自然言語処理 147－5
（2002．1．21）

入力された単語に関連のある文を選出するアルゴリズムの提案

青木 友幸　飯間 等　三宮 信夫

京都工芸繊維大学

本研究では，意志決定者が幾つかのキーワードを入力すると，データベース内の複数の参照文書の中からキーワードを含む文はもちろん，キーワードと強い関連を持つ単語を含む文を選出するアルゴリズムを提案する．本アルゴリズムでは単語間，文間の類似性や単語と文との間の相互依存性などを考慮して，各単語に重み付けを行うことによって，キーワードと関連の強い単語を含む文を選出する．ある一つの事例に対して本アルゴリズムを適用したところ，選出された文の中でキーワードを含まない文の平均3割以上が適合した．

An Algorithm for Selecting Sentences with Relation to Keywords

Tomoyuki Aoki　Hitoshi Iima　Nobuo Sannomiya

Kyoto Institute of Technology

In this paper, we propose an algorithm for selecting sentences, which are desired by a decision maker, from reference documents in a database. Such a sentence is defined as one having not only inputted words (keywords) but also words with relation to them. In this algorithm, the score of each word is calculated on the basis of the similarity between words and sentences. The average of the score of words in a sentence represents the desire for the sentence. The experimental results show that about 30 percent of the sentences selected without keywords are desirable for a decision maker.

1　はじめに

近年，情報のディジタル化によりテキストデータが大量に出回るようになった．それとともに，その中から有用な情報を選出するシステムへの関心が高まっている．同様に，学術研究の論文もテキスト化されるものが多く，既存の方法を学ぶことや，その分野の研究動向を知ることなどのために，論文検索に関する研究 [1] も多く行われている．しかし，現在の検索では，入力したキーワードそのものを含む文しか選出できない．例えば「アルゴリズム」を入力すると，「アルゴリズム」を含む文のみが選出され，同様の意味を持つ「手法」や「方法」を含む文が選出されない．しかし利用者の立場からは「手法」や「方法」のある文も選出して欲しい場合が多い．

そこで，本研究では，単語間，文間の類似性や，単語と文の間の相互依存性を考慮して，キーワードを含む文だけでなく，それに関連のあると思われる文までも選出するアルゴリズムを提案する．特に，本研究では，利用者が英語でアブストラクトを作成する際の支援として，参考となる文をデータベースから検索する状況を想定し，選出対象を「文」とし

た．しかし，本アルゴリズムで選出対象を「段落」や「文書」とすることも容易に実現できる．英語で書かれている論文のアブストラクトをデータベースとした事例に対して本アルゴリズムを適用し，選出結果が利用者のアブストラクト作成の参考になったかどうかを検証する．

2 文を選出するアルゴリズム

2.1 目的

意志決定者（Decision Maker: DM）が入力する幾つかのキーワード（Keyword: KW）を基に，データベースに収められている複数の参照文書（Reference Document: RD）の中から DM の望む文を選出する．ここで DM の望む文とは，単に KW を含む文だけでなく，KW と類似している単語を含む文も対象となる．

2.2 特徴

本アルゴリズムでは，KW を含む文の選出だけでなく，各 RD 内の重要と思われる単語（Distinctive Word: DW）と KW との類似度を定義することによって，KW と類似している DW を含む文も選出することができると考えられる．また，KW よりも他の DW と高い類似度を持つ DW を含む文を選出しないようにすることで，KW と類似している文が強調されると考えられる．以下の説明では KW と DW を合わせて『ターム（Term）』と呼ぶことにし，アルゴリズムで処理される単語をタームのみに限定する．

2.3 アルゴリズム全体の流れ

本アルゴリズムの全体の流れは以下の通りである．

前処理：あらかじめ各 RD から DW を決定しておく．

STEP 1：DM が数個の KW と選出したい文の数 L を入力する．

STEP 2：各ターム間の類似度を類似度計算アルゴリズムより計算する．

STEP 3：KW に対する各 DW の類似性を見るために，各ターム w の得点（スコア）$Score(w)$ をスコア計算法より計算する．

STEP 4：各文のスコアを，その文中に含まれているタームのスコアの平均値として計算する．

STEP 5：文のスコアの中から上位 L 個だけ文を選出する．

STEP 4 および STEP 5 の「文」を「段落」あるいは「RD」に変更するだけで，選出対象を「段落」あるいは「RD」に変更することができる．

2.4 前処理

円滑かつ正確にこのアルゴリズムを使うための前処理としてまず以下の処理を行うことが望ましい．

- 動詞の過去形，現在形の違いや，名詞の単数形と複数形などの違いによる多様性を防ぐために語幹だけにする．動詞や名詞の不規則変化にも対応出来るようにする．
- 冠詞や前置詞などの不要語（Stop Words）を取り除く．
- 省略形などの表記の揺れを統一する．

2 番目の項目の作業は自動で行えるが，1 番目と 3 番目の項目の作業は辞書が必要となる．

次に，アルゴリズムの対象となる DW を定めるために TF/IDF 法 [2] を用いる．TF/IDF 法とは単語 w に対する文書 d の重要度 $i(d,w)$ を計算するものである．より具体的には，w が d 中にどれだけ多く出現しているかという指標 $tf(d,w)$ と，w が全文書内でどれくらい特徴的かという指標 $idf(w)$ の二つを使用して $i(d,w)$ を計算する．以下に計算式を示す．次式で nd は文書の総数である．

$$tf(d,w) = d \text{ における } w \text{ の出現回数} \quad (1)$$

$$idf(w) = \log \frac{nd}{w \text{ の出現する文書の数}} \quad (2)$$

$$i(d,w) = tf(d,w) \cdot idf(w) \qquad (3)$$

全単語について各文書に対する重要度を計算し，各文書毎に $i(d,w) > \log nd$ を満たす単語を DW とする．すなわち，全文書で 1 回だけ現われる単語の重要度より大きい値を持つ単語を DW とする．

2.5 ターム間の類似度の計算

2.5.1 類似度の定義

語間関係の定義は，文中における単語の用法の類似度（similarity）からなると考えられる．Karov らは単語の意味の曖昧さを解消するために，単語間の類似度を計算するアルゴリズムを提案した [3]．また橋本は，単語間の関係による単語のグループ化の構造が会話によってどのように変化するかを見るために，Karov らの方法を利用した [4]．本研究でも Karov らの方法に基づいてターム間の類似度を計算する．

Karov らの方法は，文中での単語の使われ方から単語間の類似度を計算する．類似度を計算する鍵となる考え方は，単語と文の間の相互依存性（これを親和度（affinity）と呼ぶ）である．それは，“類似した単語は類似した文に現れ，また，類似した文というのは類似した単語で構成されるものである” という考え方である．この関係は図 1 の通りに表される．

ここで図 1 に現れる各変数は，次のように定義される．

- $aff_n(w,s)$：繰り返し計算回数 n における単語 w の文 s に対する親和度
- $aff_n(s,w)$：繰り返し計算回数 n における文 s の単語 w に対する親和度
- $sim_n(si,sj)$：繰り返し計算回数 n における文 si と文 sj の類似度
- $sim_n(wi,wj)$：繰り返し計算回数 n における単語 wi と単語 wj の類似度

各変数の最大値は 1 である．

図 1 を用いて類似度計算アルゴリズムを概説する．まず，各類似度に初期値を与える．単語 w の w（自己）に対する類似度 $sim_0(w,w)$，文 s の s（自己）に対する類似度 $sim_0(s,s)$ をそれぞれ 1 とし，それ以外の類似度を 0 とする．初期値が与えられれば，それらの値を用いて各親和度 $aff_0(s,w)$，$aff_0(w,s)$ を計算する．さらにそれらの親和度の値を用いて各類似度 $sim_1(wi,wj)$，$sim_1(si,sj)$ を計算し，前の値を更新する．さらにそれらの類似度の値を用いて各親和度 $aff_1(s,w)$，$aff_1(w,s)$ を計算し前の値を更新する．このように，この手法では繰り返し計算を行って，語間・文間類似度を強めていく．

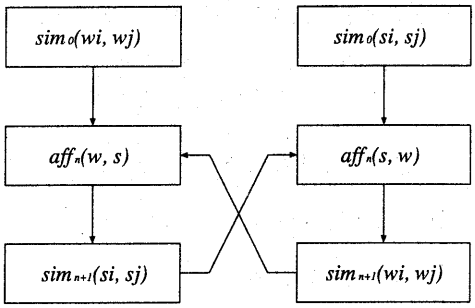

図 1: 類似度 $sim_n(i,j)$ と親和度 $aff_n(i,j)$ の関係

また，Karov らは，文の単語数が少ないほどその文の中の単語は重要であり，多くの文書で出現するような単語はあまり重要でないと考え，重要度に関する重みを各類似度に掛けるようにしている．

2.5.2 類似度計算アルゴリズム

Karov らのアルゴリズムに基づいて，本研究の類似度計算は以下のように行う．

STEP 1：文の総数 ls，単語の総数 lw，タームの種類の数 wk を求める．また各文 s のターム数 $lg(s)$ をそれぞれ求める．

STEP 2：ターム w および文 s の重みの要因 $factor(w)$，$factor(s)$ を，全てのターム w と文 s に対して次式より求める．

$$factor(w) = 1 - (w \text{の出現回数})/lw \qquad (4)$$

$$factor(s) = 1/lg(s) \qquad (5)$$

STEP 3：ターム w の文 s に対する重み $weight(w,s)$，および文 s のターム w に対

する重み $weight(s,w)$ を次式より求める．

$$weight(wi,s) = \frac{factor(wi)}{\sum_{wj \in s} factor(wj)} \quad (6)$$

$$weight(si,w) = \frac{factor(si)}{\sum_{sj \ni w} factor(sj)} \quad (7)$$

STEP 4：初期値を次式の通りに与える．

$$\begin{aligned} sim_0(wi,wj) &= 1 \quad (wi = wj) \\ sim_0(wi,wj) &= 0 \quad (wi \neq wj) \\ sim_0(si,sj) &= 1 \quad (si = sj) \\ sim_0(si,sj) &= 0 \quad (si \neq sj) \end{aligned} \quad (8)$$

STEP 5：繰り返し計算回数 $n = 0$ とする．

STEP 6：タームの文に対する親和度 $aff_n(w,s)$ を全ての w と s について次式より求める．

$$aff_n(w,s) = \max_{wj \in s} sim_n(w,wj) \quad (9)$$

STEP 7：文のタームに対する親和度 $aff_n(s,w)$ を全ての s と w について次式より求める．

$$aff_n(s,w) = \max_{sj \ni w} sim_n(s,sj) \quad (10)$$

STEP 8：文間類似度 $sim_{n+1}(si,sj)$ を全ての si と sj について次式より求める．

$$sim_{n+1}(si,sj) = \sum_{w \in si} weight(w,si) \cdot aff_n(w,sj) \quad (11)$$

STEP 9：語間類似度 $sim_{n+1}(wi,wj)$ を全ての wi と wj について次式より求める．この値は後に示すスコア計算で用いられる．

$$sim_{n+1}(wi,wj) = \sum_{s \ni wi} weight(s,wi) \cdot aff_n(s,wj) \quad (12)$$

STEP 10：全ての $sim_{n+1}(wi,wj)$ がある一定値 (0.9999) を超えれば，その時の繰り返し計算回数を N，すなわち $N = n+1$ とし，計算を終了する．そうでなければ，$n = n+1$ として STEP 6 に戻る．

表 1: 単語間のスコア計算法

wi	wj	$score(wi)(wj)$
KW	KW	$+Asim(wi)(wj)/KN$
KW	DW	$+Asim(wi)(wj)/DN$
DW	KW	$+Asim(wi)(wj)/KN$
DW	DW	$-Asim(wi)(wj)/DN$

2.6 スコア計算法

2．5項のアルゴリズムでは全ての語間類似度が最終的に1に向かって収束するために，各類似度の差が明確でない．そこで，累積語間類似度を求めることにより各類似度の差を明確にする．

まず，ターム wi の wj に対する累積語間類似度 $Asim(wi)(wj)$ を次式で求める．

$$Asim(wi)(wj) = \sum_{n=0}^{N} sim_n(wi)(wj) \quad (13)$$

次に，ターム wi の wj に対するスコア $score(wi)(wj)$ を表1より求める．ただし表1で KN はKWの個数，DN はDWの個数である．これにより，KWを含むターム間のスコアが増え，またKWを含まないターム間のスコアが減ることによりKWを含む文が選出されやすくなる．

最後に，ターム w のスコア $Score(w)$ を，次式で求める．

$$Score(w) = \sum_{wi=1}^{lw} score(w)(wi) \quad (14)$$

一例として，表2 (a) のようにターム $w1$〜$w3$ の各累積語間類似度が与えられている場合の各タームのスコアを求めると次のようになる．

$$\begin{aligned} Score(w1) &= +5.000/1 + 2.210/2 + 3.522/2 \\ &= +7.866 \\ Score(w2) &= +2.010/1 - 5.000/2 - 2.854/2 \\ &= -1.917 \\ Score(w3) &= +3.814/1 - 2.018/2 - 5.000/2 \\ &= +0.305 \end{aligned} \quad (15)$$

表 2: スコア計算の例（$KN = 1$, $DN = 2$）

(a): 累積語間類似度 $Asim(wi, wj)$

		wj		
		$w1$	$w2$	$w3$
wi	$w1$ (KW)	5.000	2.210	3.522
	$w2$ (DW)	2.010	5.000	2.854
	$w3$ (DW)	3.814	2.018	5.000

(b): スコア $Score(w)$

$w1$	$w2$	$w3$
7.866	-1.917	0.305

これより，同じDWでも，$w3$ の方が $w2$ と比べてスコアが高いことがわかる．これは，$w3$ はKWである $w1$ との類似度が高く，DWである $w2$ との類似度が低いからである．

3 適用例

国際学会ACC（American Control Conference）のOptimizationとManufacturing Systemに関するセッションの原稿のアブストラクト３５文書に対して，上記のアルゴリズムを適用する．文の数は１８１文である．規模がそれほど大きくないので，２．４項の前処理の１番目と３番目の作業は手動で行った．前処理の結果，DWの数は２０９種類となった．

3.1 KWの選出

DMの入力するKWをKW 1として以下に示す．各KWの特性を見るためにKW 2～KW 7も考慮して，それらのKWの組合せも考える．また，括弧内の数字はそのKWが含まれている文の数を表している．また，選出する文の数をそのKWが含まれている文の数の３倍とした．(例えばKW 1では１２×３＝３６文を選出した．)

- KW1 : multiobjective, vehicle, routing （１２）
- KW2 : multiobjective, vehicle （８）
- KW3 : multiobjective, routing （７）
- KW4 : vehicle, routing （９）
- KW5 : multiobjective （３）
- KW6 : vehicle （５）
- KW7 : routing （４）

3.2 評価方法

検索システムの評価などに適合率 [5] [6] がよく用いられる．本研究では，適合率を，選出された文のうちどの程度の文がDMの求める文であったかを示す指標とする．しかし，DMの求める文はDMの意向により異なるので，数人に同一KWでの結果に対してアンケートを取り，適合率を求めることで評価した．このアンケートでは，選出された文一つずつに対して適切な結果となっているか否かを判定してもらい，適切であるとみなされた文の割合を求めた．

3.3 結果と考察

３．１項の各KWに対して，このアルゴリズムを適用した．その結果に対するアンケートから適合率の平均値を求めたものを表３に示す．適合率１とは選出された文の中から良いと判断された文の割合を，適合率２とは選出された文の中でKWを含まない文の中から良いと判断された文の割合を表す．

表３の適合率２から，全体的には，選出された文の中でKWを含まない文のほぼ３割以上が適合していると言える．KW別に見ると，KWにより適合率に差が出ていることが分かる．KWが一つのときはKW 7が最も良く，KW 5が最も悪い結果となった．ところが，それらを組み合わせたKWで計算すると，１個のKWのときには値の悪かったKW 5を含んでいるKW 1～KW 3の方がKW 4より良い結果となっている．このことから，KW 5が各RDの中で，他のタームとのつながりが強いということがわかる．

表 3: 各 KW に対する結果文の適合度（平均値）

KW	適合度 1	適合度 2
KW 1	0.472	0.291
KW 2	0.528	0.354
KW 3	0.540	0.357
KW 4	0.444	0.241
KW 5	0.519	0.278
KW 6	0.600	0.400
KW 7	0.666	0.500

次に，実際にどのような文が高いスコアになるのかを，一つの例をあげて考察するために，KW 7 を入力した時の上位１２個までの実際に選出された文を以下に示す．番号は各文に与えられた固有番号である．また，アンダーラインは KW であることを，イタリックで書かれた番号は KW と関連のある文であることを示す．

154: As an application a general routing resource booking problem is considered.

168: These sets and operators imply a very compact and efficient representation of complex routing specifications.

148: A deadlock avoidance algorithm for flexible manufacturing systems containing both multiple capacity resources and mixed choices in part routing is presented.

162: A modeling framework for general routing and resource booking problems is presented.

155: A specification language is presented which makes it easy to handle both multiple and alternative resource booking problems.

167: Multiple as well as alternative resources are easily specified utilizing sets of resources.

163: The task is to specify desired routes for individual objects (products, data packets, vehicles), which are to be served by a number of shared resources (machines, computers, communication links).

164: A high level language is presented in order to simplify the specification of desired routes.

79: We discuss some related results on fixed gain SPSA and describe an application of the method to a resource allocation problem.

169: In fact, the representation is suitable for verification and synthesis based on symbolic tools such as BDDs.

91: It is assumed that the target field information is communicated to all the elements of the swarm as it becomes available.

92: A network flow optimization problem is posed whose readily obtained solution yields the optimum resource allocation among the air vehicles in the swarm.

まず，KW を含む文は必ず選出されていることがわかる．次に，KW を含まない文の中で，文 *163* が DM の求める文として選出されている．これは KW である **routing** を含む 文 *154*, *148*, *162* に **resouce** という単語があり，この単語が 文 *163* にも存在することが主たる原因である．しかしその一方，DM の求めていない文である 文 *91* なども選出されている．

4 おわりに

本稿では，単語間，文間の類似度や単語と文の相互依存性を用い，キーワードを含む文だけでなく，キーワードに関連のある文を選出するアルゴリズムを提案した．ある事例に適用した結果，選出された文の中で，キーワードを含まない文の平均約 3 割以上がキーワードに関連のある文であった．

今後の課題として，本研究では DM の希望しない文が含まれているので，これらを減らすようにアルゴリズムを改良する必要がある．

参考文献

[1] Y.Mihira : http://www.vv.ics.keio.ac.jp/~sanpei/referrelation/

[2] G.Salton : Automatic text processing : the transform, analysis, and retrieval of infomation by computer, Addison Wesley, (1989).

[3] Y.Karov and S.Edelman : Similarity-based Word Sense Disambiguation, *Association for Computational Lingustics*, Vol. 24, No. 1, pp. 41–59 (1998).

[4] 橋本敬：動的言語観に基づいた単語間関係のダイナミクス, 第 20 回システム工学部会研究資料, pp. 93–98 (2000).

[5] D.Lewis : Evaluating text categorization, *Proc. of Speech and Natural Language Workshop*, pp. 312–318 (1991).

[6] I.Witten, A.Moffat and T.Bell : Managing Gigabytes, *Compressing and Indexing Documents and Images*, New York, (1994).