自然言語処理　138－9
(2000. 7. 19)

ダイナミックプログラミングでの検索方法

山本 英子† 梅村 恭司† 山本 幹雄‡ Kenneth W.Church*

† 豊橋技術科学大学 情報工学系　‡ 筑波大学 電子・情報工学系
*AT & T Labs – Research
†441-8580 豊橋市天伯町雲雀ヶ丘 1-1
TEL: (+81)532-47-0111(ex.5430)
†eiko@avenue.tutics.tut.ac.jp, myama@is.tsukuba.ac.jp,
umemura@tutics.tut.ac.jp, kwc@research.att.com

あらまし　本論文では,技術文書でよく用いられる「機械翻訳システム」のような技術用語を念頭に置き,ダイナミックプログラミングに基づく新しい類似度を提案する．また,提案する方法を二つの基本システムと比較する．これらの二つの基本システムはどちらとも標準的な頻度と情報量の内積の類似度を利用するが,対象となる語彙が異なる．日本語技術文書のコレクションを用いて実験を行ない比較した結果,提案するダイナミックプログラミングに基づく手法が最も効果的であったことを報告する．また,質問が一つの長い技術用語を含み,その用語を構成する単語がどれも特によいキーワードでないとき,提案する手法が比較的強いことを述べる．

キーワード　用語学,編集距離,情報検索,技術用語,類似尺度

A Retrieval Method Based on Dynamic Programming

Eiko Yamamoto† Mikio Yamamoto‡ Kyoji Umemura† Kenneth W.Church*

† Dept. of Information and Computer Sciences, Toyohashi University of Technology
‡ Institution of Computer Sciences and Electronics, University of Tsukuba
*AT & T Labs – Research
†Tempaku, Toyohashi, Aichi, 441-8580, Japan,
TEL: (+81)532-47-0111(ex.5430)
†eiko@avenue.tutics.tut.ac.jp, myama@is.tsukuba.ac.jp,
umemura@tutics.tut.ac.jp, kwc@research.att.com

Abstract　We introduce a new similarity measure based on dynamic programming, intended for technical terms such as *machine translation system*, which are quite common in technical writing. We compare our proposal with two baseline systems: baseline-dict and baseline-ngram. Both baselines use standard tf/IDF cosine similarity, but on different vocabularies. For our testcollection of technical Japanese abstracts, we find that the proposed dynamic programming solution works better than others. The dynamic programming method is relatively strong when the query contains a single long technical term, and none of the words in the term are particularly good keywords.

key words　Terminology, Edit Distance, Information Retrieval, Technical Term, Similarity Measure

1. はじめに

情報検索において最も効果的な単位は何であろうか．現在の研究では，英文については単語であるという傾向にある．Damashek[7] のような例外もあるが，英文では単語の部分文字列のようなより小さな単位はほとんど用いられない．しかし，文字に基づくシステムはアジアの言語においてよく用いられる．[10),12)~14)]．一方，句や単語の列のようなより大きな単位も注目されている．Fagan の初期の研究成果[1] は見込みのあるものだった．しかしながら，Stzalkowski[8] などの多くの研究者たちによる過去十年にも渡る努力も空しく，句に関する研究は未だ効果を実証するに至っていない．また，Mitra *et al.*[18] は特に高いランクで，句はあまり役に立たないということを示唆している．

もう一方，Mitra *et al.*[18] は，句が中間のランクでは最も役立ちそうだということを発見した．中間のランクとは明らかに正解(高いランク)でも，明らかに不正解(低いランク)でもないドキュメントが位置するところである．本論文でも，実験結果から，特に質問とドキュメントの両方に技術用語，“Computer Sciences”，“Information Retrieval”，“machine translation” のような句が多く使われるとき，句が中間のランクで最も役立つことを発見した．

「技術用語」という用語を定義することは難しい．その言語を話す人は誰でも知っているというような「一般語彙」とは違い，「技術用語」はある分野の専門家ごとに使われる語彙である．一般語彙と技術用語を区別することは，翻訳者にとって難しいことである．多くの場合，翻訳者は著者と読者のそれぞれの専門分野に精通していないからである．一般的に，用語の表現が統一されていると，内容を把握しやすい．このことから，翻訳する内容に合致させるために，各用語の正しい訳は一つだけに統一される．

本論文では，一般語彙と技術用語を区別することも情報検索において重要視するべきであると考えた．技術用語が利用できると，その用語は特に関連性を捕らえるための強い手がかりとなりうる．

また，多くの技術用語は一つの単語より長く，句の形をしている．このような複数の単語からなる用語が技術文書では多く用いられる．本論文の実験は日本語の大きな技術的なアブストラクトのコーパス[16),17)] に基づいている．以前の句に関する研究もさまざまな異なったジャンルを扱ってきたが，技術的なジャンルを扱った研究は少ない．

一般語彙と違って，技術用語は非常に表現が固定されている．それは「用語」と用語が表す「概念」の一対一の関係をできるだけ保つスタイルは技術作文でのよいスタイルと考えられているからである．同義語や多義語があることは混乱のもとである．このことについて，Sager[3] は *British Standards Guide* を参照しながら，技術用語のような専門用語を次のように定義させるべきであると提案している．

> 専門用語：正確に定義される明瞭な用語が本質的である．一つの基準を持つ専門用語は，同じオブジェクトや概念を常に同じ用語で記述または表現し，類義語で記述または表現されない，無矛盾なものであるべきである．...

しかし，このような技術用語でも非常に限られた変形が可能である．例えば，一つのドキュメント内で一度現れた長い技術用語をそれ以降省略形を用いて表すことがある．Sager[3] は，英文でもときどき長い複雑な名詞列をその列にある単語を省略した形で用いることがあることを示している．

図1に示されるように，意味のある名詞列の変形はとても限定されたものしかない．そこで，本論文では，同じ用語を表す二つの名詞列は元となる名詞列の変形例ならば，文字でマッチングをとることができるのではないかと考えた．

本論文では，ある用語にいくつかの文字を挿入または削除することによってできるこの種の変形に強いダイナミックプログラミングを用いる情報検索方法 (DP 法) を提案する．この種の変形は文字の挿入や削除操作だけで作られるので，めったに文字の順序が入れ替わることはない．このことから，本論文では，提案するダイナミックプログラミングを用いて変形をモデル化する方法のほうが，完全に順序情報を無視し，文字列を単語または文字の袋として考える方法より効果的に情報を検索するのではないかと考えた．

例えば，「機械翻訳システム」と「機械翻訳の実験システム」はたいてい両方とも同じ物を表現するために使われる．これは長い句にある単語を挿入または削除することで同じ物を表現する場合があることを示している．特に句がかなり長くなる場合，文字の挿入または削除が見られる．例えば，「情報検索システム」の変形である「地理情報の検索するシステム」は漢字文字「地理」を挿入されたことによって，名詞句が長くなるため，平仮名文字「の」と「する」も挿入されている．

2. ベースラインシステム

本論文で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作成した．

dynamo trap clamp bolt → dynamo clamp bolt → clamp bolt
gearbox end cover gasket → end cover gasket → gasket
exhaust valve lifter cable → exhaust lifter cable → cable

図1 マニュアルからの変形例
Fig. 1 Examples of Variations from a Manual

単語に基づくシステムを実行するために，既存の日本語形態素解析プログラム「茶筌」[19]を用いて質問とドキュメントを単語に分割し，Salton[2]によって提案される余弦尺度 $tf \cdot IDF$ を使って単語の重付けを行なった．本研究では，「茶筌」が割り当てた品詞から名詞，動詞，未定義語を用語として用い，他の品詞の単語はストップワードと考えた．次にこのシステムで用いた類似度の関数を示す．

定義 2.1 名詞，動詞，未定義語のどれかである t を質問とドキュメントの両方に現われる単語としたとき，$tf(t)$ はそのドキュメントの単語 t の出現頻度 (term frequency) を，$df(t)$ は単語 t が出現するドキュメントの数 (document frequency) を表す．また，N はドキュメントの総数であるとする．

$$SIM_{dict} = \sum_{t} tf(t) \cdot Score(t)$$

但し，$Score(t) = -\log_2(df(t)/N)$

もう一方のシステムは文字に基づくシステムである．多くの文字に基づくシステムは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．本研究では，すべての部分文字列について，余弦尺度 $tf \cdot IDF$ を使って重付けを行なった．次にこのシステムで用いた類似度の関数を示す．

定義 2.2 α, β, ξ, η を文字列，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，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，$df(\xi)$ は部分文字列 ξ が出現するドキュメント数とある．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．

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

但し，$Comp(\xi, \eta)$ 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

但し，$Score(\xi) = -\log_2(df(\xi)/N)$

本研究では，日本語の文書を対象としている．日本語には，漢字，片仮名，平仮名の三種類の文字がある．漢字は数千文字，片仮名，平仮名はそれぞれ 50 文字ある．主に，漢字は単語を，片仮名は外来語を，平仮名は日本語の機能単語を表すために用いられることが多い．ここでいう機能単語とは単語と単語の間の「の」，「と」などのことである．また，片仮名で表される外来語は漢字で表される単語より長い傾向がある．例えば，「機械」や「翻訳」は二文字であるのに対し，「システム」は四文字である．このように，Fujii and Croft[4]でも示されているが，短い ngram は漢字には効果的であるが，片仮名には効果的でないと推定できる．したがって，中国語はすべて漢字で表されるので，短い ngram がより効果的であろう．実際に，中国語のための短い ngram を用いたシステムが議論され[12),13)]，韓国語に対しても議論されている[10)]．しかし，これらの報告があるからと言って，単に短い部分文字列だけを扱う必要はない．suffix array や PAT-tree[5),6),20),22)] などの複雑なデータ構造を使えば，非常に長い ngram を扱うことができる．

本研究では，実験対象とした日本語文書において，長い ngram と短い ngram(bigram) のどちらが性能が高いを比較した．その結果，表 2の最後の行に示されるように，この実験では長い ngram は効果があることを示すデータが得られた．

図 2にテストコレクションにある 24 番の質問を示す．この質問は「機械翻訳システム」を含んでいて，我々の検索システムではこの用語を利用して，効果的に検索することができるが，ベースラインシステムではどちらとも効果的に検索することができない質問の例である．

単語に基づくシステム (BD 法) は「機械翻訳システム」を「機械」，「翻訳」，「システム」の三つの日本語単語に区切る．しかし，三つの単語がどれも役立つキーワードではないので，BD 法では効果的に検索することができない．実際に，「システム」は特に役に立たないキーワードである．この単語は 33 万個のドキュメントの約半分に現われるので，その IDF 重みの寄与は無視できる．しかし，質問の最後の文に解説されるように，「システム」を考慮していないドキュメントは不適切とするので，「システム」はその質問には必要不可欠な単語である．

ngram に基づくシステム (BN 法) は「機械翻訳システム」という質問の中心となる用語全体を含むドキュメントに高いスコアを与え，この用語の部分しか含まないドキュメントには低いスコアを与えるので，この質問について，DP 法の次に効果的であった．BN 法はこの質問については BD 法より効果的であったが，先に示したような用語の変形を捕らえることができない．実際に，実験で用いた多くの質問について，BN 法は BD 法よりもかなり悪い結果であった．

スコア関数はより複雑なものを利用することができるが，本実験では単純な IDF に基づく関数を利用した．より複雑なスコア関数を用いると，手法の比較が不明瞭になる．本論文では，完全なスコアを得ることよりも，考慮する条件を増やすこと，つまり手法によって検索性能が向上するかどうかに興味があるので，単純なスコア関数を用いた．用語の問題は現実で，スコア関数の選択と独立に考えることができる．

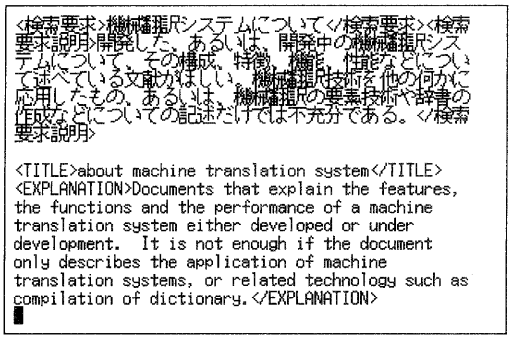

図2 質問24とその英訳
Fig. 2 Query #24 and Its Translation

3. 編集距離と提案するシステム

例えば,「機械翻訳システム」と「機械翻訳実験システム」のような一方の文字列がもう一方の文字列の数回の挿入と削除操作をすることによって作ることのできる変形である場合, 編集距離はこの変形動作をとらえるための自然な方法である. 編集距離については文献[15]の付録に記述されている. 編集距離は一方の文字列をもう一方の文字列に写像するために必要な挿入と削除操作などの最小編集操作数であり, スペル修正で広く使われている. 本研究では, この編集距離の情報検索への利用を試みた.

通常, 編集距離は二つの文字列がどれだけ違うかということを計算するものであるので, 文字列にある文字の違いを数え上げるものである. しかし, 本研究では, 文字列の一致した部分を見つけ, その一致した文字列についての重みを加算することにより情報検索を行ないたい. そこで, 通常の編集距離の場合, 文字列に違いを見つけた時点で1を加算し文字列の差を数え上げるが, これを違いが見つけた時点で, その直前まで一致していたとして, その一致した部分文字列のIDF重みを加算するように改良した. IDF重みの計算は関数 $Score$ が実行する. また, 編集距離を計算するために用いられるダイナミックプログラミングのアルゴリズムを一般化する必要がある. 通常, ダイナミックプログラミングの手続きはある時に一文字だけを考慮するものなので, スコア関数が長い部分文字列に高いスコアを与えることができるにもかかわらず, このままでは文字列を扱うことができない. そこで, 次の一文字だけでなく, その文字より前にある最も長い一致文字列にあるすべての文字を考慮できるように改良した. これは, ダイナミックプログラミングのアルゴリズムの一般化となっている. 本研究では, この提案する類似度をSWS (String Weight dynamic programming Similarity) と名付けた. この類似度はUkkonen's Enhanced Dynamic Programming ASM (Approximate String Matching)[9]に似ているが, 文字ではなく, 文字列の重みに基づいている.

定義 3.1 α, β, ξ, η を文字列とする. α_{km} を k 番目の文字から $k+m-1$ 番目の文字までの α の部分文字列, α_{n*} を n 番目の文字から最後の文字までの α の部分文字列とする. β_{km} を k 番目の文字から $k+m-1$ 番目の文字までの β の部分文字列, β_{n*} を n 番目の文字から最後の文字までの β の部分文字列とする. また, $Score$ は文字列から実数値を求める関数とする.

$$SIM_{DP} = max_{i,j}(Comp(\alpha_{1i}, \beta_{1j} + SIM_{DP}(\alpha_{i+1*}, \beta_{j+1*})$$

但し, $Comp(\xi, \eta)$ 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

但し, $Score(\xi) = -\log_2(df(\xi)/N)$

ここでは, 編集距離を計算する方法である文字ごとの重みを計算するダイナミックプログラミング法は長い一致文字列が指数的に高いスコアを得ることができないと仮定している. 言い換えれば, スコア関数がすべての部分文字列について, 不等式 $Score(\delta\gamma) \leq Score(\delta) + Score(\gamma)$ が成り立つ場合, 最良のパスは単一文字の列から構成されるということになるので, 長い句を考慮する必要がなくなる. しかし, 我々が意図するスコア関数は長い句(技術用語)に高いスコアを割り当てることがあるので, この利点を得られるようにダイナミックプログラミング法を拡張した.

表1 11点平均精度
Table 1 11 point average precision

システム	DP	BN	BD
11pt	0.281	0.164	0.154

表2 質問ごとの勝ち負け
Table 2 better or worse for each query

	better	worse
DP vs BD	23	7
DP vs BN	29	1
BD vs BN	15	15

4. 実験結果

本論文の実験では, 33万個の日本語の技術的なアブストラクトと30個の質問の正解判定を持つNacsis[16),17)]

表3 質問の分析
Table 3 analysis of some queries

質問番号	BD	BN	DP	用語	注
1	○	○	○	自律移動ロボット (124)	典型的な例
12	×	○	○	データマイニング (74)	変形の少ない例
24	×	×	○	機械翻訳システム (244)	語順が問題となる例
29	○	×	×	位置計測 (176)	キーワードにならない例

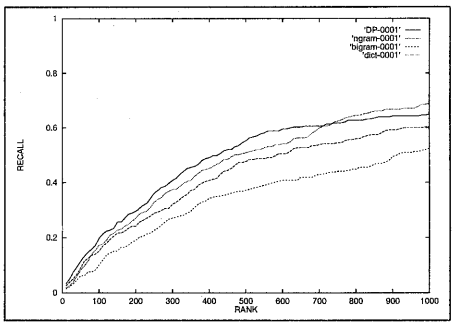

図3 質問1,「自律移動ロボット」: 簡単な質問
Fig. 3 Query #1, *autonomous mobile robot*: easy for all three systems.

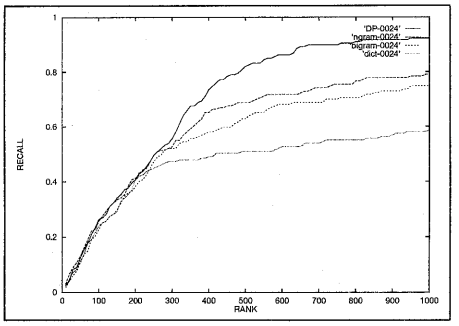

図5 質問24,「機械翻訳システム」: DP法が得意とする質問
Fig. 5 Query #24, *machine translation system*: ideal for dynamic programming

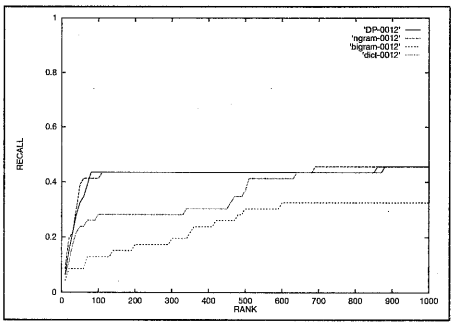

図4 質問12,「データマイニング」: BD法には難しい質問
Fig. 4 Query #12, *data mining*: hard for baseline-dict

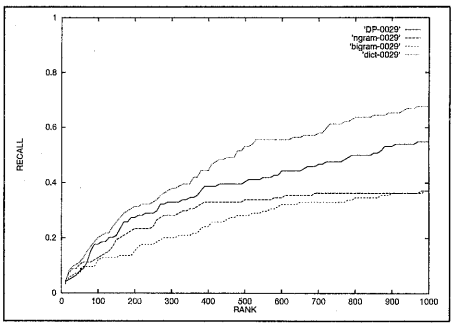

図6 質問29,「位置＋計測」: BD法が得意とする質問
Fig. 6 Query #29, *position + measurement*: ideal for baseline-dict

を利用した. この節では, 我々の提案する方法 (DP), BD 法 (BD), BN 法 (BN), の三つのシステムの結果を報告する. 多くの質問は一つの技術用語を含み, 残りの質問は三つのシステムがいづれもうまく扱うことができないものであった. 用語に関するものである.

表1にすべての質問について 11 点平均精度を示す. この表から, この実験において, DP 法は BD 法と BN 法より良い結果を得ることができたことがわかる. 表2は, 各質問ごとの 11 点平均精度を比較した結果である. この結果からも, DP 法は BD 法と BN 法より効果的であったことがわかる. また, 表2にある 23 対 7 という比は 2.6×10^{-3} のレベルで統計的に DP 法が BD 法より効果的であったことを意味している. 同様に, 29 対 1 という比は 2.9×10^{-8} のレベルで統計的に DP 法が BN 法より効果的であったことを意味している.

表3は, 30 個の質問のうち四つの質問について, 詳しく分析した結果である. この表では, "○" は性能が高いことを表し, "×" は性能が低いことを表す. また, かっこの中の数字は質問の中心となる用語がそのまま現われるドキュメントの数を表している. 質問1は, 質問の中心となる用語そのものとその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を持つ例で, このような質問はすべてのシステムにとって簡単な質問であった. 質問12は, 質問の中心となる用語そのものが良いキーワードとなり, その用語を構成する単語ごとはよいキーワードではなく, しかもその用語の変形があまり存在しない例である. このような質問では, DP 法と BN 法は強いが, 単語に基づいている BD 法には難しい質問であった. 質問24は, 質問の中心となる用語そのものが良いキーワードとなり, その用語を構成する単語ごとはよいキーワードではないが, その用語の変形が数多く存在するので, 構成単語の語順が重要となる例である. このような質問では, DP 法の

強さを見ることができる．質問29は，質問全体が一般語彙で表現され，技術用語を使われていない例である．表に示される「位置計測」は技術用語のようであるが，コレクション中でこの質問にしか現れない言葉で，この質問中で作られた複合名詞のようである．このような質問では，BD法の強さを見ることができる．

これらの四つの質問の再現率を図3-6に示す．これらの図には，bigramに基づく手法の結果も示す．これらの図から，性能に差があるとき，その差は中間のランクで最も大きくなることがわかる．つまり，情報検索性能は，中間のランクに位置付ける判断に迷うドキュメントを検出できるかどうかに左右される．また，Mitra *et al.*[18]も言うように，上位にある少数のドキュメントについては句を扱っても違いが現われないことがわかった．言い替えれば，明らかにその質問と関連のあるドキュメントはどのような手法でも検出し上位に位置付けることができるということである．

5. 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[25]やフレーズ検出を工夫したシステム[26]のほうが優れた値を求めている．これらは、情報検索の訓練課題で計数を調整したり，対象となる言語に依存した処理を行なっている．

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．BN法，BD法が文字列重みを調整することで性能が向上することと同様に，DP法でも、文字列重みを調整する余地がある．また，重みに言語に依存した情報を利用する余地もある．

ただし，IDFという言語に依存しない量だけを利用するという条件で文字列重み編集類似度を設計しており，その条件に従った実験と情報検索コンテストの結果は単純には比較できない．

5.1 単語の順序について

文字列重み類似度の定義は，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[21]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[24]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．

5.2 フレーズの検出

文字列重み類似尺度は，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3.1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[23]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[11]．文字列重み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，文字列重み類似尺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．

6. おわりに

本論文では，文字列重みを使用するダイナミックプログラミング法を提案した．そして，質問が「機械翻訳システム」というような質問の中心となる一つの技術用語を持ち，用語を構成する単語が低いIDF重みを持つ単語である場合，その方法が特に効果的であることを示した．今後の課題として，できるので，この新しいダイナミックプログラミング法とより伝統的な方法を組み合わせ，両者の利点を持つ混合システムを実現することを考えている．また，本論文では，Score関数として単純なIDF重みを用いたが，より性能を向上させる重みを発見しなければならないと考えている．

謝辞 本研究は住友電工との共同研究の成果を利用させて頂きました．深く感謝いたします．

参考文献

1) J. L. Fagan, Experiments in Automatic Phrase Indexing For Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods,it PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1987. Available as CUCS Technical Report TH87-868.

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

3) Juan C. Sager, A Practical Course in Terminology Processing, John Benjamins Publishing Company, Amsterdam/Philadelphia, 1990.

4) 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.

5) Udi Manber and E. Myers, Suffix array: A new method for on-line string searches, *SIAM Jour-*

nal on Computing, 22:5, pp. 935 - 948, 1993. http://glimpse.cs.arizona.edu/udi.html.

6) M. Nagao and S. Mori, A new method of n-gram statistics for large number of n and automatic extraction of words and phrases from large text data of Japanese, *Coling-94*, pp.611-615, 1994.

7) Damashek, Marc. Gauging Similarity with n-grams: Language Independent Categorization of Text," *Science*, Vol 267, 10 Feb 1995, pp 843-848.

8) T. Strzalkowski, L. Guthrie, J. Karlgren, J. Leistensnider, F. Lin, J. Perez-Carballo, T. Straszheim, J. Wang, J. Wilding., Natural Language Information Retrieval: Trec-5 Report, *in E. M. Voorhees and D. K. Harman (eds.), The Fifth Text REtrival Conference (TREC-5)*, pp.291–314, 1996. http://trec.nist.gov/

9) H. Berghel and D. Roach, An Extension of Ukkonen's Enhanced Dynamic Programming ASM Algorithm, *ACM Transactions on Information Systems*, Vol.14, No.1, pp.94–106, January 1996.

10) Joon Ho Lee and Jeong Soo Ahn, Using n–Grams for Korean Text Retrieval, *In proceeding of SIGIR'96*, Zurich, Switzerland, pp.216–224, 1996.

11) 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.

12) K.L.Kwok, Comparing Representations in Chinese Information Retrieval, *In proceeding of SIGIR'97*, Philadelphia PA, USA, pp.34–41, 1997.

13) Aitao Chen, Jianzhang He, Liangjie Xu, Fredric C. Gey, and Jason Meggs, Chinese Text retrieval Without Using a Dictionary, *In proceeding of SIGIR'97*, Philadelphia PA, USA, pp.42–49, 1997.

14) Yasushi Ogawa and Toru Matsuda, Overlapping statistical word indexing: A new indexing method for Japanese text, *In proceeding of SIGIR'97*, Philadelphia PA, USA, pp.226–234, 1997.

15) Robert R. Korfhage, Information Storage and Retrievel, WILEY COMPUTER PUBLISHING, Johon Wiley & Sons, Inc., Printed in USA, pp.291–303, 1997.

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

17) 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.

18) M. Mitra, C. Buckley, A. Singhal, C. Cardie, An Analysis of Statistical and Syntactic Phrases, *RIAO-97*, pp. 200-214.

19) 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.

20) Lee-Feng Chien, PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval, in proceeding of SIGIR'97 Philadelphia PA, USA, pp.50–58, 1997.

21) 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.

22) Mikio Yamamoto and Kenneth W. Church, Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus, *In proceeding of 6th Workshop on Very Large Corpora*, Ed. Eugene Charniak, Montreal, pp.28–37, 1998.

23) 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.

24) ：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.

25) 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 Retrie val and Term Recognition, NTCIR Workshop 1, pp. 49-58, September 1, 1999, T okyo Japan.

26) Sumio Fujita, Notes on Phrasal Indexing JSCB Evaluation Experiments at NTCIR AD HOC, Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrie val and Term Recognition, NTCIR Workshop 1, pp. 101- 108, September 1, 1999 , Tokyo Japan.