情 報 学 基 礎　59－2
デジタルドキュメント　24－2
（2000．7．28）

情報検索性能と表記の揺れへの寛容性を持つ類似度

山本 英子[†] 武田 善行[†] 梅村 恭司[†] 山本 幹雄[‡]

[†] 豊橋技術科学大学 情報工学系　[‡] 筑波大学 電子・情報工学系

本論文では，情報検索に利用でき，かつ表記の揺れに寛容な類似度を提案する．表記の揺れに対応することができる編集距離という手法があるが，この手法では情報検索精度が弱いことが知られている．そこで，本論文では，情報検索の性能を持ち，かつ表記の揺れにも対応することができるダイナミックプログラミングを用いた類似度の計算法を提案し，その情報検索性能が単語に基づく手法と ngram に基づく手法と比較した結果，効果的であり，かつ提案した手法が表記の揺れに寛容であることを報告する．

A Similarity Measure Suitable for Information Retrieval and Tolerant for Morphological Variation

Eiko Yamamoto[†] Yoshiyuki Takeda[†] Kyoji Umemura[†] and Mikio Yamamoto[‡]

[†] Department of Information and Computer Sciences, Toyohashi University of Technology

[‡] Institution of Computer Sciences and Electronics, University of Tsukuba

In this paper, we propose a similarity measure suitable for information retrieval and tolerant for morphological variation. Edit distance is well-known similarity measure that can cope with variations. Unfortunately, edit distance is not suitable for information retrieval due to its performance. We have improved The behavior of edit distance by extending its definition. We have compared the proposed similarity measure with the popular similarity measures for information retrieval.

1. はじめに

多くの情報検索手法では，二つの文字列が似ているか似ていないかを調べることと等価な操作を行なう．また，もう一方で文字列の意味の解析も進んでいるが，意味付けの解析は難しく，コストが高い．また，技術文書のような新しいないようを扱う場合には，特にコストが高い．なぜなら，意味付け処理のために辞書を用意しなければならなかったり，次々と新しい用法や意味が現れるような検索対象を考えた場合，このような深い意味付け処理を行なうためのシステムを調整するためにコストがかかり，また即時的でもないからである．一方，部分文字列の対応をとる操作は，非常に簡単な操作であるにもかかわらず，ある程度の情報検索精度を持つことが知られている．この操作は，単に二つの文字列に対して，一方の文字列に含まれる部分文字列がもう一方の文字列にも含まれるかどうかという単純な文字列の比較を行なう操作なので，言語が時代や場所とともに変わるために起こることであるために流動的であったり，局所的に性質が異なったりする状況の変化によって，性能を左右されにくい．

このような背景から，本研究でも文字列の対応をとる操作を用いて，情報検索を行なうことを考えた．文字列の対応をとる場合，二つの文字列の類似度を計算するための類似尺度が必要である．したがって，用いる類似度によって情報検索性能が大きく左右される．このように，情報検索性能というのは，客観的な類似尺度の評価である．情報検索は，TREC, NACSIS[6),7)]，ETC などの多くのコンテストが開催されており，評価方法が確立している．

一方，日本語では，情報検索に役立つ単語列の変形が多く存在する．技術用語などの新しい用語では変形が生じることが多い．単語列の変形の多くは，日本語の表記の揺れによるものである．そこで，本研究では，このような表現上の子細な揺れに対して寛容な類似度がほしいと考えた．その表現の揺れに寛容である類似度を用いることによって，変形が生じている場合でも二つの文字列を似ているかどうかを判定することができる．

これまでの研究で，表記の揺れに対応することができる有名なものに編集距離[2)]という手法がある．その応用として，編集距離は DNA 解析やスペルチェックなどに広く使われている．しかし，この手法の情報検索性能を測定したが，高くないことが判明した．したがって，新しい類似度が必要となる．

そこで，本論文では，編集距離を改良し，ある程度の情報検索性能を持ち，かつ表記の揺れにも対応することができるダイナミックプログラミングを用いた類似度の計算法を提案する．そして，その性能を情報検索という客観的

な評価方法で評価した．その結果，本論文で提案する類似度を用いた手法が単語に基づく手法やngramに基づく手法よりも効果的であり，かつ提案した手法が表記の揺れに寛容であることを報告する．

この論文の構成は次のとおりである．2節，3節，4節では，本論文で提案する三つの類似度の定義を一つずつ示す．5節では，情報検索性能を測るための実験方法と，比較対象となるベースラインシステムについて説明した後，実験結果を示し，実験結果からの性能評価と考察を述べる．6節では，ベースラインシステムと提案する方法との表現上の揺れに対する振舞いの違いについて考察し，最後にまとめる．

2. 単純編集類似度

前節で述べたように，表記の揺れに対応することができる手法として有名なものに編集距離がある．これは，二つの文字列の距離として，文字の削除，挿入，置換を距離の加算として考える方法である．この距離はダイナミックプログラミングで求められる距離である．この関数は，全く共通文字のない二つの文字列の場合，長い方の文字列の長さを値とする．しかし，本研究では，全く共通文字のない場合を統一的に関係がないものとして扱いたいので，どちらか長い文字列の長さから，編集距離を引いた値を考え，これを単純編集類似度とした．つまり，全く共通文字がない場合，この値は0となる．編集距離と等価な単純編集類似度は以下のように定義できる．

定義 2.1 α, β を文字列とし，x, y を異なる文字とする．“”は空文字列とする．

- 両方とも空文字のとき
  $SIM_1(\text{“”}, \text{“”}) = 0.0$
- 長さ1文字以下の異なる文字列のとき
  $SIM_1(x, y) = 0.0$
- 先頭の1文字が同じとき
  $SIM_1(x\alpha, x\beta) =$
  $MAX(SIM_1(\alpha, x\beta),$
  $SIM_1(x\alpha, \beta),$
  $SIM_1(\alpha, \beta) + 1.0)$
- 先頭の1文字が異なるとき
  $SIM_1(x\alpha, y\beta) =$
  $MAX(SIM_1(\alpha, y\beta),$
  $SIM_1(x\alpha, \beta),$
  $SIM_1(\alpha, \beta))$

編集距離と定義は同一ではないが，形式は同じであり，この類似度もダイナミックプログラミングで計算することができる．

3. 文字重み編集類似度

単純編集類似度では，すべての文字は同等に扱っていたが，例えば，ひらがなの一文字の差と漢字の一文字の差を同じ重みとすることは不自然である．例えば，「は」，「が」，「を」，「の」，「で」などの助詞や「〜する」，「〜である」，「〜した」，「〜でない」，「〜しない」などの機能語を構成するひらがなは検索の際のキーワードとしては役に立たないとみなせるが，漢字は検索の際に有用なキーワードの一部に比較的なりやすいため検索に有効であるといえるからである．したがって，文字が一致していることによる類似の寄与を文字に関する関数で与えることにする．

2節で定義した単純編集類似度と文字重み編集類似度の違いは加算に与える値である．文字重み編集類似度ではそれぞれの文字の重みである定数が与えられる．つまり，文字重み編集類似度は単純編集類似度の一般化となっている．文字重み編集類似度を以下のように定義する．

定義 3.1 α, β を文字列とし，x, y を異なる文字とする．“”は空文字列とする．

- 両方とも空文字のとき
  $SIM_2(\text{“”}, \text{“”}) = 0.0$
- 長さ1文字以下の異なる文字列のとき
  $SIM_2(x, y) = 0.0$
- 先頭の1文字が同じとき
  $SIM_2(x\alpha, x\beta) =$
  $MAX(SIM_2(\alpha, x\beta),$
  $SIM_2(x\alpha, \beta),$
  $SIM_2(\alpha, \beta) + Score(x))$
- 先頭の1文字が異なるとき
  $SIM_2(x\alpha, y\beta) =$
  $MAX(SIM_2(\alpha, y\beta),$
  $SIM_2(x\alpha, \beta),$
  $SIM_2(\alpha, \beta))$

ここでは，文字の重み関数として，助詞や機能語を構成するひらがなはほとんどすべてのドキュメントに出現するため，重みは低くなるものを想定している．このような性質を持つ重みはたくさん考えられるが，文字の一致を検出したときの情報量に相当するIDF(Inverse Document Frequency)は一つの候補である．

3.1 文字重みの効果

例として一致した文字に対する寄与が，次のように与えられる場合を考えてみる．

- x がひらがなの場合
  $Score(x) = 0.0$
- x がひらがな以外であった場合
  $Score(x) = 1.0$

ただし，x は長さ1の文字列(文字)であるとする．

表1に，文字重みの効果を示す．α と β の比較は文字に重みを付けなくてもうまくいく例である．SIM_1, SIM_2 共にすべての文字がマッチしている．どちらも「自動翻訳システム」に関する文章である．

一方，α と γ は文字に重みを付けないとうまくいかな

い例である．SIM_1ではひらがなや「的に」等といった機能語に対しても同じ重みでマッチングを行なっているため，SIM_2での値とは大きく異なっている．結果としてαとγは内容的にはそれほど類似しているものではないにも関わらず，SIM_1では大きな類似度を与えてしまっている．

このようなことからも一致した文字に対する寄与を文字の重みを用いて調節することで，人間の直観的な類似の感覚に近づく効果があることがわかる．ただし，重みとして何が適切であるかを考慮しなければならない．

4. 文字列重み編集類似度

文字重み編集類似度では，加算の対象となるものが文字であるため，文字の組合せによって，類似度を高く判定できる状況に対応できない．

例えば，ひらがな一文字からなる助詞や機能語などに関していえば，重みは低くするのが適当であるが，それらの組合せである「はしたない」などといった検索の際にキーワードになりうる文字列が双方に現れた場合，相応の重みを与えるのが適切である．そこで，合算方法を拡張して，文字列に関する重みを考え，その重みを用いて類似度を拡張したものが文字列重み編集類似度である．

文字列重み編集類似度は，長さ2以上の文字列の重みが0であるような重みについては文字重み編集類似度に一致する．また，長さ2以上の文字列の重みは0であると重みを拡張することで，同じ振舞いをする文字列重み編集類似度を作ることができる．つまり，文字列重み編集類似度は文字重み編集類似度の一般化となっている．

定義4.1 $\alpha, \beta, \gamma, \delta$を文字列とし，$\xi$を長さ1以上の文字列とする．また，$x, y$を文字(長さ1の文字列)とする．$Score$は文字列から実数値を求める関数，$MAX$は与えられた引数のうちもっとも大きいものを返す関数とする．

- 両方とも空文字のとき
  $SIM_3(\text{“”}, \text{“”}) = Score(\text{“”})$
- それ以外のとき
  $SIM_3(\alpha, \beta) =$
  $MAX(SIM_{3s}(\alpha, \beta), SIM_{3g}(\alpha, \beta))$
  - 一致している最大の文字列をξとして，
    $SIM_{3s}(\xi\alpha, \xi\beta) =$
    $MAX(Score(\gamma) + SIM_3(\delta\alpha, \delta\beta))$
    $for\ all\ \gamma;\ for\ all\ \delta; suchthat \xi = \gamma\delta$
  - そのような文字列が存在しないとき
    $SIM_{3s}(\alpha, \beta) = 0.0$
  - 任意の文字列について
    $SIM_{3g}(x\alpha, y\beta) =$
    $MAX(SIM_3(\alpha, y\beta),$
    $SIM_3(x\alpha, \beta),$
    $SIM_3(\alpha, \beta))$

ここで，文字列の重み関数としては，文字重み編集類似度と同様に，文字列の一致を検出したときの情報量に相当するIDFを使用することにした．そして，この重み関数で実際に情報検索の性能を確保することができたことを5章で報告する．

- 空文字ならば，
  $Score(\text{“”}) = 0.0$
- それ以外ならば，
  $Score(\xi) = -\log_2(df(\xi)/N)$

情報検索のシステムにおいては，この重みをさらに精密に調整することで性能が向上することが知られている．しかし，本論文では，最初の近似としてIDFを利用することは自然であると考えた．

5. 実　　験

本研究では，情報検索の問題を使って，提案する類似度の性能を評価することにした．情報検索では，話題が合致すれば正解と判定される．性能を比較するために行なった情報検索の実験は情報検索コンテスト[6),7)]に従うものとする．質問文はキーワードではなく文章で表され，30問，検索対象は各種学会のアブストラクト330,000件である．性能比較の実験において，ドキュメント集合と質問と正解の三つ組が与えられる．

5.1 提案した類似度間での性能比較

始めに，提案した三つの類似度，単純編集類似度，文字重み編集類似度，文字列重み編集類似度をそれぞれ用いたシステムで情報検索を行なった結果を表2に示す．表に示す値は30個の質問に対する11点平均精度(11 point average precision)であり，この値を使って三つのシステムの性能を比較する．この表は，単純編集類似度や文字重み編集類似度より，文字列重み編集類似度が効果的であることを示している．

表1　文字重みの効果

引数	SIM_1	SIM_2
α, β	8.0	8.0
α, γ	9.0	5.0

α：　「機械で自動的に翻訳するシステム」
β：　「自動翻訳システム」
γ：　「人手で直感的に表示するシステム」

表2　提案した類似度ごとの11点平均精度

SIM_1	SIM_2	SIM_3
0.0001	0.2032	0.2808

5.2 ベースラインシステム

5.1節で示したように，本論文で提案する三つの類似度

では，文字列重み編集類似度が最も効果的であることがわかった．次に，提案する文字列重み編集類似度を用いたシステムと比較する対象として，二つのベースラインシステムを紹介する．本研究では，ベースラインシステムとして，単語に基づくシステムとngramに基づくシステムを考慮している．

まず，単語に基づくシステムとしてBD法(baseline-dict)を示す．このシステムは類似度算出の前に，日本語形態素解析プログラム「茶筌」[5]を用いる．「茶筌」は大きな日本語の単語辞典を使って文字の順列を単語に区切り，品詞を割り当てるためのシステムである．本研究では，名詞，動詞，未定義語を用語として使うことにし，他の品詞の単語はストップワードとして考え，使用しないことにした．

単語を抽出した後，Salton[1]によって提案される$tf \cdot IDF$の重みを用いる内積スコアリング関数を使い，一致した単語の重み付けを行なう．

次にBD法で用いた類似度の関数の定義を示す．

定義5.1 tは比較される各々の文字列両方に現われる単語，$tf(t)$はそのドキュメントの単語tの出現頻度(term frequency)，$df(t)$は単語tが出現するドキュメントの数(document frequency)，Nはドキュメントの総数である．

$$SIM_{dict} = \sum_{t} tf(t) \cdot (-log_2(df(t)/N))$$

次に，文字に基づくシステムとしてBN法(baseline-ngram)を示す．このシステムのみに限らず，ほとんどの文字に基づくシステムは，処理に先駆けて質問とドキュメントを短く，できるだけ重ねながら，バイグラム(bigram)またはトライグラム(trigram)のようなngramに区切る．ここで，適切な長さを決定することは一般には難しい．

日本語では，ひらがな，カタカナ，漢字の三種類の文字が使われる．ひらがなは日本語の機能単語を表現するために使われる文字で50文字，カタカナは外来語を表現するために使われる文字で50文字，漢字は昔中国から来た文字で数千文字ある．また，カタカナは漢字よりかなり少ないので，カタカナで構成される単語は漢字で構成される単語より長い傾向がある．例えば，漢字で構成される単語「機械」や「翻訳」はどちらとも二文字であるのに対し，カタカナで構成される単語「システム」は四文字である．このことから，Fujii and Croft[3]が示したように，短いngramは漢字にはかなり効果的であるが，カタカナには効果的でないことが推定できる．したがって，短いngramは対象とする言語が中国語であるとき，より効果的であることが推定できる．なぜなら，中国語では使われる文字がすべて漢字だからである．実際に長いngramを考慮することは，複合語をマッチングを行なう情報検索[9]の報告で，共起情報を用いないケースに相当する．DP法も長いngramを検出している．そこで，本研究では，長さが多様であることを考え，用いたBN法は質問にあるすべての部分文字列について，一致判定と重みを計算することにした．それぞれの部分文字列について，標準的な内積尺度を使って$tf \cdot IDF$の重みとスコアを計算する．このようにすることによって，提案する文字列重み編集類似度での条件との比較ができるようになる．違いは，合算に加算を使う(BN法)か，ダイナミックプログラミングを使うかである．

以下にBN法で用いた類似度の関数の定義を示す．Score関数は一致した部分文字列の重みであり，本研究では，典型的なIDFに基づく関数を使う．$df(\xi)$は部分文字列ξの出現ドキュメントの数である．通常，dfは単語を対象とするが，ここでは部分文字列を対象とする．

定義5.2 α, β, ξ, ηを文字列とする．α_{ik}をi番目の文字から$i+k-1$番目の文字までのαの部分文字列とし，β_{jk}をj番目の文字から$j+k-1$番目の文字までのβの部分文字列とする．また，$Score$は4節に示したものと同じものとする．

$$SIM_{ngram} = \sum_{i,j,k} Comp(\alpha_{ik}, \beta_{jk})$$

ただし，$Comp(\xi, \eta)$は次のように定義される．

- $\xi = \eta$ならば，$Score(\xi)$
- $\xi \neq \eta$ならば，0.0

BN法は，形態素解析も使用せず，文字のレベルで一致をとることためノイズが混入しても情報検索ができると考えら，表記の揺れに寛容であるかどうかも検討を行なう．

5.3 情報検索性能

この節では，本論文で提案する文字列重み編集類似度を用いる手法(DP)，BD法(BD)，BN法(BN)の三つのシステムの結果を報告する．

表4はそれぞれのシステムを二つずつ各質問について，表3に示す11点平均精度(11 point average precision)を使って比較し，すべての質問について数値で判定した結果から作られたものである．この表は，DP法がBD法やBN法より精度が高いことを示している．

5.4 BD法との比較

簡単な情報検索の質問については，それぞれのベースラインは同じような性能を示す．質問1は，用語と用語を構成する単語の多くがそれらのIDF重みによって示されるようなよいキーワードである用語を含む質問であり，すべてのシステムにとって簡単な質問である．図1にトップランクドキュメントに関する再現率を示す．このことにより，それぞれのベースラインは比較の条件が揃えられていることがわかる．

形態素解析のようなシステムはシステムに関して未知の単語が重要となる質問では問題がある．図2はそのような質問に関する再現率のグラフである．BD法はデータ

表3　それぞれのシステムの質問ごとの11点平均精度

質問番号	BD	BN	DP	質問番号	BD	BN	DP
1	0.2462	0.2216	0.3006	16	0.0180	0.3275	0.8263
2	0.2099	0.5022	0.6704	17	0.0471	0.0011	0.0407
3	0.0257	0.0028	0.0148	18	0.0788	0.0051	0.0980
4	0.1048	0.1815	0.3136	19	0.2091	0.4901	0.6890
5	0.1046	0.0008	0.0013	20	0.5660	0.5705	0.7775
6	0.0580	0.0221	0.1890	21	0.0881	0.0713	0.1583
7	0.0005	0.0010	0.0024	22	0.0516	0.0703	0.2493
8	0.0836	0.0086	0.2119	23	0.2003	0.1475	0.3481
9	0.0157	0.0372	0.2402	24	0.1915	0.2887	0.3234
10	0.3751	0.2991	0.3618	25	0.6076	0.2291	0.4877
11	0.2533	0.0303	0.2638	26	0.1087	0.4665	0.5136
12	0.1008	0.1687	0.1566	27	0.0659	0.0696	0.2556
13	0.0707	0.0150	0.0655	28	0.0072	0.0435	0.0589
14	0.4107	0.3051	0.4257	29	0.1543	0.0882	0.1151
15	0.1538	0.1436	0.2116	30	0.0094	0.0142	0.0527

表4　DP法はBD法やBN法より高精度である.

X vs Y	X win	Y win
DP vs BD	23	7
DP vs BN	29	1
BD vs BN	15	15

表5　DP法はBN法より表記の揺れに寛容である.

順位	DP	BN	文字列
1	1.000	1.000	機械翻訳システムの出力を人間が編集することが必要である.
2	0.843	0.625	機械翻訳のシステム出力を人間が編集することが必要である.
3	0.708	0.561	機械翻訳結果を人間が編集することが必要である.
4	0.699	0.397	機械による自動翻訳システムの出力は, 人間が編集することを必要とする.
5	0.696	0.536	機械による翻訳のシステムのあとに人間が編集することが必要である.
6	0.695	0.658	機械翻訳システムの出力を人間の手を加えることが必要である.
7	0.689	0.877	機械翻訳システムの出力を編集することが必要である.
8	0.617	0.531	コンピュータによる翻訳出力を人間が読んで, 編集することが必要である.
9	0.407	0.245	機械翻訳システムの問題点は, そのままでは使えず, 後編集が必要なことである.
10	0.373	0.109	機械翻訳したあとに人間の編集作業が必要である.
11	0.332	0.104	機械の出力する翻訳結果が不十分であることが多いので, 人間が内容をみて訂正することが必要である.
12	0.307	0.134	自動翻訳の結果を手で編集することは必要である.
13	0.292	0.076	機械翻訳の結果は人間がチェックして訂正する必要がある.
14	0.274	0.102	システム上, 必要ならば, 人間が翻訳したものを機械で編集する.
15	0.272	0.191	人間は機械翻訳システムを必要としている.
16	0.238	0.056	翻訳作業は, 人間の英知が必要な作業で, 機械的な編集作業ではない.
17	0.235	0.081	機械翻訳は人間の能力を補助するシステムである.
18	0.212	0.044	マニュアルを翻訳するときに, 編集のための機械が必要となる.
19	0.210	0.074	機械翻訳の翻訳結果の品質は人間には及ばない.
20	0.204	0.074	マシントランスレーションの出力を手で訂正することは必要である.
21	0.174	0.051	人間の翻訳作業にシステム上必要な機械を購入する.
22	0.153	0.068	機械翻訳の分野では例による翻訳が注目されている.
23	0.108	0.039	これはシステム上は必要のない, かつ, 意味のない翻訳である.

マイニングを「デー」「タマ」「イニング」に分割してしまい, 情報検索の性能がでない. 辞書に単語を登録するだけでBD法の性能は向上するが, 常に新しい情報を対象としなければならない場合, そのコストは大きい. BN法, DP法とも辞書のメンテナンスが不要であるため, 両者はこの点でBD法よりも優れている.

表4に示すように, 提案する文字列重みの編集類似度は, 情報検索などにおいて形態素解析を使用したものと同等以上の性能を上げており, 形態素解析を使用しない方法としては注目に値する評価結果となっていると考える.

別の立場からすると, 文字列重みの類似尺度は, ある意味での「語」を抽出しているという解釈も可能である. 文字列重み編集類似度は, そのスコアの合算の過程で, 類似判定に効果のある部分文字列を選び出す処理を行なっ

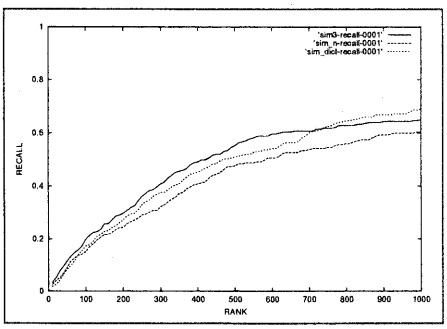

図1 質問1,「自律移動ロボット」: 簡単な質問

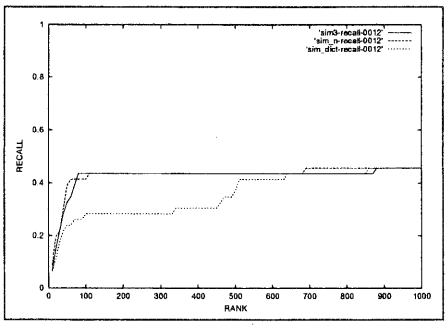

図2 質問12,「データマイニング」: 用語を正しく単語に分割できないため, BD 法には難しい質問.

ている. つまり, 定義4.1に示される定義式で MAX によって選び出された文字列は, 類似判定に効果がある文字列として選び出されている. ある意味では, 類似判定に効果のある文字列として, あるカテゴリに属する単語を抽出していると解釈することも可能である. これは, 類似判定ごとに「語」の定義を変更することで効果を上げている報告[10] と同様に, 固定的な単語分割を行なわない方法の一つと解釈することもできる.

5.5 BN 法と DP 法の比較

この節では, BN 法と DP 法を比較し, 表現上の揺れに対する振舞いの違いについて考察する. BN 法は一致した文字列に対するスコアを加算を行なっているという意味で代表的な方法である. それ以外の条件は BN 法と DP 法は揃っている. 表5では, 順位1の文字列に関して, 自分を含む18個の文字列と, それぞれの類似度を BD 法, BN 法により算出したものである. 比較のために, 類似度は同一文字列を与えた場合との相対値としてある. 順位は DP 法の結果に従って示した.

まず, 順位2は1文字を挿入したものであるが, BN 法のスコアが大きく減少しているのがわかる. DP 法は編集距離を拡張したものであるので, 1文字の差について穏やかに反応しており, 表記のゆれに関して良好な性質を持っていることがわかる.

結果が食い違っているのは順位,10, 11, 13, 14, 15 である. 特に, 13, 15 を比較すると, DP 法が重要なキーワードの構造にしたがって, 分別する能力があることが示唆される.

6. 同じデータセットを用いた情報検索システムとの比較

情報検索の問題に限れば, DP 法より高性能の他の方法がある. 実際, 情報検索においては, 確率モデルを使ったバイグラムシステム[12] やフレーズ検出を工夫したシステム[13] のほうが優れた値を求めている.

本論文で提案する文字列重み編集類似度を用いたシステムの性能は, 教科書に記述されているようなシステムと高性能なシステムの中間であった. 本論文に示す実験では, 文字列重み編集類似度は単純に IDF を利用しているが, Score 関数の改良を行なうことによって, より性能を向上させることができると考えている. しかしながら, 文字列重み編集類似度は, 辞書や形態素解析システムを使用せずに動作しているため, 新しい単語や用法が生まれるような状況でも性能の低下が起こる要因がないのは, フレーズ検出を行なっているシステムにはない特徴である. また, 予備の試験データからのチューニングを行なわない状態で性能がでており, これは正解判定からパラメータをチューニングする必要がある確率モデルにはない特徴である.

6.1 単語の順序について

文字列重み編集類似度の定義は, 順序を保存するという制限のもとでの最大のスコアの合算値となっている. キーワードの検索においても順序を保存することで検索精度が向上するという報告[8] があるが, 文字列重み類似度は, キーワードに限定していないところが異なる.

順序情報を利用するという意味では, 形態素解析を行ない, 内容を示す名詞や動詞の列にとして扱い, その順序の情報を利用して検索の結果を向上させている研究[11] があるが, 形態素解析が必要であるのは, 提案している手法と異なっている. また, 修飾語の欠損と追加がある場合にも, 提案する手法では, 類似性を検出できるところが異なる.

6.2 フレーズの検出

文字列重み編集類似度は, そのスコアの合算の過程で, 類似判定に効果のある部分文字列を選び出す処理を行なっている. つまり, 定義4.1に示される定義式で MAX によって選び出された文字列は, 類似判定に効果がある文字列として選び出されている. この一連の文字列は, 検索に効果があるひとかたまりと解釈できる. 言い替えれば, 文字列重みの類似尺度によって, 検索のための「分離している複合語」を抽出しているという解釈ができる. これは, 類似判定ごとに「語」の定義を変更することで効果を上げている情報検索システム[10] と同様に, 情報検索の尺度を用いる分割方法の一つと解釈することもできる.

検索に効果がある文字列の集合を選ぶよく行なわれる方法は，共起関係を利用する方法である[4]．文字列重み編集類似度で選び出される文字列の集合は，共起によるものとは異なる文字列となる．端的には，文字列重みによるものは，IDF が高ければ，まったく統計的に独立に出現し，全く共起関係にない文字列でも組として検出される．実際に，文字列重み編集類似度で求まる一群の文字列の性質を分析することは行なう価値のある今後の課題である．

7. 今後の課題

文字列重み編集類似度は単純に IDF を利用しているが，Score 関数の改良を行なうことによって，より性能を向上させることができると考えている．重みの決定は，多くの要因があり，方式によりもっとよい重みは異なると考えられ，文字列重み編集類似度の性能を向上させる重みを発見することは，今後，行なうべき課題である．

8. ま と め

本論文では，情報検索の性能を持ち，かつ表記の揺れにも対応することができるダイナミックプログラミングを用いた類似度の計算法を提案し，その文字列重み編集類似度の情報検索性能とその類似度が持つ性能を議論した．実験を行ない，単語に基づく手法と ngram に基づく手法と比較した結果，提案した手法が効果的であり，かつ提案した手法が表記の揺れに寛容であることを報告した．

参 考 文 献

1) Gerard Salton and Christopher Buckley, Term-Weighting Approaches in Automatic Text Retrieval, *Information Proceeding and Management*, 24, pp.513–523, 1988.

2) Robert R. Korfhage, Information Storage and Retrieval, WILEY COMPUTER PUBLISHING, John Wiley & Sons, Inc.,Printed in USA, pp. 291-303, 1997.

3) Hideo Fujii and W. Bruce Croft, A Comparison of Indexing Techniques for Japanese Text Retrieval, *In proceeding of SIGIR'93*, Pittsburgh PA, USA, pp.237–246, 1993.

4) 高木 徹, 木谷 強：単語出現共起関係を用いた文書重要度付与の検討情報処理学会, 情報学基礎研究会報告, FI41-8, 1996

5) Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Osamu Imaichi, and Tomoaki Imamura, Japanese Morphological analysis System ChaSen Manual, *NAIST Technical Report, NAIST-IS-TR97007*, February 1997, http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html

6) Kando, N. et al., NTCIR:NACSIS Test Collection Project, *20th Annual Colloquium of BCSIRSG*, Autrans, France, March 25-27, 1997.

7) Kageura, K. et al., NACSIS Corpus Project for IR and Terminological Research, *Natural Language Proceeding Pacific Rim Symposium'97*, Phuket, Thailand, pp.493-496, December 2-5, 1997.

8) 田中英輝：長い日本語表現の高速類似検索手法情報処理学会, 自然言語処理研究会報告, NL121-10, 1997

9) 山田剛一, 森 辰則, 中川 裕志：複合語マッチングと共起情報を併用する情報検索情報処理学会論文誌, Vol. 39, No.8, pp. 2431-2439

10) 小澤智裕, 山本幹雄, 山本英子, 梅村恭司：情報検索の類似尺度を用いた検索要求文の単語分割言語処理学会大会, A5-2, 1999

11) ：大竹清敬, 増山 繁, 山本 和英：名詞の連接情報を用いた関連文書検索手法情報処理学会論文誌, Vol.40, No.5, pp. 2460-2467, 1999

12) Aitao Che, Fredric C. Gey, Kazuaki Kishida, Hailing Jiang and Qun Liang, Comparing multiple methods for Japanese and Japanese-English text retrieval, Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, NTCIR Workshop 1, pp. 49-58, September 1, 1999, Tokyo Japan.

13) Sumio Fujita, Notes on Phrasal Indexing JSCB Evaluation Experiments at NTCIR AD HOC, Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, NTCIR Workshop 1, pp. 101- 108, September 1, 199, Tokyo Japan.

14) Eiko Yamamoto, Kyoji Umemura, Tomohiro Ozawa, Mikio Yamamoto, Kenneth W. Church, Character Based Information Retrieval using Generalized String Similarity, Proceedings of the IREX Workshop, pp. 95-100, 1999.