未踏テキスト用シソーラスの自動構築システムの開発

本論文では, 辞書に記載されていない単語を含む, 整備されていないテキストを未踏テキストと考え, そのような未踏テキストを理解し検索するのに役立つシソーラスを, 辞書を用いずに自動的に構築する手法を提案する. 未踏テキストを理解するためには, テキストに含まれる未知語を理解する必要がある. 未知語を理解するのにもっとも役立つ関連語は同義語である. そこで, 本研究では, 未踏テキスト用シソーラスとして, 関連語リストを生成することを試みる. 特に, 辞書に記載されていないような未知語に焦点を当て, 辞書を用いずに関連語の対を抽出することを考える. 本研究では, 関連語を同じように使用される単語と定義し, シソーラスは関連語対として抽出する. これを実現するために, まず統計処理で候補となる単語を切り出すという処理を行い, 切り出された単語について, 同じように使用されているかを判定する. 本論文では, 辞書を用いない関連語リストの構築方法を提案し, この方法によって関連語と判定された単語対を分析した結果を報告する.

Automatic Thesaurus Building System for Brandnew Text Data

In this paper, we propose a new method that generates pair of related words without any dictinaries. When we need to understand new text, a list of related words may help to understand the text when the word is unknown word to us. In this study, we define the related wotds as the words whose usage is similar to each other. This definition makes it possible to generate the list without using any kind of dictionaries. Our system also extracts words without any dictionaries, and then judges whether two words are used in similar way. Finally, we report the precision of the obtained lists using various kinds of corpora.

1. はじめに

日々増え続ける新聞記事や WWW のテキスト情報には新しい概念を示す単語が日々生成されている. この生成される単語は新規語であるが故に, ほとんど辞書に記載されていない. 本研究では, このような辞書に記載されていない単語が含まれるテキストは人間によって整備されていない未踏テキストと呼ぶ. 未踏テキストは最新情報を記述した文書であることが多い. その最新情報を理解することやその情報に関連するテキストを検索するには, テキストに出現する新しい概念を表す単語を理解する必要がある. しかし, このような新しく生成された未知語をそのまま理解し検索に用いることは難しい. また既知語であっても, テキストが扱う分野によってはその単語の意味が他の分野で用いられる既知の意味と異なる場合や, 作成者によっては同じ意味を表す単語であるのに異なる表記が用いられる場合がある. これらの場合もその単語を理解し検索に用いることは難しい. このような単語を理解するために, その単語に関する情報として関連語が考えられる. たとえば, 未知語に関連する既知語を知ることができれば, その未知語を理解することができ, 未知語に関連する情報を検索することができる. また, 未知語に関連する他の未知語を特定することにも役立つ. そこで, 本研究では, 未踏テキストから未知語, 既知語を問わずその単語と関連語が対となった関連語リストを未踏テキスト用シソーラスとして, 自動的に構築することを試みる.

テキストからシソーラスを構築するためには, まずはじめに, テキストから対象となる単語の切出しが必要である. この単語の切出しは, 辞書を用いずに文字列の頻度情報のみで語分割を行い, テキスト中のキーワードを抽出できるシステム [4),5)] を利用すればできる. このシステムを利用することによって, 本研究では, テキストから未知語, 既知語を問わず関連語リストの対象となる単語を切り出すことができる.

次に, 切り出された単語集合において考えられ得る単語の対をすべて関連語であるかどうかを判定すればよいのだが, これは現実的ではない. そこで, 考え得る単語の対を対象とすべき対に絞り込む必要がある. また, 抽出するシソーラスに登録するべき関連語の定義も問題である. シソーラスには, 同義語, 類義語, 上位

語, 下位語などさまざまなものがある. 本研究では, テキスト集合中で同じように使用される単語をシソーラスに登録することにした. シソーラスでもっとも重要な単語は同じ意味で用いられる単語や似た意味で用いられる単語である. たとえば, 二つの単語の関係が同義関係である場合, 一方の単語を含む文に対して, その単語の部分を他方の単語に置き換えても, その文の意味は同じになる. 言い換えると, 同じように使用される二つの単語は同義語であり得る. また, 同義語以外の上位語, 下位語についても同じように使用される傾向があるのではないかと考えられる. そこで, 本研究では, 関連語をテキスト集合中で同じように使用される単語と定義実際に抽出された単語はどのようなものであるかを分析する.

2. 問題定義

従来, シソーラスの構築には, テキストから単語を切り出す工程から辞書が用いられている. たとえば, 日本語テキストから単語を切り出すために辞書による形態素解析システム「茶筌」がよく用いられる [8]. これは, ユーザ辞書を登録することもできる. しかし, 日々新しい単語が生成される今日では, 未知語が出てくるたびに登録するのは手間がかかる. また, 未知語に対応するために辞書が膨らんでいくため, 辞書を蓄えられる大容量の記憶媒体が必要である.

一方, シソーラス構築の際には計算機の性能が問題となる. この問題は, シソーラスの対象となる単語の数やシソーラスの要素となる関連語の定義によっては, 計算コストがかかるためである.

そこで, 本研究で提案するシステムは次の条件のもとでシソーラスを構築する.

定義 2.1 シソーラス構築の条件

- どの工程においても辞書を用いない.
- 汎用計算機で実現できる.

本研究では, この条件のもとでシソーラスを構築する. 本研究の目的は未踏テキストの理解に役立つシソーラスを構築することである. 未踏テキストを理解するためには, テキスト中の未知語を理解する必要がある. 未知語を理解することにもっとも役立つ関連語は同じ意味で用いられる単語や似た意味で用いられる単語である. このことから, 本研究で構築するシソーラスを関連語リストとした. そして, 関連を判定する方法として, 二つの単語が対象とするテキスト集合中で同じように使用されるかどうかを調べることにした. 具体的には, 二つの単語が対象とするテキスト集合中で前後に同じ文字列を持って出現するかどうかを調べることにした. たとえば, 「年賀状を印刷しなければならない」という文がある場合, 「印刷」を「プリント」に置き換えても同じ意味の文になる. この単語の置換えは「印刷」と「プリント」が同義語であるが故にできることである. このことから, 逆に, 二つの単語が前後に同じ文字列を持って出現するのであれば, 同義関係にあるケースを多く含む関連語ではないかと想定し, 関連語の判定を行うことにした.

本研究では, 関連語を前後に同じ文字列を持つ, テキスト中で同じように使用される単語と定義する. したがって, 本研究で得られる関連語はあくまで同じように使用される単語である. 関連語リスト $Relevants$ を以下のように定義する.

定義 2.2 関連語の定義

x, y は文字列, a, b は判定される単語, xay, xby はそれぞれ単語 a, b の前後に文字列 x, y を結合した文字列とする. $cf(z)$ はテキスト集合における文字列 z の総出現頻度とする. $score(a, b)$ は出現頻度情報に基づいて定義した a, b のスコア関数とする.

$$Relevants = \{(a, b) \mid score(a, b) > \alpha\} \qquad (1)$$

スコア関数 $score(a, b)$ には, 語の特徴度を表し, 語が特徴的に多く出現することの数量的な評価になっていると考えられる $cf \cdot IDF$[2] を採用した. これは, $IDF(z) = 0$ ならばすべてのテキストに出現し, $cf(z) = 0$ ならばテキスト集合に一度も出現しないことを表し, 意味のある単語であれば両方を考慮したものであるという考えに基づいている. この $cf \cdot IDF$ は, 現在の検索システムで広く用いられている指標であり, その有用性は経験的に実証されている. この $cf \cdot IDF$ に基づき, 本研究では, 二つのスコア関数を定義する. 次に, 本研究で定義したスコア関数を示す.

定義 2.3 スコア関数

$cf(z)$ はテキスト集合における文字列 z の総出現頻度, $df(z)$ は文字列 z が出現するテキスト数, N はテキストの総数とし, $IDF(z)$ を $-\log(df(z)/N)$ としたとき, $score(z)$ を $cf(z) \cdot IDF(z)/\log(N)$ とする.

(その 1) $cf(xab) > 1 \wedge cf(xby) > 1$ のとき, $cf \cdot IDF$ の積を加算.

$score(a, b) = \sum_{x,y} score(xay) \cdot score(xby)$

(その 2) $cf(xab) > 1 \vee cf(xby) > 1$ のとき, $cf \cdot IDF$ の高い方を加算.

$score(a, b) = \sum_{x,y} MAX(score(xay), score(xby))$

スコア関数その 1 は一致する前後文字列も含むそれぞれの単語に関する文字列がどちらとも $cf(z) > 1$ であるならば, それぞれの文字列に関する $cf \cdot IDF$ の積を加算する関数である. これは, テキスト集合において

偶然同じように使われているというケースを考慮しない関数とした．これは，情報検索において，$cf(z) = 1$ である単語は稀であるため，検索に有用でないという経験的な考えから，そのような単語に関する関連語を抽出しないように定義した関数である．一方，スコア関数その 2 はどちらか一方の文字列が $cf(z) > 1$ であるならば，それぞれの文字列に関する $cf \cdot IDF$ の高い方を加算する関数である．これは，一方の単語は稀な単語であるが，他方の単語がテキスト集合においてある程度の特徴度を持つ単語であるならば，二つの単語の関連は有用であり得るというケースを考慮した関数とした．この関数は，テキスト集合においてその 1 で切り捨ててしまう稀な状況にある情報は実際に有用でないのかということを調査するために考案した．

本研究では，定義式を用いて関連語を抽出する．しかし，関連語を判定する工程において，テキストから切り出された単語集合で考え得る単語の対をすべて対象とすると，計算量の問題が生じる．たとえば，本研究で単語を切り出すために利用するシステムは 125Mbytes のテキスト集合から約 10 万単語を切り出す．この場合，判定する対は 100 億ということになる．汎用計算機上で本研究のシステムは調べる前後文字列の長さを 4 文字とした場合，このテキスト集合において一対の判定に 0.001 秒程度かかる．これは，100 億対を判定するのに 120 日かかるということである．これは実用的な計算時間ではない．そこで，本研究では，テキストから切り出された単語集合で考え得る対を関連語となる候補の対に絞り込む．

通常，関連語の抽出に利用される出現分布は共起情報であるが，一つの文書において同じ概念を表す単語を二つ以上用いられることは少ない．これは，一つの文書は唯一の著者によって書かれるものであるので，単語は統一される傾向にあるためである．特に技術文書においては読者の理解を容易にするために，故意に単語が統一される傾向にある．一方，同義語や類義語ではなく，推移関係にある単語などは同じ文書に頻繁に現れる．そこで，本研究では，単語の対について，一方の単語と同じ文書に頻繁に現れる単語が，他方の単語とも同じ文書に頻繁に現れるならば，その単語対は推移関係にある同義語や類義語である可能性があるとして，本研究で考慮する関連語の候補とした．候補の絞込みは，各テキストから切り出された単語を空白で区切って一行に並べた単語集合を用いて行う．関連語の候補 $Candidates$ は次の式で定義される．ここで，x, y, z, a, b, c は単語，xyz は三つの単語が連なった単語列，$P(w)$ は単語列 w の出現確率，α は閾値とする．

定義 2.4　関連語の候補

$Tri(\alpha) = \{xyz | \frac{P(xy)}{P(x)P(y)} > \alpha \wedge \frac{P(yz)}{P(y)P(z)} > \alpha\}$ のとき，

$$Candidates = \{(a, b) \mid xaz \in Tri(\alpha) \wedge xbz \in Tri(\alpha) \wedge x \neq a \wedge x \neq b \wedge z \neq a \wedge z \neq b\} \quad (2)$$

この定義は，切り出された単語集合の要素である単語を切り出された順に並べた列において，二つの単語が前後それぞれ同じ単語を持つのであれば関連語の候補とすることを表す．

3. シソーラス構築手法

本研究では，以下に示す工程を経て，前節で定義した未踏テキスト用シソーラスとなる関連語リストを生成する．

(1) テキスト集合から単語を切り出し，対象とする単語集合を求める．
(2) 単語集合から単語の対を作成し，関連語の候補となる対に絞り込む．
(3) 候補が関連語の対であるかを判定し，関連語リストを生成する．

以下の節では，この三つの工程を順に説明する．

3.1 単語の切出し

第一の工程は関連語リストの対象となる単語の切出しである．本研究では，新規テキストに含まれる未知語を理解することに役立つ関連語の発見を目的とするため，テキストにある未知語，既知語を問わず単語を切り出さなければならない．しかし，日本語には語の境界がないため，日本語テキストは計算機にとって処理しにくいという問題がある．このため，関連語のリストの対象となる単語，特に未知語の切出しに失敗する場合が多い．そこで，本研究では既存の未踏テキスト中のキーワード抽出システムを利用する [4),5)]．このシステムは辞書を用いず，テキストの部分文字列から概念を示す単語と判定できる文字列を頻度情報のみで切り出し，その単語をキーワードとして抽出するシステムである．本研究では，このシステムを用いて抽出したキーワードを関連語リストの対象となる単語として扱う．図 1にこのシステムを用いてテキストから抽出されるキーワードの例を示す．このキーワード抽出システムに関する詳細は文献 [4),5)] に譲る．

3.2 候補の絞込み

切り出された単語集合から考え得る単語対をすべて関連関係にあるかを調査する場合，計算量が問題となる．そこで，第二の工程は第一の工程で抽出した単語の集合から考え得る単語対を関連語となる候補の対に

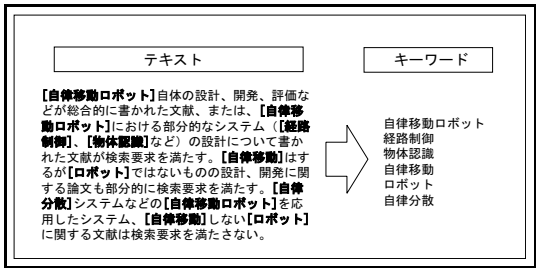

図 1　抽出されるキーワードの例

絞り込む．式 2.4で候補とする対を定義したが, その定義式をそのまま実現すると, 計算時間がかかる．そこで, 以下の式を用いて効率的に候補となる対を選び出す．本研究では, 以下の定義式を用いて, 絞込みを行う．

定義 3.1　候補の選択

$$Bi(\alpha) = \{xy \mid \frac{P(xy)}{P(x)P(y)} > \alpha\}, \qquad (3)$$
$$F(c) = \{x \mid xc \in Bi(\alpha)\},$$
$$B(c) = \{y \mid cy \in Bi(\alpha)\} \text{ のとき},$$
$$BF(a) = \{B(x) \mid x \in F(a)\},$$
$$FB(a) = \{F(y) \mid y \in B(a)\},$$
$$Candidates = \{(a,b) \mid b \in FB(a) \wedge BF(a)\}$$

例 3.2　ここに, 次の三つの文がある．このとき, 下線部分がキーワード抽出システムによって切り出された単語とする．

「昨夜, 年賀状 を 印刷 して 郵便ポスト に 投函 した. 」
「昨夜, 原稿 を 印刷 して 郵送 した. 」
「昨夜, 原稿 を 修正 して 郵送 した. 」
「昨夜, 宣伝チラシ を 100 部 印刷 した. 」
「昨夜, 雪 が降った. 」

ここで, 単語「年賀状」に関する候補を考える．

$Bi(\alpha)$ = {「昨夜」「年賀状」,「年賀状」「印刷」,
「印刷」「郵便ポスト」,「郵便ポスト」「投函」,
「昨夜」「原稿」,「原稿」「印刷」,「印刷」「郵送」,
「原稿」「修正」,「修正」「郵送」,「昨夜」「宣伝チラシ」,
「宣伝チラシ」「印刷」,「昨夜」「雪」} のとき,

F(「年賀状」) = {「昨夜」},
B(「年賀状」) = {「印刷」},
BF(「年賀状」) = {「年賀状」,「原稿」,「宣伝チラシ」,「雪」},
FB(「年賀状」) = {「年賀状」,「原稿」,「宣伝チラシ」},

このとき,「年賀状」に関する候補は次のものとなる．

$Candidates$ = {「年賀状」「原稿」,
「年賀状」「宣伝チラシ」}

この例では,「年賀状」に関する候補のほか,「修正」「印刷」という単語対も候補となる．

実験では, 関連語対の候補を作る単語は $cf > 3$ のものとし, 閾値 α を 2.0 とした．これらの数値は, 値を変化させた実験を行った結果から経験的に決定した値である．

3.3　関連語の判定

第三の工程はテキスト集合から候補の対が関連語の対であるかどうかを判定する．通常, 関連語を取り出す手法は単語の出現分布が類似しているかどうかを判定し関連語を取り出すが, 本研究では, 式 1に定義した判定式を用いて, 単語の前後に接続している文字列の関係を基に関連語の対であるかどうかを判定する方法を検討する．ここで, 問題となるのが調べる文字列の長さである．調べる文字列が短すぎると, 偶然に前後に接続する文字列が一致する単語の対が増え, 実際には関連のない対でも抽出される．また反対に, 長すぎると, 実際には関係のある対でも取りこぼされてしまうという問題が起こる．このように, 本研究において, 調べる文字列の長さは関連語を抽出するための重要なパラメータである．そこで, 実験では, 調べる文字列の長さを変化させることによって, 抽出される関連語対はどのように変化するかを観察する．また, スコア関数の違いによって, 抽出される関連語対の違いを観察する．それぞれのスコア関数の閾値には経験的に決定した値, その 1 は 5.0, その 2 は 3.0 を設定した．これは, NTCIR1 データから得られた結果を観察することによって決定した値である．

4.　実験の概要

4.1　対象とするテキスト集合

実験では, 日本語で書かれた NTCIR の学術文書データ [1] と毎日新聞記事データ [7], 中国語で書かれた新聞記事データ [6] をそれぞれテキスト集合として, 各テキスト集合用の関連語リストの構築を行った．本研究で提案するシソーラス構築システムは辞書を用いないため, 言語に依存しない．このことを検証するために, 日本語と中国語でそれぞれ書かれたテキスト集合を対象として実験を行う．次に実験の対象としたテキスト集合について説明する．

(a) NTCIR の学術文書データ
NTCIR の学術文書データは NTCIR1 と NTCIR2 があり, 様々な分野から学術文書の抄録を集め, 構築されたテストコレクションである．表 1 に実験対象とした日本語コレクションを示す．一つのデータには, 識別番号, タイトル, アブストラクト, 著者によって付与されたキーワードなどが含まれている．実験では, 一つの文書が持つタイトルとアブストラクトをタブで連結し, 一行にし

たものを一つのテキストとして使用する.

表 1 実験対象とした NTCIR の仕様

テキスト集合	件数 (Mbytes)	内容
NTCIR1	332,921(125)	NTCIR1 J コレクション (学会発表データベース)
NTCIR2G	116,177(98)	NTCIR2 J コレクション (学会発表データベース)
NTCIR2K_1	100,000(138)	NTCIR2 J コレクション (科研費補助金研究成果概要データベース) の一部
NTCIR2K_2	100,000(135)	NTCIR2 J コレクション (科研費補助金研究成果概要データベース) の一部
NTCIR2K_3	87,071(117)	NTCIR2 J コレクション (科研費補助金研究成果概要データベース) の一部

(b) 毎日新聞記事データ

毎日新聞記事データは 1991 年版から 1994 年版までを使い, 一年分をそれぞれテキスト集合として, 年版ごとの関連語リストを構築する. 表 2に実験対象とした新聞記事データを示す. 一つのデータには, 識別番号, 見出し, 本文などが含まれている. 実験では, 一つの記事に含まれる文章すべてをタブで連結し, 一行にしたものを一つのテキストとして使用する.

表 2 実験対象とした毎日新聞記事データの仕様

テキスト集合	件数 (Mbytes)	内容
MAI1991	91,200(85)	CD-毎日新聞 1991 版
MAI1992	101,468(85)	CD-毎日新聞 1992 版
MAI1993	91,774(85)	CD-毎日新聞 1993 版
MAI1994	101,057(115)	CD-毎日新聞 1994 版

(c) 中国語新聞記事データ

本論文では, 構築したシステムが言語独立なシステムであることを示すために中国語を対象とした実験を行う. 表 3に実験対象とした中国語新聞記事データを示す. 一つの記事データには, 識別番号, 見出し, 本文などが含まれている. 実験では, 一つの記事に含まれる文章すべてをタブで連結し, 一行にしたものを一つのテキストとして使用する.

表 3 実験対象とした中国語新聞記事データの仕様

テキスト集合	件数 (Mbytes)	内容
CIRB_1	65,673(85)	CIRB010 の一部

4.2 関連語リストの評価方法

本研究で対象とするテキスト集合は内容もさらに言語を問われない. これは, 辞書を全く用いずにテキスト集合にある情報だけを使って, 関連語リストを構築できるためである. このため, はじめの工程で切り出される単語には未知語も含まれる. したがって, 辞書を用いて, 得られた関連語リストの評価を完全に行うことができない. さらに, 単語が辞書にあってもテキスト集合特有の使われ方や意味を持つ単語の場合, 評価することは難しい. そこで, 本研究では, 実験で調べる前後文字列を 2 とした場合得られた単語対から 500 件をランダムに選び, 五人の人間が各単語対が有用であるかどうかを判定する. 判定者は単語対にある単語を知らない場合, どんな手法を用いてでもその単語について調べ, 自分なりの判定を下すものとする. そして, 各個人が以下に示す四段階で有用性を判定した結果を集め, 総合的な判定を付け, 関連語リストの適合率を出し, 評価とする. 判定は以下の四段階で行った.

(1) 同じように使用される単語対 (関連語対) である.
(2) 関連がある単語対である.
(3) 関連がない単語対である.
(4) 単語対ではない.

判定 1,2,3,4 はそれぞれ 2,1,-1,-2 の点数を持ち, 総合的な判定はこれらの合計について, 4 点以上ならば「関連語対である」, -6 点以下ならば「単語ではない」とした. この点数は, 「関連語対である」ならば, 「単語対である」と判定される対でありかつ, 五人のうち三人が「関連語対である」と判定し, 残り二人が「関連がない単語対である」と判定しても総合的に「関連語対である」と判定するように設定した. また, 「単語対ではない」ならば, 五人のうち一人が「関連語対である」と判定しても残りの四人が「単語対ではない」と判定すれば総合的に「単語対ではない」と判定するように設定した. これは, 「単語対ではない」という判定に特別に高い閾を設けるためである.

4.3 実験結果

表 4にそれぞれのテキスト集合から調べる前後文字列の長さを 2,3,4,5,6 とした場合, スコア関数その 1 またはその 2 を用いて得られた単語対の数を示す. 表 5に, 調べる前後文字列を長さ 2 とした場合に得られた単語対からランダムに選んだ 500 対について, 人間によって有用性を判定した結果を示す. ただし, CIRB_1 については得られる単語対の数が 500 以下であるため, 得られた単語対をすべて判定の対象とした. 各欄には, 500 対のうち調べる前後文字列の長さに対して得られた単語対の数を分母とした場合, 有用と判定された対の割合（適合率）を括弧外に示す. たとえば, NTCIR1 において, 長さ 3 の場合, 500 対のうち 170 対がシステムによって得られた単語対の数であった.

表 4　得られた単語対の数

テキスト集合	スコア関数	2 文字	3 文字	4 文字	5 文字	6 文字
NTCIR1	その 1	1448	547	230	75	38
	その 2	2849	1098	443	150	79
NTCIR2g	その 1	8442	3564	2061	766	173
	その 2	14822	6342	3489	1351	374
NTCIR2k_1	その 1	11399	3589	1458	590	270
	その 2	26743	9696	3982	1564	788
NTCIR2k_2	その 1	10469	3183	1297	463	201
	その 2	18936	6104	2277	827	352
NTCIR2k_3	その 1	13204	3902	1396	473	214
	その 2	22374	7064	2452	813	375
mai1991	その 1	4112	1924	1154	669	423
	その 2	5538	2610	1517	970	598
mai1992	その 1	1339	604	318	207	96
	その 2	1761	807	400	263	125
mai1993	その 1	1822	1053	510	292	140
	その 2	3045	1787	1026	601	374
mai1994	その 1	12164	5429	2612	1271	681
	その 2	15549	7161	3473	1685	889
CIRB_1	その 1	122	24	14	11	8
	その 2	224	40	16	13	9

このとき, 人間によって有用と判定された対の数が 127 であったため, 適合率は 127/170*100=約 74.7%となる. また, 括弧内には「単語対である」と判定された対の割合を示す. たとえば, NTCIR1 において, 長さ 3 の場合, 500 対のうち 170 対がシステムによって得られた単語対の数であった. このとき, 人間によって「単語対である」と判定された対の数が 158 であったため, この割合は 158/170*100=約 92.9%となる.

まず, スコア関数について考察する. 表 4から, 得られる単語対の数について, スコア関数その 1 よりその 2 のほうが約 25-50%多く単語対を得ることができることがわかる. しかし, 表 5から, スコア関数その 1 のほうがその 2 よりも適合率が高い場合が多いことがわかる. しかし, スコア関数その 2 のほうが 10%以上も適合率が高い場合もあるが, 実際にシステムから得られる単語対の数を考慮すると, その 1 の適合率と同程度である. このことから, スコア関数その 1 を用いて関連語リストを構築したほうが有用な関連語リストを得られる場合が多いと考察する.

次に, 調べる前後文字列の長さについて考察する. 表 4,5から, 長さ 3 以下では得られる単語対の数が多いが, テキスト集合によっては適合率が長さ 4 に比べ極端に少なく, 長さ 5 以上では適合率は高いが, 得られる単語対の数が長さ 4 に比べ極端に少ないことがわかる. このことから, 本論文では, 得られる対の数と適合率を両方考慮すると, 長さ 4 が適当であると考察する. しかし, 長さ 5 の場合の適合率は非常に高く 80.0%の場合もあり, 適合率を優先するのであれば, 長さ 5 が適当である.

上記のスコア関数と調べる前後文字列の長さについての考察を踏まえて, スコア関数その 1 を用いて長さ 4 の場合に得る単語対について見ると, 「単語対である」と判定される単語対の割合は, NTCIR では 75.0-91.8%で非常に高く, 新聞記事では 61.1-78.8%, 中国語新聞記事では 50.0%であった. そして, 適合率を見ると, NTCIR では 42.9-77.0%, 新聞記事では 11.1-28.7%, 中国語新聞記事では 50.0%であった. この二種類の値から, 「単語対である」と判定される単語対についての適合率を考えると, NTCIR では 54.6-83.9%, 新聞記事では 16.4-47.0%, 中国語新聞記事では 100.0%であった. このことから, 中国語新聞記事においては「単語対である」と判定される単語対であれば, 関連語対であると判定されるということがわかる. 一方, 日本語新聞記事では他のテキスト集合に比べ適合率が低かった. このことについては次節で単語列の分析を行い, 追求する.

また, 表 5から, 本実験で用いたキーワード抽出システムが「単語ではない」と判定される単語を切り出す割合は, NTCIR では約 10-15%, 新聞記事では約 20-45%, 中国語新聞記事では約 25-75%であることがわかる. これは, キーワード抽出システムが情報検索に有効なキーワードを抽出するために構築されたシステムであるため, 単語ではない文字列を切り出すことが原因である. ここで「単語ではない」と判定された単語対については次節で分析する.

表 5　各テキスト集合における実験結果

テキスト集合	スコア関数	2 文字	3 文字	4 文字	5 文字	6 文字
NTCIR1	その 1	66.6(88.6)	74.7(92.9)	77.0(91.8)	80.0(86.7)	70.0(80.0)
	その 2	65.8(86.0)	75.4(94.2)	76.3(96.1)	52.4(90.5)	38.5(84.6)
NTCIR2g	その 1	52.4(84.8)	60.3(87.0)	65.3(89.8)	73.0(89.2)	40.0(80.0)
	その 2	44.0(88.0)	53.3(91.5)	62.1(90.3)	64.9(83.8)	50.0(70.0)
NTCIR2k_1	その 1	52.4(85.6)	56.8(89.9)	60.4(94.3)	77.8(100.0)	66.7(100.0)
	その 2	35.0(85.8)	41.4(91.1)	39.7(93.1)	51.9(100.0)	28.6(100.0)
NTCIR2k_2	その 1	46.4(84.4)	52.7(84.0)	46.2(75.0)	61.5(76.9)	57.1(57.1)
	その 2	44.8(86.2)	48.2(82.5)	50.0(63.2)	56.5(69.6)	45.5(54.5)
NTCIR2k_3	その 1	39.2(86.8)	49.0(86.7)	42.9(78.6)	50.0(75.0)	33.3(66.7)
	その 2	43.6(86.8)	54.1(88.5)	61.2(83.7)	71.4(78.6)	77.8(88.9)
MAI1991	その 1	35.4(79.4)	22.6(75.8)	24.0(78.8)	24.0(64.0)	23.9(63.0)
	その 2	35.0(80.2)	32.5(78.7)	30.5(83.8)	35.5(84.9)	40.0(86.0)
MAI1992	その 1	20.1(67.6)	20.5(56.8)	25.4(63.4)	18.8(62.5)	16.7(55.6)
	その 2	22.0(70.4)	19.1(63.1)	20.2(56.0)	12.2(55.1)	16.0(76.0)
MAI1993	その 1	16.0(68.0)	8.5(64.8)	11.1(67.5)	6.1(63.3)	7.1(64.3)
	その 2	11.8(71.0)	10.6(75.4)	5.6(80.0)	1.9(83.8)	3.1(87.7)
MAI1994	その 1	19.4(54.8)	27.3(58.8)	28.7(61.1)	40.0(66.0)	30.8(61.5)
	その 2	14.6(52.8)	19.2(56.7)	25.8(61.9)	31.9(59.6)	28.0(56.0)
CIRB_1	その 1	37.7(73.0)	47.8(62.5)	50.0(50.0)	26.4(36.4)	25.0(25.0)
	その 2	36.7(74.6)	45.0(70.0)	50.0(50.0)	46.2(46.2)	22.2(22.2)

5. システムが判断した関連語対の分析

本節では, システムが判断した関連語対はどのような関連を持っているかを分析する. まず, NTCIR1 から得られた関連語対の一部を表 6に示す. 1-11 番は同じものを表す単語同士ではないが, 同じように使われる単語対である. 特に 5 番や 11 番は専門分野ならではの関連語対である. 本システムでは, 実際に同義語や類義語と呼ばれる関連語より, このような関連語対を多く抽出する. 12-17 番は同義語や類義語, 省略形の関連語対である. 18-24 番は表記がカタカナやひらがな, 文字が追加されたもので異なる関連語対である. これらは一般に表記の揺れといわれる. この表記の揺れは経験的に知ることが多い. 25-28 番は文字コードが異なる関連語対である. これらも表記の揺れに属する. 29-33 番は反義語や同じ上位語を持つ関連語対である. これらは辞書に記載されていることが多い. ここに示したのは NTCIR1 から得られた関連語対であるが, その他の NTCIR データから得られる関連語対はほとんど以上の五つに分類される. これは, NTCIR のテキスト集合は論文抄録であることが多くな要因である. 論文にはその論文のキーワードとなる単語が文書に多く含まれる傾向にある. したがって, その単語がうまくキーワード抽出システムによって切り出され, 語の統一を故意的に図っている文書が多いので, 関連語対の判定がしやすく, 人間によって関連語対であると判定される単語対を多く得ることができるためである. このことから, 本システムは NTCIR のようなテキスト集合用の関連語リストを作成することに有用であると考察する.

次に, 毎日新聞記事から得られた関連語対の一部を表 7を示す. 新聞記事データでは, 人名, 社名, 地名, 団体名に関する関連語対が非常に多かった. 一般用語に関するものは少なかった. システムから名前ばかりの単語対が得られるため, 判定者は全員一致で「関連語対である」と判定される関連語対が多かった. 一方,「単語対である」と判定されているが,「関連語ではない」と判定される対が多かった. このような単語対の多くは漢数字や数字でできた単語対であった. たとえば,「６１．２キロ, ５０．８キロ」という単語対に対して, 判定者五人のうち二人は「関連語対である」と判定し, 残りの三人は「単語対ではない」と判定するため, この対は総合的に「単語対である」と判定されるが,「関連語対である」とは判定されない. 新聞記事データにおいて, 特に mai1993 では, このような単語対が多く得られるために適合率が低いことがわかった.

6. おわりに

本論文では, 未知語, 既知語を問わず単語を未踏テキスト中のキーワードとして抽出し, かつその単語を理解することに役立つ関連語リストを未踏テキスト用シソーラスとして自動構築する手法を提案した. この手法は辞書を用いずに, テキストから単語を切り出し, その単語がテキスト集合において同じように使われるかを判定することによって, 対象としたテキスト集合に特化した関連語リストを生成する. これに伴い, 辞

表 6　NTCIR1 から得られた関連語対の一部

識別番号	単語 1	単語 2	識別番号	単語 1	単語 2
1	体育館	校舎	17	静磁波	静磁前身体積波
2	文字	単語	18	メッキ	めっき
3	明るさ	輝度	19	レーダー	レーダ
4	記号	ＬＩＳＰ	20	ダイバーシティ	ダイバーシチ
5	表情	顔画像	21	炭化珪素	炭化ケイ素
6	列車	鉄道	22	蛋白質	タンパク質
7	飛行	航空	23	粘土	粘性土
8	ゲノム	ＤＮＡ	24	遺伝的アルゴリズム	遺伝アルゴリズム
9	複合名詞	名詞句	25	靭性	靱性
10	象牙質	エナメル質	26	不攪乱	不撹乱
11	連接	共起	27	被爆	被曝
12	プラズマディスプレイ	ＰＤＰ	28	頸部	頚部
13	授業	講義	29	コンバータ	インバータ
14	推敲支援	校正支援	30	冷房	暖房
15	廃棄物	ごみ	31	ＳＲＡＭ	ＤＲＡＭ
16	増幅器	アンプ	32	構文解析	形態素解析

表 7　毎日新聞記事から得られた関連語対の一部

識別番号	単語 1	単語 2	識別番号	単語 1	単語 2	識別番号	単語 1	単語 2
1	西岡氏	小泉氏	8	積水ハウス	ユニチカ	15	上告	棄却
2	小渕	橋本	9	トヨタ	ヤナセ	16	判決	訴訟
3	ジーコ	アルシンド	10	エイズ	ＨＩＶ	17	暴投	四球
4	セルビア人	クロアチア人	11	痴呆	痴ほう	18	先住民	アイヌ
5	中国	台湾	12	南アフリカ	南ア	19	EAEC	APEC
6	長野県	静岡県	13	ダイヤルＱ２	ダイヤル２Ｑ	20	若花田	若ノ花
7	日本ビクター	ユアサ産業	14	落札	入札	21	寄り切り	押し出し

書を用いずに, 現実的に計算できる関連語の定義を示した. この定義のもとで, 人間によって有用であると判定され, 情報検索に利用できる見込みのある関連語リストを抽出することができた. そして, 実験に用いたテキスト集合から有用な関連語リストを生成するためには, 調べる前後文字列の長さは 4 が適当であろうことを報告した. しかし, この長さはテキスト集合に依存しかつ, 得られる関連語対の数と適合率のトレードオフ問題である. また, 本研究で構築したシステムは他言語も扱うことができる, 言語独立なシステムであることを示した. 本論文における実験で得られた関連語リストを情報検索に用いた場合の評価は今後の課題とする.

参 考 文 献

1) Noriko Kando, Kazuko Kuriyama, Toshihiko Nozue, Koji Eguchi, Hiroyuji Kato, and Souichiro Hidaka, Overview of IR Tasks at the First NTCIR Workshop, *Proceedings of NTCIR1 Workshop*, Vol.1, pp.11–44, 1999.
2) 相澤彰子, 語と文書の共起に基づく特徴度の数量的表現について, 情報処理学会論文誌, Vol.41, No.12, pp.3332–3342, 2000.
3) Kenneth W. Church, Empirical Estimates of Adaptation, Coling2000, pp.180–186, 2000.
4) 田中路子, 武田善行, 仲村大也, 山本英子, 梅村恭司, 純統計処理によるキーワードの抽出実験, 第 *42* 回プログラミング・シンポジウム報告集, pp.155–158, 2001.
5) 武田善行, 梅村恭司, キーワード抽出を実現する文書頻度分析, 計量国語学, 第二十三巻二号, pp.65–90, 2001.
6) CIRB010 に関する文献参照
7) 毎日新聞社, 毎日新聞データ, 1991 年版,1992 年版,1993 年版, 1994 年版,1995 年版, 1996 年版,1997 年版,1998 年版 ,1999 年版,2000 年版.
8) 松本裕治, 北内啓, 山下達雄, 平野善隆, 今一修, 今村友明, 日本語形態素解析システム「茶筌」version 1.5.