF 形式での書き出し

TFIDF・*IDF* による重要語ランキングから適当な閾値以上の部分だけを持ってきて, 特徴ベクトルを作成し, ページ同士を比較する. 適当に定めた閾値以上のページを同じクラスタとする. その結果を RDF 形式で出力する.

4.7 RDF の活用

生成した RDF を解析して Web サイトに表示し, 既存の検索エンジンよりも検索結果にアクセスしやすくする.

5 実験

5.1 クラスタリング

今回の提案手法で解析できたページは, 検索語により異なるが, 各検索語に対し 100 件のページを取得したところ, 平均して 8 割以上のページを解析することができた. 簡略化のために各検索語に対し 20 件のページを取得し, 「雪国」「池袋」「学習」という単語に対して, 解析を実施し, その結果を掲載する.

5.1.1 評価方法

以下に述べる F 尺度で評価する. クラスタリング結果 $C = \{C_1, C_2, \cdots, C_k\}$ と自分で作成した正解となるクラスタ $A = \{A_1, A_2, \cdots, A_k\}$ を用いる. 再現率 R_{hk} と 精度 P_{hk} は $[0, 1]$ の値を取り, それぞれ正解クラスタに近いほど, クラスタリング結果に近いほど 1 に近い値となる. A_h と C_k に対する F 尺度 F_{hk} は R_{hk} と P_{hk} の調和平均である. クラスタリング結果に対する F 尺度 F は, A_h に対して, F_{hk} が最大になるような k を求めて F_{hk} を算出し, 各 h に対して重み付き平均を取ったもので値が大きいほど良い. 再現率と精度の調和平均を取る理由は, 再現率と精度の計算式の分母が違うためである. 調和平均を取ることで, いくつの要素数を含んでいるかという重みを考慮に入れた平均値を算

[§] http://mecab.sourceforge.net/ を参照
[¶] http://dumps.wikimedia.org/jawiki/ を参照
[‖] http://developer.yahoo.co.jp/ を参照

出することが可能である．

$$R_{hk} = \frac{|A_h \cap C_k|}{|A_h|}, \quad P_{hk} = \frac{|A_h \cap C_k|}{|C_k|}$$

$$F_{hk} = \frac{2R_{hk}P_{hk}}{R_{hk}+P_{hk}}, \quad F = \sum_{h=1}^{K} \frac{|A_h|}{N} \max_k F_{hk}$$

6　結果・考察

6.1　得られた TFIDF 値

正常にできた場合

表 1 に本手順で得られた結果を，図 1 にそのサイトのスクリーンショットを示す．`http://www.yamadalabi.com/ikebukuro-md/index.html` に対しての解析結果だが，上手くできているのでキーワードを見て内容が推測できる．

正常に解析できなかった場合

表 2 に本手順で解析できなかった例を示す．また，図 3 にそのサイトのスクリーンショットを示す．`http://www.chiryohi.com/cnt/cnt_natural.html` に対して解析した結果だが，最初に読み込み中の文字だけが表示されるページが解析対象となってしまい，上手く単語を抽出することができなかった．

図 1 表 1 のターゲットとなったページ

表 1 上手く解析できた例

単語	TF	TFIDF
ヤマダ電機	0.128205128205	1.00244040935
情報	0.128205128205	0.300492660257
ポイント	0.0769230769231	0.278524224753
ケイタイ	0.025641025641	0.264513368419
サイト	0.0769230769231	0.231500303971

6.2　クラスタリングと評価

一例として，『雪国』という単語について解析を行った．先行研究のシステム NOCTURNE[1] のクラスタリング結果と共に，結果の一部をクラスタに付与されたラベルを 表 1 に示した．また，複数の単語について本システムでのクラスタリング精度の評価値を表 2 に示す．

図 2 表 2 のターゲットとなったページの読み込み直後

図 3 表 2 の左の画像のページからのリダイレクト後

表 2 上手く解析できなかった例

単語	TF	TFIDF
ただ今	0.166666666667	1.6183142608
読み込み	0.166666666667	1.23857557621
治療	0.166666666667	0.68192424179
費	0.166666666667	0.492405697855
中	0.166666666667	-0.0410001487189

6.3　出力された RDF

排他的にクラスタリングされており，各ページの重要箇所を最初から表示するための位置データを持つ．“雪国”という単語を検索した場合，以下のような形式で出力される．全ては載せることはできないので一部のみ掲載することとする．

```
<rdf:Description rdf:about="雪国">
  <ex:cluster_0 keyword="康成, 康成, 川端">
    <rdf:li rdf:resource="http://www.amaz…"
      ex:text="川端康成の代表作。　--このテキストは、文庫… 商"/>
    <rdf:li rdf:resource="http://d.hate…"
      ex:text="川端康成　製作：…芳野尹孝"/>
    <rdf:li rdf:resource="http://ww..guni.htm"
      ex:text="文庫・雪国：川端康成…康成庫・雪国：川端康成、岩波文庫・新潮日本"/>
  </ex:cluster_0>
  <ex:cluster_1 keyword="雪国, 雪国, 湯沢">
    <rdf:li rdf:resource="http:www…ts.html/"
      ex:text="雪国は津南がもたらしてくれる素晴ら...して津"/>
    …
  </ex:cluster_1>
  ...
</rdf:Description>
```

6.4　RDF の活用

一例として，Web サイトで図 4 のように読み込んで表示することができる．この使用例では一つ一つの div タグの中にクラスタ内の URL とそのサブのタグとページから抜き出したテキストを表示して，画面の情報量を

表 3 本システムで生成されたクラスタ (一部抜粋)

本システム		NOCTURNE	
DF	TFIDF	DF	TFIDF
雪国まいたけ 年初来	川端 康成	雪 くらし 情報	雪 写真 掲示板
健康	宿泊		
越後湯沢 南魚沼 ゆきぐに	観光 湯沢	生活 掲示板 社会	マガジン くらし 研究
ブラウザ フレーム サポート	ブラウザ フレーム サポート	写真 更新	案内 社会
	株価	研究 ゆき トンネル	情報 生活 ゆき
上毛高原 焼きいも			

表 4 クラスタリング精度の評価

	雪国	池袋	学習	平均
TF	0.483	0.521	0.530	0.511
TFIDF	0.526	0.571	0.637	0.578

増やすようにしている. 全ては収まらないので一部のみ掲載することとする.

他にも, 今回提案した RDF を多く集めることで, 単語のネットワークを作成し, 検索エンジンのような仕組みを考えることもできる.

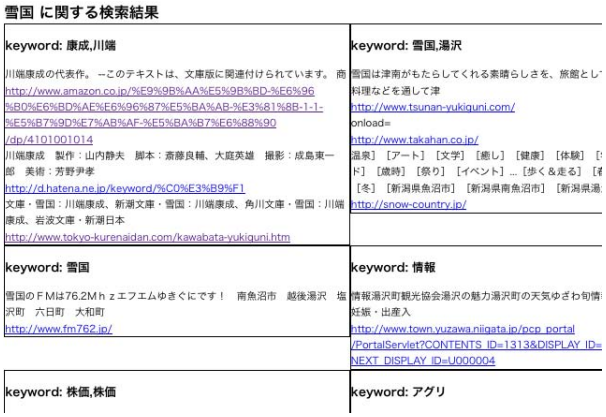

図 4 RDF の活用例

6.5 考察

先行研究のシステムとは検索を行った時期が違うので結果そのものを比べることはできないが, *TFIDF* の値を用いてクラスタリングした場合, html の構造により解析できないページはあったものの, 多くの場合そのページの特徴を表すような単語を上手く抽出することができた. 一方, *IDF* の値を用いた場合は広告の単語や本当に珍しい固有名詞ばかりを拾ってしまい, ほとんどそれらしい単語を拾うことができなかった. また今回 URL のリストを取得するために用いた Yahoo! 検索 API の提供するフィルタにより html のみを取得するように指定しているので, 通常の検索エンジンであれば検索結果として返される PDF や ms ワード 等のファイルがヒットしないのに加え, 通常の Yahoo! Japan のポータルでの検索結果と結果が微妙に違い, 動画のページがヒットしにくい等, 有利な環境となっていた.

今回は 20 件でのクラスタリング結果なので, 精度にそれほど差が出ていないが, クラスタリング精度は全体的に非常に低く, 抜本的な方法の改善が必要と思われる. しかし, 正解クラスタをどのように作成するかで結果が変化するので, 評価値は絶対ではない. 自分以外の人に正解クラスタを作成してもらい, それを教師データとして学習できるような汎用性の高いモデルが望ましい.

7 まとめ

単語の大域的な重要度を Wikipedia から計算してページの内容から *TFIDF*, *IDF* の値を求めて, 排他的クラスタリングを行い, その精度を評価した.

また, RDF として出力することで, 出力された形式を生かすようなアプリケーションを通して, ただページを表示するだけでなくページの主要な部分にフォーカスしてページを閲覧できるようにした.

8 今後の課題

今回の解析では, html の解析のみに絞っていたため, 一切解析ができないページはなかったが, javascript を用いて広告を挟むページ等, 構造的に有効な特徴語を拾うことができないページがあったので, それらも含め解析できるようにすることである. また, Web 上には PDF, ms ドキュメント等, 様々なデータ形式があるので, それらも含めてより正確な解析することである.

また, 評価が手作業に依存している部分があり, 評価と改善のプロセスが回しづらいので, 評価の全自動化を考えることも大きな課題である.

参考文献

[1] 成田宏和, 太田学, 片山薫, 石川博. Web 文書検索のための非排他的クラスタリング手法の提案. *2P-01, Data Engineering Workshop 2003*, 2003.

[2] 村松亮介, 福田直樹, 石川博. 分類階層を利用した検索エンジンの検索結果の構造化とその提示方法の改良. 電子情報通信学会第 19 回データ工学ワークショップ, B, Vol. 6, , 2008.