5C-1

文字列間の前処理付きオフライン全文検索エンジン類似度距離

佐藤 哲 †

楽天株式会社 †

1. はじめに

昨今の情報社会の中で，記号列間の表記的な類似性に留まらず，記号列の意味を考慮した類似性の測定方法が求められている．例えば情報量の多いインターネット上から望む情報を探し出すには，提示した例（検索キーワードなど）と意味的に似ている情報が自動的に提示されることが望ましい．そのため我々は，Web 上の辞書サイト Wikipedia のデータを利用して文字列の意味を考慮し，文字列間の類似度を計算するシステムを試作したが，記憶容量コスト，計算コストが高いことが問題であった．そこで本発表では，前処理により日本語の問題の解決や情報量圧縮などを行い，その後高速に文字列間の意味的な類似度距離を計算するシステムを提案する．

2. 文字列間の意味的な類似度距離計算

文字列間の距離について，従来は表記の違いを考慮するものが中心であり，現在においても入力ミスの検出など多くの分野で使用されている [1]．しかし，それらは文字列の意味の類似性を考慮したものではない．文章の構造を考慮した類似度の計算法としては，WordNet あるいは日本語 WordNet を利用した研究がある [2][3]．ただし本研究では，構造の情報が乏しい短い文字列同士の類似度の推定も可能とするため，単語の関連語が蓄積されていると考えられる Wikipedia コンテンツを利用する．

類似度を計算する式は次の正規化 Lucene 距離（Normalized Lucene Distance:NLD）である：

$$NLD(x,y) = 1.0 - e^{-\alpha NGD(x,y)}. \qquad (1)$$

ただし

$$\alpha = \log\left(\frac{\min(f(x), f(y))}{f(x,y)}\right) \qquad (2)$$

である．また，

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \qquad (3)$$

であり [4]，N は検索対象の総ページ数を，$f(x)$ は単語 x を検索エンジンにて検索した場合のヒット数を，$f(x,y)$ は 2 単語を同時に検索した場合のヒット数を表す．検索エンジンは，データ量が十分に多ければ Web 上のオンライン検索エンジンである必要はない．本研究では，検索エンジンとして Apache Lucene[1] を用いている．Lucene はオフラインの検索エンジンであるため，検索対象のコンテンツは編集やカスタマイズが可能である．

3. 正規化 Lucene 距離計算への前処理の導入

本研究では，以下のようなシステムを作成した．

検索対象とするコンテンツとしては，Wikipedia 日本語版の XML 形式データ [2] を用いる．Wkipedia の XML データからはプログラミング言語 Ruby の XML パーサライブラリである Nokogiri[3] を用いて検索に必要なデータを抽出される．また，英語にはない日本語の問題である単語の区切りを検出するための形態素解析の辞書には NAIST-JDIC[4] を，形態素解析エンジンには Igo-Analyzer[5] を採用した．

かつて我々は，これらの採用技術の下で，次のような処理を行うシステムを構築した [5]：

(1) Nokogiri により Wikipedia XML よりコンテンツを抽出
(2) Lucene により，検索インデックスを作成
(3) Lucene 及び Igo-Analyzer により Wikipedia コンテンツを検索し，式（1）により距離を計算

このシステムを 2011 年実装と呼ぶことにし，その評価結果は単語間の類似度を推定する目的には良好な結果が得られたが，Wikipedia データの検索インデックスが巨大で記憶容量コストが高く，それに伴い検索時間の長くなり類似度距離の計算コストが高くなることが課題であった．

そこで本発表では，次のように前処理を強化し最適化された検索インデックスを作成することで各種コストを削減しようと試みた：

(1) Nokogiri により Wikipedia XML よりコンテンツを抽出
(2) Igo-Analyzer により Wikipedia コンテンツを形態素解析し，Wikipedia 1 ページの分かち書きされた形態素を一意化したのち，Lucene により検索インデックスを作成

Similarity estimation between the words using off-line normalized search engine distance with preprocessing
†Tetsu R. Satoh, Rakuten Inc.

[1] http://lucene.apache.org/
[2] http://dumps.wikimedia.org/jawiki/latest/jawiki-latest-pages-articles.xml.bz2
[3] http://nokogiri.org/
[4] http://sourceforge.jp/projects/naist-jdic/
[5] http://es.sourceforge.jp/projects/igo/

表 1: NLD 実装差

項目	2011 年実装	2012 年実装
前処理 (1)	XML 解析	XML 解析
前処理 (2)	index 作成	形態素解析
前処理 (3)		形態素一意化
前処理 (4)		index 作成
距離計算 (1)	形態素解析	ヒット数調査
距離計算 (2)	ヒット数調査	距離計算
距離計算 (3)	距離計算	

(3) Lucene 及び英文用のアナライザにより Wikipedia コンテンツを検索し，式（1）により距離を計算

この今回紹介するシステムは 2012 年実装と呼ぶことにする．ここで形態素の一意化とは，同じページに複数回現れた同じ形態素を一つとしてファイルに書き出す処理のことを指す．

簡易なアイディアであるが，式 (1) がページのヒット数を元に計算され，ページ内の単語の複数ヒット数には関係ないため，重複形態素を削減して検索インデックスのサイズを縮小したことで記憶容量・計算量の削減を図ろうとしたシステム実装である．検索インデックスファイルののサイズ縮小と共に，検索時に日本語形態素解析を行わないことも高速化に寄与している．形態素解析は，検索インデックスを作成する直前に行われ，その結果が重複形態素削減に用いられる．過去の実装と今回の実装の比較を表 1 に示す．以後，2011 年実装を NLD_1，2012 年実装を NLD_2 と表記する．

4. 文字列間類似度測定比較実験

本節では，NLD_1 と NLD_2 の精度及び速度の比較を行う．実験は，単語データとして楽天ジャンルの「レディースファッション」「パソコン・コンピュータ」の中からそれぞれ 7 つずつ単語を抽出し，全単語間で類似度距離を計算した．計 14 単語は次のようなものである：ワンピース, ドレス, トップス, アウター, スーツ, 浴衣, 水着, パソコン, プリンタ, ディスプレイ, モニター, マウス, スキャナ, ハードディスク．式（3）中のパラメータ N は 1000000 である．

図 1，図 2 に，それぞれ NLD_1，NLD_2 により計算した距離のグラフを示す．X 軸及び Y 軸は，座標 0 から 6 がレディースファッションに属する単語，座標 7 から 13 がパソコン・コンピュータに属する単語であることを示している．後述するように大きく記憶容量の削減を行ったにもかかわらず，図より 2 つのジャンルが識別でき，両者はほぼ同等の精度の結果が得られていることが分かる．

記憶容量コストの面では，表 1 中の形態素一意化の処理により，検索対象テキストファイルサイズが 4.5G バイトから 1.8G バイトへと約 2.7G バイトの大きな縮小を達した．また，そのテキストファイル

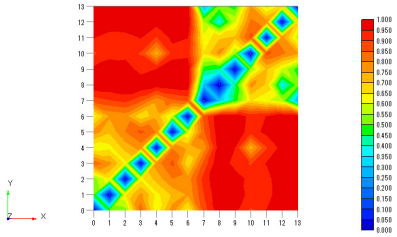

図 1: NLD_1 による類似度測定

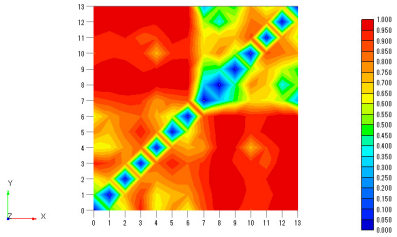

図 2: NLD_2 による類似度測定

の Lucene 検索インデックスのサイズは，2.3G バイトから 830M バイトへと約 1.5G バイト縮小できた．

文字列の検索にかかる時間は，NLD_1 が平均 1400 ミリ秒，NLD_2 が平均 1296 ミリ秒と，平均で 1 回の検索時間が約 104 ミリ秒短縮できた．今回の実験では実施していないが，Lucene 検索インデックスサイズを縮小できたことにより，インデックスをメモリ上に配置することができるため，チューニング次第でさらに高速化できる可能性があると考えられる．

実験に用いた計算機は CPU が Pentium M(1.6GHz) のノート PC で，OS は FreeBSD 8.2，Lucene の実行には Diable Java 1.6.0 を用いた．

5. おわりに

本発表では，提案済みである文字列の意味を考慮した類似度距離を計算するシステムに対し，記憶容量コスト及び計算コストを下げる改良案を提案した．

参考文献

[1] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, Vol. 33, pp. 31–88, 2001.

[2] T. Wang and G. Hirst. Refining the notions of depth and density in wordnet-based semantic similarity measures. *Proc. 2011th Conf. Empirical Methods in Natural Language*, pp. 1003–1011, 2011.

[3] 福岡知隆, 服部峻, 久保村千明, 亀田弘. 単語間の意味カテゴリー距離に基づく用例ベース未知語意味カテゴリー推定. 第 10 回情報科学技術フォーラム, Vol. 2, pp. 323–324, E-047 2011.

[4] R. L. Cilibrasi and P. M. B. Vitányi. The google similarity distance. *IEEE Trans. Knowledge and Data Engineering*, Vol. 19, No. 3, pp. 370–383, 2007.

[5] 佐藤哲. オフライン全文検索エンジンを用いた文字列間の正規化類似度距離. 第 10 回情報科学技術フォーラム, Vol. 2, pp. 397–398, F-008 2011.