情報検索の類似尺度を用いた検索要求文の単語分割

小澤智裕　山本幹雄（筑波大学）
山本英子　梅村恭司（豊橋技術科学大学）

1．はじめに

現在の情報検索システムは単語(キーワード)を中心とした理論・実装が主流である。しかし、日本語では単語の概念が明確ではないため、さまざまな問題が生じる。そもそも人間が自然だと思う単語分割が情報検索にとって有用であるかはまだ結論の出ていない問題である。中国語の研究[Chen97]では、漢字のbigramが単語に区切る方法と同程度の性能を持つことが示されている。

本稿では、現在標準の類似度であるIDF(Inverse Document Frequency)付きVector Spaceモデルを尺度とし、これを最大化する検索要求文の単語分割を求める情報検索システムを検討する。実験結果を通して、本システムは形態素解析等であらかじめ単語分割を行なうシステムに比べ情報検索の精度が高く、また、分割結果は人間が見て自然であることを報告する。

2．キーワードに基づく情報検索

2.1 Vector Spaceモデル

情報検索では、検索要求と個々の検索対象との類似度をなんらかの基準で定義し、類似度の近いものから順位を付けて表示する。キーワード、あるいは単語に基づく情報検索における代表的な類似度はVector Spaceモデル[Salton83]を利用したものである。この方法は、検索要求と検索対象をそれぞれある空間上のvectorとして表現し、2本のvectorの内積あるいは角度のcosineで類似度を定義する。もっとも簡単なモデルでは、キーワードごとに空間の次元を設定し、vectorの各次元の値は検索要求、検索対象が含むキーワードの個数にそのキーワードの重要さの重みをかけたものである。各キーワードの重みはIDF(inverse document frequency)と呼ばれる値が用いられる。検索要求文と検索対象としてのあるドキュメントをそれぞれq, dとすると、2つの類似度 $sim_k(q,d)$ は以下のような式で定義される。

$$sim_k(q,d,T_q) = \frac{1}{W_d}\sum_{t\in T_q} tf(t,q)\cdot tf(t,d)\cdot idf(t) \quad (1)$$

ここで、

W_d = d の正規化係数（例えば、d の長さ）
T_q = q 中のキーワードの集合
$tf(t,q)$ or $tf(t,d)$ = q または d 中の t の出現回数
$idf(t) = \log(1 + D / df(t))$
$df(t)$ = t を含むドキュメントの数
D = ドキュメントの総数

である。ここで、T_q は形態素解析等で計算前に決定される。W_d として d の長さをとった場合、この類似度は「検索要求中のキーワード、特に重要度の高い（IDF値の大きい）キーワードを多く含み、かつドキュメントの長さが短い」検索対象を検索要求に類似していると見なす。

2.2 Vector Spaceモデルの日本語への適用と問題点

2.1節で見たように、Vector Spaceモデルはキーワードに基づく類似度である。日本語の場合、英語とは異なり単語境界が明示的でないので、最初の段階で単語分割処理が必要となる。単語分割を行なうためには、形態素解析システムを用いることが多い。日本語を対象とした一般的な情報検索システムを構築するには、まず検索対象としてのドキュメントを形態素解析し、名詞等のキーワードを抽出して索引を作成する。その後、ユーザからの検索用キーワードを用いて、対象ドキュメントを絞りこむ。

最近は形態素解析システムの精度も向上しているがまだ完全ではない。特に辞書にない未知語の部分に対して誤る傾向が強いが、不幸にして、未知語は固有名詞等の重要なキーワードになる場合が多い。標準的な辞書を用いた形態素解析をしている限り、この問題は解決できない。

また、ユーザの入力するキーワードと形態素解析で抽出した検索対象のキーワードが食い違う場合が生じる。例えば、長い複合語はどこまで、あるいはどの部分をキーワードと見なすかにはっきりした基準はない。唯一の形態素解析結果から索引を作成し、ユーザが入力したキーワードをそのまま用いた場合、この食い違いによって精度が劣化する。これを避けるために、長い複合語は分解して、その構成要素を索引に登録し、ユーザの入力したキーワードも長いものを分解してキーワードとして用いる方法が一般的である。しかし、短い単語は検索対象を絞

る力に弱く、分解をやりすぎると無用のドキュメントが検索されてしまうという大きな欠点がある。この問題は、キーワードと共にキーフレーズ（句）を要求入力として許す場合、英語でも同様に問題となる。

3．全ての部分文字列を使った類似度

3.1 類似度を最大とする単語分割

2節で述べた従来法の問題は、検索対象のドキュメントをあらかじめ決められたキーワードに分割する必要があったため生じている。本節では、検索対象のすべての部分文字列をマッチング可能なキーワード候補と考え、かつ検索要求としてのキーワードリスト、あるいは文の任意の分割の中で(1)式を最大とする分割がその検索対象に対する最適の分割であるとする方法を提案する。この方法では、検索対象ごとに検索要求の単語分割が異なる。類似度 $sim_s(q,d)$ は(2)式で定義される。

$$sim_s(q,d) = \max_{s \in S(q)} sim_k(q,d,T(s)) \qquad (2)$$

ここで、

$S(q)$ = qの可能なすべての単語分割の集合
$T(s)$ = 単語分割sから単語の集合に変換する関数

である。すなわち、qに対する単語分割の全集合の中で、(1)式の類似度を最大とする分割をドキュメントdに対する正しい分割とみなし、そのときの(1)式の類似度をqとdの類似度と定義する。ここで、$T(s)$の関数の選び方でいくつかのバリエーションが生じる。例えば、IDF値が小さい単語はキーワードとして対象記事を絞りこむ力が弱いので、削除する。あるいは、長さ1の単語（すなわち文字）は、一般にキーワードとしてよくないので削除する等である。また、実験結果から(2)式のままでは長さ1の単語が優先される傾向があったので、4節の実験結果は部分文字列の重みをidf(t) * length(t)として長い部分文字列を優先するヒューリスティックスを入れたものである。

実際の検索においては、すべての検索対象としてのドキュメントに対して、(2)式を計算する必要がある。単純な実現方法では膨大な計算時間を必要とするが、Suffix Arraysデータ構造[Manber&Myers93]とViterbiアルゴリズムによって以下のように計算時間を削減できる。

（1）すべてのドキュメントをまとめて、Suffix Arrayを構成する。
（2）qの可能な分割を表現する半順序構造を構成する。
（3）半順序構造中の各単語候補を部分文字列として含むドキュメントをSuffix Arrayを用いて検索する。これは、Nをすべてのドキュメントの合計の長さとしたとき、O(logN)の時間で計算できる。このとき、qの単語候補を少なくとも1つ含むドキュメントのリストを同時に作成する。
（4）qの単語候補を少なくとも1つ含むドキュメントのそれぞれに対して、qの半順序構造中で(2)式の値をViterbi探索で求める。最大値となるパスが、ドキュメントに対する最適なqの分割を意味している。
（5）各ドキュメントの類似度をソートし、大きい順に出力する。

3.2 最適分割の近似

前節の手法で、かなり計算量は削減できるが、まだ現実的な応用には計算量が大きすぎる。ここでは、計算コストを削減するために検索対象全体を用いて、検索要求文の1つの分割をまず求め、それを使って従来と同様の方法で検索を行なう方法を述べる。(1) 式を最大化するにはIDF値の大きなキーワードを抽出できればよいことが分かる。これから、分割された各単語のIDF値の合計が最大になるような分割が検索にとってよい分割である可能性は高い。各部分文字列のIDF値はSuffix Arrayを利用した[Yamamoto&Church98]の方法を利用して高速に計算できる。より長めの部分文字列を優先するヒューリスティックスを入れると、準最適な分割 $\hat{s}_q$ は以下のように定義される。

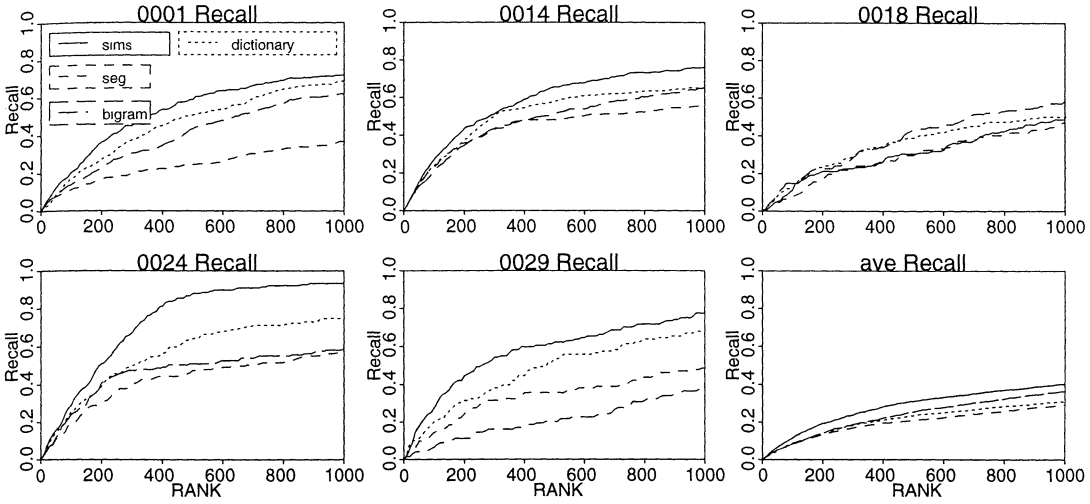
$$\hat{s}_q = \arg\max_{s \in S(q)} \sum_{t \in s} idf(t) \cdot length(t) \qquad (3)$$

ここで、

$length(t)$=tの長さ

である。実験によると、length(t)を掛けても、なお長さ1の文字が優先される傾向があったため、4節の実験では実際にはlength(t)-1を掛けている。また、idf値は単に頻度の低い部分文字列に対して大きいため、これを出現頻度で補正したResidual IDF値[Church & Gale 95]を実際には用いている。

(3)式による分割を行った後で、(1) 式を使った普通のVector Spaceモデルによる検索を行なう。ただし、対象記事はあらゆる部分文字列がマッチング対象となり得るので、Suffix Arrayを用いた全文索引が必要である。この点は従来法と異なる。



4．評価実験

4.1 情報検索による評価

NACSISコレクション[Kando97]を用いて情報検索の評価を行った。NACSISコレクションは約33万件の学会論文等のアブストラクトのデータベースである。現時点で、30課題の検索要求文がとその正解文献が定義されている。30課題から比較的正解文献数が多い課題を５つ選び、検索文献数に対するrecallのグラフを図１に示す。各グラフは各課題ごとの結果を示している。縦軸がrecall、横軸が検索した文献の数である。実際にはPrecisionとRecallの値を求めたが、２つは同じ傾向を示していたため、図１にはrecallのみを載せた。比較手法は、以下の通りである。

(1) 提案手法１（ドキュメントごとに分割を可変させる手法。3.1節。図１中ではsims）
(2) 提案手法２（最初に検索要求文を分割する方法。3.2節。図１中ではseg）
(3)バイグラム (すべてのバイグラムをキーワードとした従来法。2節。図中ではbigram)
(4)従来法 (形態素解析を行ってキーワードを抽出した従来法。2節。図中ではdictionary)

図１のaveは５課題の平均を意味しており、この図よりsims、bigram、dictionary、segの順でよいことが分かる。simsの方がbigramより優った原因としては、日本語の場合は仮名の部分に関して、bigramでは記事を絞りこむ力が弱いのではないかと想像される。辞書を用いた形態素解析を用いた従来方法はさらに悪い。これは、NACSISコレクションが学術論文のアブストラクトを集めたものであり、検索要求文に長い複合語からなる専門用語が含まれているためであると考えられる。提案手法は検索対象としてのNACSISコレクション全体を辞書として使用していると考えられ、比較的有効な単語分割を行なうのに対して、辞書に基づく形態素解析は専門用語を過剰分割してしまい、個々の単語が有効な検索キーにならなかったと考えられる。部分文字列のridf(t)*length(t)の合計を最大とする分割を形態素解析として捉える方法は辞書をつかわない分割にもかかわらず辞書を使ったものに迫る精度を上げている。

4.2 単語分割結果

次に提案手法がどのように単語分割を行ったかを見ていく。本論文で述べた２つの手法(式(2), (3))は、Vector Spaceモデルを尺度とした検索要求文の自動単語分割手法とも考えられる。図２は検索要求文の一部であり、この文に関して(2)式の類似度を計算する際に抽出され、検索に使用された部分文字列（長さ２以上）を図３に示す。(a)と(b)は異なる文献に対する結果である。文献Aでは、「自律移動ロボット」という言葉が出ていないため、構成要素の名詞に分割されて使用されているが（実際、「自律行動ロボット」という単語が文献Aでは使用されていた）、文献Bではこの言葉が出てくるのでそのまま使用されている。文献Bの方が類似度が大きくなっている。図４は、式(3)（ただし、ridf(t)*(length(t)-1)の合計を最大化する）で求めた図２の文に対する分割である。部分文字列の左の数値はridf(t)*(length(t)-1)の値である。よいキーワードとなりえるものは、ある程度人間の直観と合うような

「または、自律移動ロボットにおける部分的なシステム(経路制御、物体認識など)の設計について書かれた文献が検索要求を満たす。」

図2　検索要求文の一部

26.5192: 自律
6.3561: 移動
52.8893: ロボット
4.6111: にお
2.7726: ける
15.0408: システム
8.9773: 経路
2.3472: いて
2.1972: れた

(a) 文献Aに対する部分文字列

168.1886: 自律移動ロボット
7.7972: 制御
6.9315: 認識
7.4404: について

(b) 文献Bに対する部分文字列

図3 (2)式の最大化によって抽出された部分文字列

0.3486: または
0.0000: 、
1.9217: 自律移動
4.0810: ロボット
0.5370: における
0.4938: 部分
0.2268: 的な
2.9401: システム
0.0000: (
3.4069: 経路制御
0.0000: 、
1.2294: 物体
1.1683: 認識
0.2245: など
0.1576:)の
0.8551: 設計
0.2408: について
0.1794: 書か
0.5305: れた文献
0.0000: が
1.0719: 検索
0.3790: 要求
0.3975: を満たす
0.0000: 。

図4 (3)式の最大化による単語分割

分割がなされている。また、分割がおかしく意味をもたないものは（「的な」や「)の」など）、比較的小さなridf(t)*(length(t)-1)の値を持っており、この値が小さな部分文字列は検索キーとして使用しなければより高精度な検索が期待できる。実際、実験によるとridf値がある一定値よりも小さいものを無視すると、精度も向上し、かつ高速になることが示されている。

5．おわりに

情報検索システムの類似度を使用して、情報検索の性能を上げる単語分割という考え方を示した。提案手法では、辞書等の人間が与えた言語学的な知識を一切使わずに情報検索の類似尺度のみを用いて単語分割が自動的になされる点に特徴がある。提案手法は、辞書を用いた形態素解析を用いる方法、あるいはbigramを用いる方法よりも高精度であることが評価実験により示された。また、この結果得られた分割は人間にとっても自然な分割といえるものであった。情報検索システムに効果的な分割は数式で与えられた理論的分割であり、自然な分割は人間の情報処理の結果である。これが一致したことにより、人間の考える自然な分割についての一つの意味付けを与える結果となっている。

謝辞　本研究においては学術情報センターのテストコレクション1（予備版, コンペティション参加者用）を使用しました。関係者に感謝いたたします。

参考文献

[Chen 97] Aitao Chen, Jianzhang He, Liangjie Xu, Fredric C. Gey and Jason Meggs, Chinese text retrieval without using a dictionary, In proceedings of SIGIR'97, Philadelphia PA, USA, pp.42-49, 1997.

[Church & Gale 95] Church, K. and W. Gale, Poisson mixtures, Natural Language Engineering, 1(2), pp.163-190, 1995.

[Kando 97] Kando, N. et al., NTCIR: NACSIS test collection project, 20th Annual Colloquium of BCSIRSG, Autrans, France, March 25-27, 1997.

[Salton83] G. Salton and M.J. McGill, The SMART and SIRE Experimental Retrieval Systems, pp.118-155, New York: McGraw-Hill, 1983.

[Manber&Myers 93] Udi Manber and E. Myers: Suffix array: a new method for on-line string searches, SIAM Journal on Computing, 22:5, pp.935-948, 1993. http://glimpse.cs.arizona.edu/udi.html.

[Yamamoto&Church 98] M. Yamamoto and K.Church: Using suffix arrays to compute term frequency and docuement frequency for all substrings in a corpus, In proceedings of 6th Workshop on Very Large Corpora, Ed. Eugene Charniak, Montreal, pp.28-37, 1998.