Vol.2011-SLP-89 No.16
2011/12/19

Suffix Array を用いた高速 STD における キーワード分割に関する理論的検討

桂田浩一[†]　入部百合絵[†]　新田恒雄[†]

筆者らはこれまで Suffix Array を用いた高速音声キーワード検索法を提案し，大規模音声データベースを対象に高速性を示してきた．この手法では Suffix Array を利用すると同時に，反復深化的探索の導入，長いキーワードの分割検索により高速な検索結果の取得を実現している．このうちキーワードの分割検索を行う際には，分割の有無によって検索結果が変化しないよう検索閾値の調整を行っているが，これまではキーワードを等分割した場合の理論的検討しか行っていなかった．本稿ではキーワードを任意の分割数，分割長にして検索閾値を調整する場合に，閾値が満たすべき条件について理論的に検討する．

Theoretical Basis of Keyword Division in the Fast STD Using Suffix Array

Kouichi Katsurada[†]　Yurie Iribe[†]　and　Tsuneo Nitta[†]

We have proposed a spoken term detection (STD) method using a suffix array, and have evaluated its search speed on a large speech database. This method employs iterative lengthening search and keyword division for quicker search, as well as utilizing the suffix array as a data structure. In this method, we control the threshold value to keep the search results unchanged when searching for the divided keywords. However, we have only considered the cases where the original keyword is equally divided. In this paper, we investigate theoretical conditions to be satisfied by the threshold values when the keyword is divided into arbitrary length and number of sub-keywords.

1. はじめに

ストレージの大容量化，低価格化に伴い音声および動画像のコンテンツがあらゆる場所に蓄積されるようになった．最も大量に蓄積されているのが web で，大手の動画投稿サイトでは数百万時間以上の動画像コンテンツが保持されており，今後も加速度的に増加することが予測される．Web に限らず，テレビ局，ラジオ局，放送大学等の機関や，コールセンター，各種会議においても膨大な量の音声・動画像コンテンツが日々蓄積されている．こうした動的メディアを効率よく利用するには，これらのコンテンツが含む音声情報に対する検索技術が非常に重要である．例えば動画像の一シーンの発話をキーワードや文章で検索できれば，動画像提供サービスの利便性は格段に向上すると考えられる．このような背景から筆者らは web や放送局，放送大学，コールセンター，議会等での検索サービスを想定し，一万時間規模の音声ドキュメントからキーワードを短時間で検出する高速キーワード検索法を提案してきた [1)-4)]．

我々の手法では二分探索が可能な Suffix Array[5)]をデータ構造として用い，これに DP マッチングを適用する方法 [6)]を用いることで高速な曖昧検索を実現している．また Suffix Array の利用に加えて，反復深化的探索，キーワード分割の導入により更なる高速化を実現している．これらの要素技術のうちキーワードの分割においては，分割して検索した場合に分割しない場合と同一の検索結果が得られるよう検索閾値の調整を行っている．これは，閾値を調整しないと検索性能が悪化することがあるためである．この調整によって検索性能の悪化が防がれ，高速かつ高性能の検索システムを構築できたが，これまではキーワードを等分割した場合の検討しか行っていなかった．そこで本稿ではキーワードを任意の分割数，分割長にした場合に検索結果を変化させないための閾値の条件について理論的に検討する．

2. Suffix Array を用いた高速 STD 法

2.1 Suffix Array を用いた曖昧検索

Suffix Array(接尾辞配列)[5)]は，テキスト中の全ての suffix(接尾辞)を辞書順にソートしたもので，テキスト検索で検索キーワードを効率的に見つけ出すためのデータ構造である．例えば，``abracadabra"というテキストに対して Suffix Array を構築すると，図 1 のようになる．図中の index はその suffix がテキストの何文字目から始まるかを示す．ここでキーワード``bra"が出現する位置を検索したい場合，Suffix Array を二分探索すると index が 8 と 1 の位置に出現することが効率的に得られる．Suffix Array ではソートされた index のみを保持すれば良いため，必要なデータ領域を小さくすることができ，また任意の文字列の出現位置を文字単位で検索できる．

[†] 豊橋技術科学大学
Toyohashi University of Technology

Text	a	b	r	a	c	a	d	a	b	r	a
Index	0	1	2	3	4	5	6	7	8	9	10

Suffix	Index
a	10
a b r a	7
a b r a c a d a b r a	0
a c a d a b r a	3
a d a b r a	5
b r a	8
b r a c a d a b r a	1
c a d a b r a	4
d a b r a	6
r a	9
r a c a d a b r a	2

Suffix array

図 1　Suffix Array

山下らは Suffix Array を用いたテキスト曖昧検索のアルゴリズム [6]を提案している．山下らのアルゴリズムでは，Suffx Array を木構造に見立てて探索を行う．木構造の根から全てのパスに対して DP マッチングを行い，各パスと検索キーワードとの Cut-off 距離を求める．その際，Cut-off 距離がある閾値を越えたら，そのノード以下の部分木の探索を打ち切る``枝刈り"を行う．この枝刈りを行うことで高速な曖昧検索を実現している．Cut-off 距離は次式で定義される．

$$cutdist(m) = min_{1 \le k \le K} P_{k,l} \qquad (1)$$

ここで m は現在のノード，K はキーワード長であり，$P_{k,l}$ は DP マッチングによるキーワード $a_1a_2...a_k$ と系列 $b_1b_2...b_l$ の間の累積距離を表す．$b_1b_2...b_l$ は根からノード m までに辿った枝に設定された文字の系列である．

探索において枝が刈られることなく，検索キーワードとの距離 $P_{k,l}$ が閾値以下となるノードに到達したら，そのノードを根とする部分木に属する suffix の index をキーワードの出現位置として出力する．例として，テキスト"abracadabra"からキーワード``bra"を閾値を 1 として検索したときの途中経過を図 2 に示す．``ac"の枝と``ad"の枝は Cut-off 距離が閾値を越えたため枝刈りが行われ，それ以降は探索されない．また，``bra"の枝は検索キーワードとの Cut-off 距離が閾値内であるため，この枝の部分木のうち累積距離が最小値となる``bra"と``bracadabra"の index が検索結果として出力される．

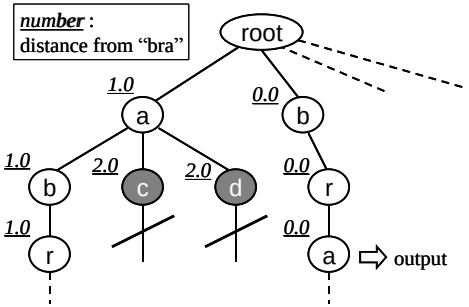

図 2　Suffix Array の探索

	a	i	u	e	o	k	...
低舌性	-	+	+	-	-	-	
高舌性	+	-	-	-	-	+	
破裂性	-	-	-	-	-	+	
破擦性	-	-	-	-	-	-	
:							

図 3　弁別特徴テーブル

2.2 音声検索の概要

提案手法は音声データに対して音声認識処理を施し，その結果得られる音素列から Suffix Array を構築して検索を行う．音声データからの音素列取得には LVCSR(Large Vocabulary Continuous Speech Recognition)を用いる．

検索で用いる DP マッチングの累積距離の定義式は以下の通りである．

$$P_{i,j} = \min \begin{cases} P_{i-1,j} + D \\ P_{i-1,j-1} + d(a_i, b_j) \\ P_{i,j-1} + I \end{cases} \qquad (2)$$

ここで，a_i は検索キーワード $a_1a_2...a_k$ 中の音素，b_j は系列 $b_1b_2...b_l$ 中の音素，$P_{i,j}$ は $a_1a_2...a_i$ と $b_1b_2...b_j$ の累積距離，$d(a_i, b_j)$は a_i, b_j 間の局所距離を表す．D と I はそれぞれ脱落，挿入ペナルティである．

音声認識では音素によって誤り易さが異なる．そこで，音素間の音響的距離を適切に表す弁別特徴 [7] から求めた距離を局所距離 $d(a_i, b_j)$に用いることにした．弁別特徴とは調音様式・調音位置から音素を弁別したもので，図 3 に示すように+または－を取る 15 次元の素性により音素が定義される．本研究では各音素間でこの素性のハミング距離を求め，局所音素間距離とした．検索の際には累積音素間距離を閾値として設定し，閾値内の音素列を検索結果として検出することにした．

3. キーワード分割

3.1 キーワード分割と問題点

2.1 節で説明したアルゴリズムでは，枝刈りの閾値に対して処理時間が指数関数的

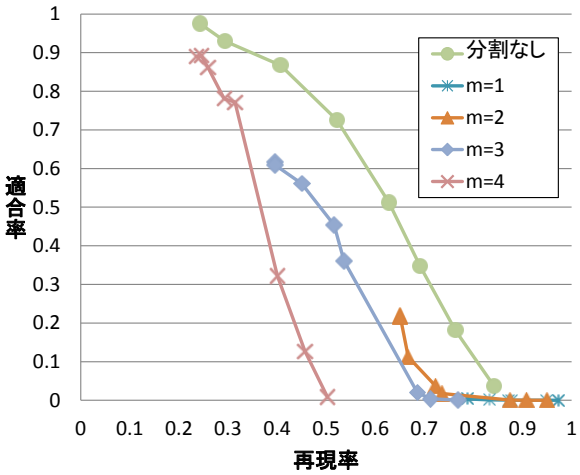

図 4　　単純な閾値付与法と検索法を採った場合の検索性能

に増加することが，山下らによって確認されている．これは閾値が増加すると探索範囲が指数的に広がるためである．閾値は検索キーワードの長さに比例して増加させる必要があるため，検索キーワード長に対して指数的に処理時間が増大する．そこで，この問題を解決するためにキーワードを分割し，分割キーワードを元のキーワードの代わりに検索する手法を導入する．

このとき，元のキーワードに与えられた閾値に基づき各分割キーワードの閾値を改めて設定する必要がある．また，各分割キーワードの検索結果が得られた後に，それらの結果からキーワード全体の検索結果を得る必要がある．ここで最も単純な検索手順を考えると，次の様になる．

1. 閾値 T のキーワードを n 分割するとき，分割キーワードの閾値を T/n とする．
2. n 分割した分割キーワードのうち，m $(1 \leq m \leq n)$個以上見つかった箇所をキーワード全体の検索結果とする．

この方法を採ったときの検索性能を実験的に確認した．実験では日本語話し言葉コーパス(CSJ)から 24 音素の既知語を 100 個選択し，再現率と適合率で検索性能を計測した．キーワードの分割数は n=4 とした．図 4 に結果を示す．

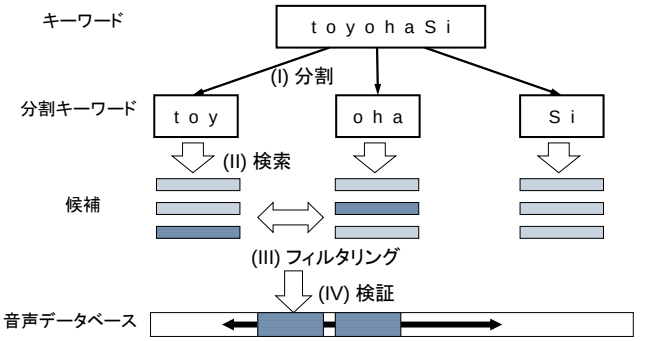

図 5　　検索の流れ

図に示すように，分割検索した場合，分割しない場合と比較して性能が大幅に悪化している．これは，分割しない場合には DP マッチングによって累積音素間距離以内の結果が過不足なく検索できているのに対して，分割検索した場合には，部分的にはキーワードと類似しているものの全体的に大きく離れた結果が検索されることや，閾値以内であるにもかかわらず検索結果として得られない検索対象領域があるためである．したがって検索性能の悪化を防ぐには，分割しない場合と同様の結果が得られるよう，分割方法，分割キーワードの閾値，検索アルゴリズムを工夫する必要がある．

3.2 分割検索のアルゴリズム

本手法では，キーワードを分割検索した場合に，分割しない場合と同一の結果が得られるよう，次の方針で検索を行うことにする．

- 閾値 T のキーワードを分割して検索する際に，累積音素間距離 T 以内のキーワードを見落とすことがないよう，分割キーワードの閾値を必要十分に大きな値にして検索を行う．
- 分割キーワードの検索結果から音声データベース中の近傍領域にあるものをフィルタリングし，その中からキーワード全体として累積音素間距離が T 以内のものを検索結果として出力する．

この方針に基づき，本手法では図 5 に示す 4 ステップで検索を行うことにする．検索は (I)分割，(II)検索，(III)フィルタリング，(IV)検証の各ステップから構成される．まず (I)分割のステップでは閾値 T のキーワードを n 個に分割する．各分割キーワー

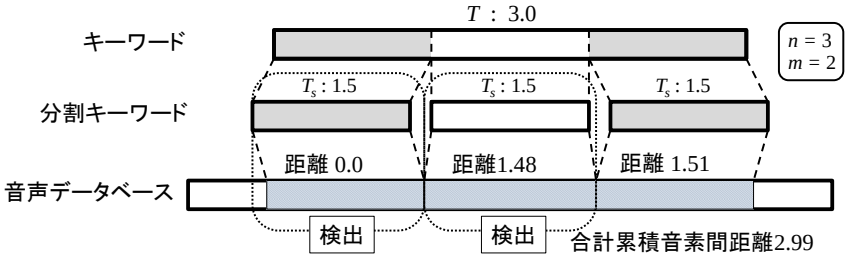

図 6　　等分割の場合の分割キーワードの閾値

ドの閾値は T_1～T_n に設定される．続く (II)検索のステップでは，n 個の分割キーワードの出現位置が音声データベースから検索される．次の (III)フィルタリングステップでは，音声データベース中の近傍領域に m 個（種類）以上の分割キーワードが検索された箇所が検出される．最後に (IV)検証のステップでは検出された領域から，キーワード全体として累積音素間距離が T 以内の箇所が抽出される．

ここで問題になるのが，各分割キーワードの閾値 T_1～T_n をどのように決定すれば良いか，という点である．以下ではこの T_1～T_n が満たすべき条件を検討する．

3.3 等分割の分割キーワードの閾値が満たすべき条件

筆者らは文献 [3) 4)]においてキーワードを等分割した場合の閾値を式(3)の通り設定すると，分割しない場合と同一の結果が得られると述べた．すなわち，キーワードを等分割にする場合，式(3)の T_s のように分割キーワードの閾値を設定することが，図 5 の (II)検索ステップにおいて累積音素間距離が T 以内のキーワードを見落とさないための条件である．

$$T_s = \frac{T}{n-m+1} \tag{3}$$

付録 A.1 に式(3)の導出過程を示す．また，図 6 に n=3，m=2，T=3.0 の場合の例を示す．この例では T=3.0 であり，3 つの区間の距離はそれぞれ 0.0，1.48，1.51 で合計累積音素間距離が 2.99 である．したがってこの区間は検出されなければならない．式(3)に従うと T_s=1.5 となり，距離が 1.5 より小さい二つの分割キーワードが検出されるため，この区間は検出される．もし T_s が式(3)の値より小さい場合，一つの区間しか検出されない場合があり得るため，閾値以下のキーワードを見落とさないためには式(3)のように設定する必要がある．

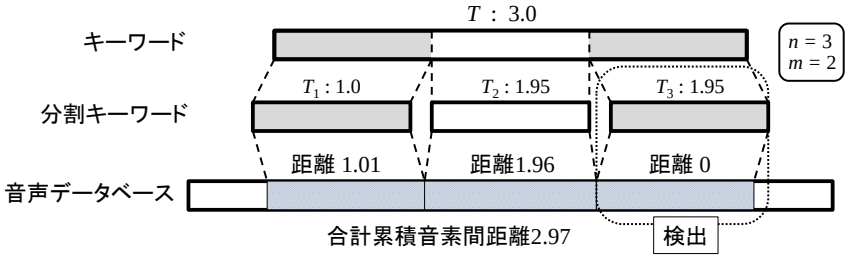

図 7　　任意の分割数，分割長の分割キーワードの閾値

3.4 任意の分割数，分割長の分割キーワードの閾値が満たすべき条件

前節の式(3)によって，キーワードを等分割した場合の閾値を定めることが可能になったが，キーワードを任意長，任意数に分割してそれぞれに異なる閾値を与える場合に，どのように閾値を付与すべきかについては，これまで議論してこなかった．そこで閾値 T のキーワードを n 個に分割して，そのうち m 個以上の分割キーワードを検出するという条件の下で，各分割キーワードの閾値 T_1～T_n が満たすべき条件を検討した．結果を条件(4)に，導出過程を付録 A.2 に示す．

$$\text{任意の} n-m+1 \text{個の} T_{i_1},\cdots,T_{i_{n-m+1}} \text{について，} T_{i_1}+\cdots+T_{i_{n-m+1}} \geq T \tag{4}$$

各分割キーワードの閾値が $T_1 \leq T_2 \leq \ldots \leq T_n$ を満たすとき，条件(4)は次の条件(5)と等価である．

$$T_1+\cdots+T_{n-m+1} \geq T \tag{5}$$

図 6 の例で示すと，任意の二つの T_s について $T_s + T_s \geq T$ であることから，条件(4)を満たす．したがって少なくとも二つ以上の分割キーワードが検索される．また，図 7 の例では $T_1 + T_2$ = 2.95 であり式(5)の $T_1 + T_2 \geq T$ を満たさない．このため，合計累積音素間距離が 2.97 であるにもかかわらず，T_3 の 1 箇所しか検出できていない．もし $T_2 = T_3$ = 2.0 であれば，$T_1 + T_2 \geq T$ となり少なくとも二つ以上の分割キーワードが検出できる．

なお，式(4)において $T_1 = T_s, \ldots, T_n = T_s$ とし，かつ不等号を等号にした場合，式(3)が求まることに注意されたい．つまり式(4)は式(3)を一般的な形にしたものであると言える．

4. まとめ

本報告では Suffix Array を用いた高速音声キーワード検索で行うキーワード分割において，分割キーワードの検索閾値が満たすべき条件を理論的に検討した．本稿で示した条件内で閾値を設定することにより，分割をしない場合と同一の検索結果が得られることが保証される．本稿で導いた条件は一つの不等式のみで表される非常に緩やかなものであることから，具体的にこの条件下で閾値を設定する方法は数多く考えられる．

そこで今後はこの条件内でどのように閾値を決定すれば最も検索時間を短縮できるかについて，実験的に検証すると同時に，キーワード内の音素の分布やキーワード長，分割キーワード検索の途中結果から自動的に最適の分割法と分割サイズを決定する方法についても検討していきたい．

参考文献

1) 手島 茂樹，桂田 浩一，新田 恒雄： “Suffix Array を用いた高速なキーワード検索”，情報処理学会研究報告 2009-SLP-77，No.3 (2009-7).
2) Kouichi Katsurada, Shigeki Teshima and Tsuneo Nitta: "Fast Keyword Detection Using Suffix Array", Proc. of INTERSPEECH2009, pp.2147-2150 (2009-9).
3) 澤田 心太，桂田 浩一，入部 百合絵，新田 恒雄： “高速音声ドキュメント検索における検索クエリ分割手法およびマッチング手法の比較評価”，第 5 回音声ドキュメント処理ワークショップ講演論文集，(2011-3).
4) Kouichi Katsurada, Shinta Sawada, Shigeki Teshima, Yurie Iribe and Tsuneo Nitta: "Evaluation of Fast Spoken Term Detection Using a Suffix Array", Proc. of INTERSPEECH2011, pp.909-912 (2011-8).
5) U.Manber and G.Myers : Suffix arrays: a new method for on-line string searches, SIAM J.Computing, vol.22, no.5, pp.935-948 (1993).
6) 山下 達雄，松本 祐治 : Suffix Array を用いたフルテキスト類似用例検索，情報処理学会研究報告 NL，vol.97，no.85，pp.83-90 (1997).
7) T.Fukuda and T.Nitta : Orthogonalized Distinctive Phonetic Feature Extraction for Noise-Robust Automatic Speech Recognition, IEICE Trans., vol.E87-D, no.5, pp.1110-1118 (2004).

付録

付録 A.1 キーワードを等分割した場合の閾値の設定法

K をキーワード，$k_1, \ldots, k_n$ を分割キーワード，S を音声データベース中の音素列，$s_1, \ldots, s_n$ を S の部分音素列，$D\ (\leq T)$ を K と S の累積音素間距離，d_i を k_i と s_i の累積音素間距離とする．一般性を失うことなく，$d_1, \ldots, d_n$ は $d_1 \leq \ldots \leq d_n$ のように昇順に並んでいるものとする．このとき，式(3)のように各分割キーワードの閾値を設定することが，累積音素間距離が T 以内のキーワードを見逃さないための条件である．

【証明】

$D \leq T$ であることから，$d_1 + \ldots + d_n \leq T$ が導かれる．この式は次の様に変形できる．

$$d_m \leq T - (d_1 + \ldots + d_{m-1}) - (d_{m+1} + \ldots + d_n)$$

$d_1 \leq \ldots \leq d_n$ より，d_m は $d_1, \ldots, d_{m-1}=0$，かつ $d_{m+1}, \ldots, d_n = d_m$ であるときに最大値となる．この場合次式が満たされる．

$$(n - m + 1)\, d_m \leq T$$

どのような条件においても最低 m 個の近傍候補を検出するには，T_s は d_m の最大値以上でなければならない．したがって式(3)が導かれる．

付録 A.2 キーワードを任意長，任意数に分割した場合の閾値設定法

K をキーワード，$k_1, \ldots, k_n$ を分割キーワード，S を音声データベース中の音素列，$s_1, \ldots, s_n$ を S の部分音素列，$D\ (\leq T)$ を K と S の累積音素間距離，d_i を k_i と s_i の累積音素間距離とする．一般性を失うことなく，$d_1, \ldots, d_n$ は $d_1 \leq \ldots \leq d_n$ のように昇順に並んでいるものとする．このとき，式(4)が次の二つの補題から明らかに成立する．

（補題１） 任意の $n - m + 1$ 個の分割キーワードの閾値 $T_{i_1}, \ldots, T_{i_{n-m+1}}$ について $T_{i_1} + \cdots + T_{i_{n-m+1}} \geq T$ が成り立つとき，$d_{j_1} \leq T_{j_1}, \ldots, d_{j_m} \leq T_{j_m}$ となる少なくとも m 個の d_{j_k}, T_{j_k} の組が存在する．

（補題２） ある $n - m + 1$ 個の分割キーワードの閾値 $T_{i_1}, \ldots, T_{i_{n-m+1}}$ について $T_{i_1} + \cdots + T_{i_{n-m+1}} < T$ が成り立つとき，$d_{j_1} \leq T_{j_1}, \ldots, d_{j_m} \leq T_{j_m}$ となる m 個の d_{j_k}, T_{j_k} の組が存在しない場合がある．

Vol.2011-SLP-89 No.16
2011/12/19

【補題１の証明】

$D \leq T$ であることから，d_1 +…+ $d_n \leq T$ が導かれる．もし m 個のd_{j_k}, T_{j_k}の組が存在しないとすると，$d_{j_k} \leq T_{j_k}$を満たす組は $m-1$ 個以下となり，$n-m+1$ 個以上の組について次式が成り立つ．

$$d_{j_1} > T_{j_1}, \ldots, d_{j_{n-m+1}} > T_{j_{n-m+1}}$$

上式より$d_{j_1} + \cdots + d_{j_{n-m+1}} > T_{j_1} + \cdots + T_{j_{n-m+1}} \geq T$となるが，これは d_1+…+ $d_n \leq T$ に反する．したがって少なくとも m 個のd_{j_k}, T_{j_k}の組が存在する．

【補題２の証明】

$T_{i_1} + \cdots + T_{i_{n-m+1}} < T$であることから，$T_{i_1} + \cdots + T_{i_{n-m+1}} + \Delta = T$ $(\Delta > 0)$と表せる．ここで$d_{i_1}, \ldots, d_{i_n}$が次の条件を満たす場合を考える．

$$d_{i_1} = T_{i_1} + \frac{\Delta}{N}, \quad d_{i_2} = T_{i_2} + \frac{\Delta}{N}, \ldots, \quad d_{i_{n-m+1}} = T_{i_{n-m+1}} + \frac{\Delta}{N} \quad (N \geq n-m+1),$$

$$d_{i_{n-m+2}} = 0, \ \ldots, \ d_{i_n} = 0$$

このとき次式が成り立つ．

$$\begin{aligned} d_{i_1} + \cdots + d_{i_n} &= d_{i_1} + \cdots + d_{i_{n-m+1}} + d_{i_{n-m+2}} + \cdots + d_{i_n} \\ &= T_{i_1} + \cdots + T_{i_{n-m+1}} + \frac{n-m+1}{N}\Delta \leq T \end{aligned}$$

したがって d_1+…+ $d_n \leq T$ が成り立つが，$d_{i_1} > T_{i_1}, d_{i_2} > T_{i_2}, \ldots, \ d_{i_{n-m+1}} > T_{i_{n-m+1}}$であるため，$d_{i_{n-m+2}} \leq T_{i_{n-m+2}}, \ldots, \ d_{i_n} \leq T_{i_n}$しか成り立たない．このとき $m-1$ 個のd_{j_k}, T_{j_k}の組しか存在しない．