単一言語コーパスにおける文の自動対応付け手法

平尾 努[†] 鈴木 潤[†]
磯崎秀樹[†] 前田英作[†]

一般に，同じ内容について書かれた異なる文書を対象として，同一内容の文間に対応付け（アラインメント）を行ったコーパスを作成することができる．こうした対応付けコーパスは，構文変形規則や言い換え規則など，文生成に関する知見を得ることができ，自然言語処理の基盤技術を開発するうえで必須である．特に，文書要約研究の立場からは，上記だけでなく，学習・評価データとしても重要な価値を持つ．こうしたコーパスを大量に収集するためには，高性能な自動対応付け手法が必要とされる．しかし，従来手法は，類似度を計算するためのテキストの基本単位や類似尺度そのものに問題があった．そこで本論文では，対応付けの基本単位として依存構造木の経路を用い，各経路に最も類似する経路を有する文を対応文とする手法を提案する．単語の言い換えを吸収するため，経路を単語とその意味カテゴリからなるノードの系列と見なし，系列パターンに基づき経路間の類似度を決定する．Text Summarization Challenge（TSC）で作成された単一文書，複数文書要約のデータを用いて提案手法による対応付け結果を人間による対応付け結果と比較した結果，単一文書要約データの場合で F 値 0.95〜0.97 程度，複数文書要約データの場合に F 値 0.72〜0.83 程度という従来手法より高い成績であった．

Automatic Sentence Alignment for Monolingual Corpora

Tsutomu Hirao,[†] Jun Suzuki,[†] Hideki Isozaki[†]
and Eisaku Maeda[†]

Monolingual aligned corpora are valuable for natural language processing. In order to generate text, we can learn various kinds of knowledge from such corpora. For instance, summary sentences aligned with sentences from original documents are useful for the study of automatic summarization. However, conventional methods are not suitable for one-to-many or many-to-one correspondence. Moreover, the similarity measure for alignment is not optimal. In this paper, we propose an automatic alignment method for these monolingual corpora. First, we transform a sentence into the set of paths in its dependency structure. Next, we calculate similarity between the paths based on ESK (Extended String Subsequence Kernel) which consider both sequential patterns and semantic labels. By using these procedures, we can derive a one-to-many or many-to-one correspondence among sentences. Experimental results using TSC (Text Summarization Challenge) corpora, which align summary sentences with original sentences, showed that our method obtained 0.95–0.97 F-measure for single document summarization data and 0.72–0.83 F-measure for multiple document summarization data.

1. はじめに

高度な自然言語処理を実現するための基盤として，書きかえによる文生成技術（いわゆる，text-to-text generation）への要求が近年特に高まってきている．文生成のための要素技術には，語彙の言い換えのように比較的単純なものから，複数の文を構文変形しつつ1文にまとめたり，1文を複数の文に分解したりするなど高度な処理を必要とするものもある．こうした文生成技術に関する研究を進めるうえで，同じ内容について書かれた異なる文書間で同一内容の文を対応付けたコーパス（対応付けコーパス）を整備することが不可欠である．大量の対応付けコーパスは，そこから言い換え規則や構文変形規則など文生成に関わる基本規則を抽出できるなど様々な自然言語処理にとって価値の高い資産となる．さらに，対応付けコーパスは応用的側面からも重要である．たとえば，要約されたテキストと元テキストとを対応付けたコーパスは，文書要約

† 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation

研究の学習データ，評価用データとして不可欠である．

このように文間の対応付けは，高度な自然言語処理を実現するうえで必須の技術である．しかし，これまでの研究は，対応元と対応先における語順の制約が強いという問題や，対応を決定するために用いる類似尺度が適切でないという問題があった．

そこで，本論文では，まず係り受け関係を正確に扱うため，文を依存構造木に基づく経路の集合であると考え，各経路に対して最も似た経路を有する文を対応文として採用する手法を提案する．また，経路を単語と意味カテゴリをからなるノードの系列データと考え，その部分列に基づき類似度を決定する．

以下，まず 2 章では単一言語での文対応付けの題材とする要約と元テキストからなるコーパスに関してそれらがどのように対応するかを TSC（Text Summarization Challenge）[9] のデータから実例をあげて観察する．3 章では，こうした要約と元テキストの対応付けに関する先行研究を紹介し，それらの問題点を指摘する．4 章では，提案手法である文の依存構造経路とそれらの類似度に基づく文対応付け手法について詳細を述べる．5 章では，実験に用いた TSC コーパスの詳細と対応付け実験の結果を報告し，考察する．6 章で本論文の結論を述べる．

2. 元テキストから要約への変換操作

人間が要約文を生成するとき，要約の元となるテキストに現れた表現を再利用することが多い[5]．Jing ら [5] は人間が行う要約の生成の過程を分析することにより，文生成に用いられる操作を以下の 6 つに分類している．

文短縮（setntence reduction） 元テキスト中の 1 文から不要語句を削除すること．

文結合（sentence combination） 元テキスト中の複数の文をまとめて 1 文とすること

構文変形（syntactic transformation） 元テキスト中の文の構文を変形すること．

語彙的言い換え（lexical paraphrasing） 元テキスト中の語句の意味的言い換えを行うこと．

抽象化/具体化（generalization/specification） 元テキスト中の節や句などを抽象化，あるいは具体化すること．

文の並びかえ（reordering） 元テキストにおける文の順序を入れ替えること．

上記の分類体系をふまえたうえで，本論文では，元テキストと要約との対応関係を以下のように分類した．TSC2 のデータから実例をあげて説明する．

文　短　縮

単語，文節，句などを削除することにより要約文を生成する操作．

元テキスト： 4 月の東京**都知事選に民主党の鳩山**邦夫**副代表が出馬**表明したのを**受けて，自民党は，**元国連事務次長の**明石康氏を擁立することになった．**

要約文： 都知事選に民主党の鳩山副代表が出馬するのを受け，自民党は明石康氏を擁立することになった．

文　結　合

元テキストにおける複数の文を結合して要約文を生成する操作．この結合の際には，多くの場合，文短縮が併用される．次の例では，2 つの文から要約文 1 文が生成されている．

元テキスト（A）：応募総数は，今回**過去最高の約 415 万編**にも及んだ．

元テキスト（B）：参加学校数は全国の小，中学校，高校，それに海外日本人学校を含め，**3 万校近くにのぼっている．**

要約文： 応募総数は，過去最高の約 415 万編で，参加学校数は 3 万校近くにのぼっている．

文　分　割

元テキストの 1 文を複数の要約文に分割する操作．次の例では 1 文から 2 つの要約文 A，B が生成されている．

元テキスト：授賞式は 12 月 10 日，ストックホルムで行われ，賞金 760 万スウェーデン・クローナ（約 1 億 2,000 万円）が贈られる．

要約文（A）： 授賞式は 12 月 10 日にストックホルムで行われる．

要約文（B）： 賞金は 760 万スエーデン・クローナ（約 1 億 1,500 万円）です．

文分割と結合の併用

元テキストの文を分割しつつ，同時に他の文との結合も行って，要約文を生成する操作．次の例では，元テキスト（A）を分割したものが要約文（A）全文と要約文（B）の一部に対応し，元テキスト（B），(C) を結合したものが要約文（B）の一部に対応する．

元テキスト（A）：まず，韓国や台湾などの競合他社から価格攻勢が強まる可能性の高い**次世代メモリー 64 メガビット DRAM（記憶保持動作が必要な随時読み出し書き込みメモリー）への新規投資を中**

止し，今後は**付加価値の高い**システム・オン・チップや次世代の 256 メガ DRAM の**開発投資にシフトし**ていく．

元テキスト（B）：また，**長期的には**半導体事業のうち**パソコン用などの汎（はん）用 DRAM** 事業の割合を減らしていく方針だ．

元テキスト（C）：今後，必要な **DRAM は台湾メーカーから供給する**方針だ．

要約（A）：次世代メモリー 64 メガビット DRAM（記憶保持動作が必要な随時読み出し書き込みメモリー）への新規投資を中止．

要約（B）：付加価値の高い開発投資にシフトし，長期的にはパソコン用などの汎用 DRAM は台湾メーカーから供給する．

3. 関連研究とその問題点

2 章で観察したとおり，要約の 1 文と元テキストの 1 文が 1 対 1 で対応するとは限らないので，要約 1 文に対して複数文の対応を許すような工夫が対応付けには必要となる．こうしたことを考慮した対応付けの研究はいくつかみられる．

Banko[1] らは，単語の集合（bag-of-words）として文を表し，要約文と元テキスト文との間の単語の重なりに応じたスコアよって対応の有無を決定する手法を提案している．文のような短いテキストに対して bag-of-words 表現を用いると，語順の制約が完全になくなるため，的確な類似度を測ることが難しくなる[11),14)]．さらに，対応の有無を決定するための最適な閾値を求めることは容易ではない．

また，Marcu[8] は，節に相当するユニット（談話構造基本単位）の集合として文を表現し，要約から得たユニット集合と元テキストから得たユニット集合の類似度が最大となるよう，元テキストから得たユニットを取捨選択する手法を提案している．ユニット間の類似度には bag-of-words によるコサイン類似度を用いている．ユニットは単語よりも意味的にまとまった単位であるが，文中の連続した単語列であるため，要約と元テキストでユニットをまたぐような語順の入れ替わりがある場合には類似度が極端に低下するという問題がある．したがって，ユニットを類似度計算の基本単位とする手法は，特に日本語のように語順に対する自由度の高い言語には向かない．

一方，Jing ら[5] は，要約中で隣接する単語対が元テキストでも近接して出現しやすいという知見を利用し，HMM と Viterbi アルゴリズムを用いて対応付けを行う手法を提案している．この手法は，単語の隣接関係に着目しているので，前の 2 つの研究よりもより語順を意識した手法といえる．しかし，隣接という強い制約があるため，要約の語順と元テキストの語順が大きく入れ替わっている場合にうまく対処できないという問題がある．さらに，複数文書要約のように似た単語列が複数出現する場合の挙動も不明である．

日本語テキストを対象とした研究としては，竹内らの研究[16] がある．彼らは，文を文節による依存構造木の経路（木の「葉」から「根」に至るまでの最短経路．本論文では以降，これを Dependency Tree Path: DTP と呼ぶ）集合として表して，要約文の DTP に対し，文節を単位として最長一致する（編集距離が最小となる）DTP を持つ文を対応文とする手法を提案している．さらに，いったん，対応文を決定した後，要約文の中に未対応の文節が連続して残っている場合には，同一の手法を未対応部分に再帰的に適用して対応文を決定することで，要約文に対して複数の元テキスト文を対応付けるように工夫している．一般的に，日本語の場合，同じ意味の文であるなら語順が変化しても依存構造は変化しない場合が多い．よって，依存構造木の経路に着目することは日本語に適した手法であるといえる．しかし，この手法には，以下の問題点が存在する．(1) 文節を単位としてマッチングをとるため，文節内で語彙の言い換えのような変化が起った場合，マッチングがとれない．(2) 未対応文節が 1 つの場合にはそこに対応付けが行われず，重要な情報が欠落する場合がある．たとえば，「1994 年の APEC がマレーシアで始まった.」という要約文に対して，「1994 年の APEC が始まった.」，「マレーシアでの開催である.」という候補文があった場合，前の文を対応付けた時点で連続する未対応文節は「マレーシアで」のただ 1 つになるので，後の文のように対応付けすべき文がある場合でもその後の対応付けが行われない．(3) 意味的にほぼ同一であり，かつ，表層上も似た単語を用いた 2 文であっても最長一致文節列が大きくなるとは限らない．たとえば，「ホテルでとった昼食は美味しかった」と「昼のホテルのランチは美味しかった」では最長一致文節列はたかだか，2 文節となり，過小評価されてしまう．

このように従来の手法には，語順に対する制約，類似度を測るための基本単位，類似尺度そのものなどに問題が存在する．

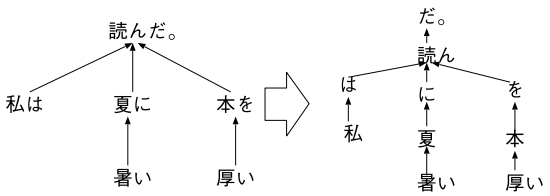

図 1 依存構造木の例
Fig. 1 An example of dependency tree.

4. 依存構造木の経路とそれらの類似度に基づく対応付け

3 章で示した従来法の問題点を解決するため，本論文では，DTP を単語とその意味カテゴリを要素とするノードの系列データとしてとらえ，DTP 間の類似度を部分属性列の一致に基づき計算し，各 DTP に対して最も類似する DTP を有する文を対応文とする手法を提案する．

要約文のすべての DTP に対し，最も類似した DTP を持つ文を対応文とすることで，複数の対応文を過不足なく割り当てることができる．また，DTP を文節列とはとらえず，単語と意味カテゴリを要素とするノード列であると見なすことで，語彙の言い換えも含みつつ柔軟に類似度を計算できる．なお，以下，4.1 節，4.2 節では説明の便宜上，DTP は単語のみから構成されるものとして説明する．

下記に各処理についての詳細を述べる．

4.1 文から DTP への分解

先述したとおり，日本語は語順が柔軟であることを考えると（意味を保持したまま）語順が変化しても依存構造は変化しない場合が多いことから，文を依存構造木として表現することは適切である．しかし，文間の対応関係は 2 章で説明したとおり，ある文の一部分に別の文の一部分が対応するといった形になっているので，依存構造木間の類似度を直接計算し，対応文を決定する手法は適していない．

つまり，文を依存構造木で表現したうえで，その部分構造に対して対応文を割り当てることが必要となる．そこで本論文では，DTP を対応付けの基本単位とする．以下に DTP の例を説明する．

文の依存構造（かかり受け構造）とは文節間の修飾，被修飾関係を表したものであり，以下の例文の依存構造木は**図 1** の左側で示される．本論文では，工藤ら[14]と同様に文節間のかかり関係を単語間のかかり関係に変換する（図 1 の右側参照）．基本的には文節内の単語は直後の単語にかかり，文節の最後の単語はかかり先の文節の主辞にかかるとする．

「私は暑い夏に厚い本を読んだ．」

図において矢印の始点の文節が終点の文節を修飾していることを表しており，文の最終文節を「根」と見なすことで，木として表現している．DTP は木の「葉」から「根」に至るまでの各経路である．よって，以下の DTP を得ることができる．

- 私 は 読ん だ
- 暑い 夏 に 読ん だ
- 厚い 本 を 読ん だ

本論文では，すべての文をこうして得た DTP を用いて対応文を決定する．

4.2 DTP を用いた文の応付け

DTP を用いた対応付け手法について説明する．2 章の文結合の例に関して，元テキスト文（$s1$，$s2$），要約文（$a1$）からは以下の DTP が得られる．

a_1-1 応募 総数 は のぼっ て いる．
a_1-2 過去 最高 の 約 415 万 編 に も のぼっ て いる．
a_1-3 参加 学校 数 は のぼっ て いる．
a_1-4 3 万 校 近く に のぼっ て いる．
s_1-1 応募 総数は 及ん だ．
s_1-2 今回 及ん だ．
s_1-3 過去 最高 の 約 415 万 編 に も 及ん だ．
s_2-1 参加 学校 数は のぼっ て いる．
s_2-2 全国 の それ に 含め のぼっ て いる．
s_2-3 小 中 学校 高校 それ に 含め のぼっ て いる．
s_2-4 海外 日本人 学校 を 含め のぼっ て いる．
s_2-5 3 万 校 近くに のぼっ て いる．

ここで，要約 a_1 の DTPa_1-i に対して最も類似する元テキスト s_j の DTPs_j-k を対応付けることを考える．いま，単純に，a_1-i と s_j-k の単語（ここでは，自立語）の重なりが大きいほど 2 つの DTP が類似していると考えると，(a_1-1,s_1-1)，(a_1-2,s_1-3)，(a_1-3,s_2-1)，(a_1-4,s_2-5) という類似した DTP の組が得られる．ここで，対応する DTP を持つ元テキストの文を対応文とすると，対応文は s_1，s_2 となる．このように DTP に分解して対応付けを行うことで，1 対多の文対応を自然に考慮できる．以下に手順の詳細を述べる．

Step 1 元テキスト（source）に含まれる文をすべて DTP の集合へと変換する☆．いま，元テキストのある文を s_j と表し，そこから得られる DTP 集合を $F(s_j)$ と表す．

Step 2 Step 3〜Step 5 を要約に含まれる文 a_i に対して適用する．

☆ 実験には，CaboCha[6] の解析結果を用いた．

Step 3 要約に含まれる文 a_i を DTP の集合へと変換する．ここで，a_i から得られる DTP 集合を $F(a_i)$ と表す．$F(a_i)$ のすべての要素に対して Step 4〜Step 5 を適用する．

Step 4 $P_{a_i} \in F(a_i)$，$P_{s_j} \in F(s_j)$ を用いて P_{a_i} と s_j の類似度を以下の式で定義する．

$$\mathrm{sim}(P_{a_i}, s_j) = \max_{P \in F(s_j)} \mathrm{sim}(P_{a_i}, P)$$

Step 5 P_{a_i} に対して $\mathrm{argmax}_{s_j \in source}\mathrm{sim}(P_{a_i}, s_j)$ となる s_j を対応文とする．

ただし，提案手法では，ある 1 つの要約文に対して最もふさわしい対応文集合を 1 つ選ぶということを目標としていることに注意されたい．これは，文生成に関わる広義の言い換え知識をより確実に獲得したいという考えから設定した目標である．なお，ある要約文に対して対応文としてふさわしい文が複数ある場合には，上記手続きをすべての要約文に対して適用し，元テキストの DTP 集合から $F(s_j)$ を取り除き，Step1〜5 を適用するということを繰り返すことによって実現できると考える．ただし，Step1〜5 を繰り返し適用する際には，$\mathrm{sim}(P_{a_i}, s_j)$ がある閾値以上のものを対象とするなどの工夫が必要であろう．

4.3 DTP 間の類似度計算法

従来より，文，文書などの 2 つのテキスト（それぞれ，o_1，o_2 とする）間の類似度には，式 (1) で定義されるコサイン類似度（距離）[4] を用いることが多い．

$$\mathrm{sim}_{\cos}(o_1, o_2) = \frac{\sum_t w_{t,o_1} w_{t,o_2}}{\sqrt{\sum_t w_{t,o_1}^2 \sum_t w_{t,o_2}^2}} \quad (1)$$

ここで，w_{t,o_1}，w_{t,o_2} は，o_1，o_2 に出現する単語 t に対する重みであり，TF，IDF，TF·IDF などが用いられる．これは，テキストに出現する個々の単語を独立に評価して類似度を計算しているといえる．しかし，重みを考慮したとしても単純に個々の単語の重なりをみるだけでは，単語の並びを考慮していないので，テキスト間の類似性をとらえきることはできない[10),15)]．このような問題を背景として近年では，単語の組合せ（共起）を考慮してテキスト間の類似度を計算する手法が提案されている．たとえば，機械翻訳結果の自動評価には n-gram に着目した手法[10)]，テキスト分類には，テキストを文字の系列データと解釈してその系列パターン（部分文字列）に着目したストリング・カーネル（String Subsequence Kernel，以下，SSK）[7)] や SSK を拡張したワード・シーケンス・カーネル（Word Sequence Kernel，以下，WSK）[2)] を用いた手法が提案されている．

本論文では，上記 WSK を改良した拡張ストリング・カーネル（Extended String Subsequence Kernel，以下，ESK）を用いて DTP の類似度計算に用いる．以下に，カーネル関数について説明し，SSK，WSK について説明する．最後に ESK の詳細を述べる．

カーネル関数（K）は，2 つの対象 $\mathbf{x}$，$\mathbf{x}'$ に対し，それらをある関数 ϕ で写像した空間においての対象間の内積計算として定義される[12)]．つまり，$\phi(\mathbf{x})\cdot\phi(\mathbf{x}') = K(\mathbf{x}, \mathbf{x}')$ となる．ただし，K は，対象を陽に ϕ で写像することなくその値を効率的に計算できる．K が ϕ で写像した空間での対象間の内積であることを考えると，$\mathbf{x}$ と $\mathbf{x}'$ の類似度を表していると考えることができる[12)]．

SSK は，入力対象をテキストとし，ϕ がその部分記号列を基底とする空間へ写像することに相当するカーネル関数である[7)]．このとき，基底に対する座標値は，着目した部分記号列の重み付き総和であり，重みはスキップも含めた部分記号列長 ℓ に応じて減衰パラメータを用い，λ^ℓ で与えられる．特別な場合として λ を 1 とすると，2 つの記号列間の SSK のカーネルの値は，両者に共通に含まれる部分記号列の数の和に相当する．たとえば，「abaca」と「abbab」という記号列に対して，3 個の組合せ（以降，組み合わせる記号の数を d と表す）からなる部分記号列を考えると，これらの記号列に共通で，かつ，記号のスキップを許したうえで 3 つの記号の組合せ（$d = 3$）からなる部分記号列は「aba」だけであり，それぞれに 2 回出現☆している．よって $d = 3$ のとき，カーネルの値は 4 となる．ここで，組合せ数 d だけの部分記号列のみを計算の対象としていること，スキップも含めた部分記号列の長さ ℓ に依存して λ^ℓ が部分記号列の重みとして与えられることを考えると，d が大きい場合や部分記号列が長い場合にはカーネルの値が小さな値をとる傾向になるという問題がある．

そこで WSK では，以下の拡張を SSK に対して施している☆☆．

- d 個以下のスキップを許したすべての部分記号列の組合せを考慮する．
- スキップを含めた部分記号列長 ℓ ではなく，スキップした記号の数 $\ell - d$ に応じて $\lambda^{\ell-d}$ で部分記号列の重みを与える．

これは，i 個の記号組合せによる SSK を $K_{\mathrm{ssk}(i)}$ として表すと，$K_{\mathrm{wsk}(d)} = \sum_{i=1}^{d} K_{\mathrm{ssk}(i)}/\lambda^i$ であることを示している．

☆ 「abaca」には，「a-b-a」と「a-b-*-*-a」，「abbab」には，「a-b-*-a」と「a-*-b-a」がある．「*」は文字のスキップを表す．

☆☆ これ以外の拡張もあるが，本論文では扱わない．

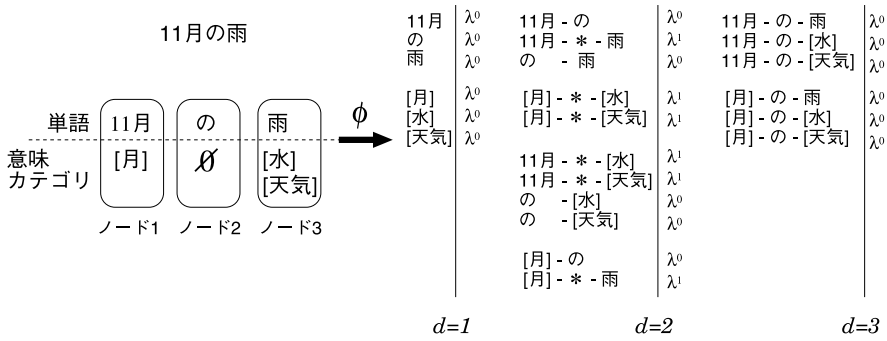

図 2 ノード列を入力とする拡張ストリング・カーネル
Fig. 2 Extended String Subsequence Kernel.

表 1 「abaca」と「abbab」の部分記号列とその値
Table 1 Components of vectors corresponding to 'abaca' and 'abbab.' Bold letters indicate common subsequences.

subsequence	abaca	abbab
abb	0	$1+2\lambda^2$
aba	$1+\lambda^2$	2λ
abc	λ	0
aab	0	λ^2
aac	λ	0
aaa	λ^2	0
aca	λ^2+1	0
ab	1	$2+\lambda+\lambda^3$
aa	$2\lambda+\lambda^3$	λ^2
ac	$1+\lambda^2$	0
ba	$1+\lambda^2$	$1+\lambda$
bb	0	$1+\lambda+\lambda^2$
bc	λ	0
ca	1	0
a	3	2
b	1	3
c	1	0

ここで，対象「abaca」と「abbab」を入力として K_{wsk}('abaca','abbab') の値を計算する．両者を ϕ で写像すると最大 3 個までの部分記号列とその重みは **表 1** となる．なお，共通する部分記号列を太字で表している．よって，$K_{\mathrm{wsk}(3)}$ は以下の式で求まる．

$$K_{\mathrm{wsk}(3)}(\text{'abaca','abbab'}) = \overbrace{(1+\lambda^2)\times 2\lambda}^{aba} + \overbrace{1\times(2+\lambda+\lambda^3)}^{ab} + \overbrace{(2\lambda+\lambda^3)\times\lambda^2}^{aa} + \overbrace{(1+\lambda^2)\times(1+\lambda)}^{ba} + \overbrace{3\times 2}^{a} + \overbrace{1\times 3}^{b}. \quad (2)$$

しかし，WSK は，表層での単語マッチングしか行わないので，元テキストを言い換えて要約を生成した場合にはうまくマッチングがとれない．そこで本論文では，単語の意味属性も考慮してマッチングをとるため，WSK に対して「記号」を単語とその意味カテゴリを要素とする「ノード」として扱えるように改良を加えたカーネル（ESK）を提案する．ESK を用いることで，言い換えを吸収したマッチングが可能となる．

図 2 の例で説明する．「11 月の雨」というテキストはそれぞれの単語と意味カテゴリからなるノードとして表される☆．1 つの単語に対して複数の意味カテゴリが割り当てられていれば，ノードは単語と複数の意味カテゴリから構成される．ESK における写像関数 ϕ は，テキストを単語または意味カテゴリの部分要素列，単語と意味カテゴリの双方を含む部分要素列を基底とする空間へと写像する．$d=3$ とし，「11 月の雨」というテキストを ϕ で写像すると，部分要素列とその重みは図 2 のとおりとなり，その空間で内積を計算したものがカーネルの値となる．

厳密には，$K_{\mathrm{esk}}(T,U)$ は以下の式で定義される．ただし，T，U はノード列を表す．

$$K_{\mathrm{esk}}(T,U) = \sum_{m=1}^{d} \sum_{t_i \in T} \sum_{u_j \in U} K_m(t_i,u_j). \quad (3)$$

$$K_m(t_i,u_j) = \begin{cases} val(t_i,u_j) & \text{if} \quad m=1 \quad \text{otherwise} \\ K'_{m-1}(t_i,u_j)\cdot val(t_i,u_j) \end{cases} \quad (4)$$

$K'_m(t_i,u_j)$ は以下の式で定義される．ただし，t_i，u_j は T，U のノードを表す．$val(t_i,u_j)$ は，ノード間で一致した要素数を返す関数である．

☆ ノードの要素は自由に定義でき，単語とその品詞や固有表現カテゴリなどを用いることも可能である．

表 3 要約と抜粋の文字数と文数
Table 3 Description of abstracts and extracts.

			抜粋		要約		抜粋/要約
A1	文数	Short	268	(0.304)	217	(0.246)	1.24
		Long	462	(0.524)	366	(0.415)	1.26
	文字数	Short	11,048	(0.324)	6964	(0.201)	1.59
		Long	19,152	(0.561)	13,639	(0.400)	1.40
A2	文数	Short	258	(0.293)	206	(0.234)	1.25
		Long	442	(0.502)	366	(0.415)	1.21
	文字数	Short	11,337	(0.332)	6,844	(0.201)	1.66
		Long	18,762	(0.550)	13,842	(0.406)	1.36
A3	文数	Short	272	(0.309)	234	(0.265)	1.16
		Long	452	(0.513)	403	(0.457)	1.12
	文字数	Short	12,171	(0.357)	6,732	(0.197)	1.81
		Long	19,532	(0.573)	13,396	(0.393)	1.46
B1	文数	Short	428	(0.176)	354	(0.146)	1.21
		Long	710	(0.293)	610	(0.251)	1.16
	文字数	Short	26,834	(0.241)	10,707	(0.096)	2.50
		Long	41,189	(0.370)	21,210	(0.190)	1.94
B2	文数	Short	333	(0.137)	234	(0.096)	1.42
		Long	615	(0.254)	447	(0.184)	1.38
	文字数	Short	21,417	(0.192)	10,580	(0.095)	2.02
		Long	36,866	(0.331)	21,344	(0.191)	1.73
B3	文数	Short	371	(0.153)	271	(0.112)	1.37
		Long	630	(0.260)	492	(0.203)	1.28
	文字数	Short	23,504	(0.211)	10,589	(0.095)	2.22
		Long	38,121	(0.342)	21,478	(0.222)	1.77

表 2 評価実験に用いた TSC データの詳細
Table 2 Description of TSC data.

	単一文書	複数文書
文書セット	–	30
文書数	30	224
文数	881	2,425
文字数	34,112	111,472

$$K'_m(t_i,u_j)=\begin{cases}0 \quad \text{if} \quad j=1 \quad \text{otherwise}\\ \lambda K'_m(t_i,u_{j-1})+K''_m(t_i,u_{j-1})\end{cases} \tag{5}$$

さらに，$K''_m(t_i,u_j)$ は以下の式で定義される．

$$K''_m(t_i,u_j)=\begin{cases}0 \quad \text{if} \quad i=1 \quad \text{otherwise}\\ \lambda K''_m(t_{i-1},u_j)+K_m(t_{i-1},u_j)\end{cases} \tag{6}$$

なお，類似度としてカーネルの値を用いる場合には以下の正規化を施した式を用いる．

$$\mathrm{Sim}_{\mathrm{esk}}(T,U)=\frac{K_{\mathrm{esk}}(T,U)}{\sqrt{K_{\mathrm{esk}}(T,T)K_{\mathrm{esk}}(U,U)}} \tag{7}$$

5. 評価実験と考察

5.1 評価に用いたコーパスの分析

評価実験には，TSC2 で作成された要約データを用いた．TSC2 では，単一文書要約，複数文書要約の 2 つのタスクが設定されている．各タスクで用いられたテストセットの詳細を**表 2** に示す．それぞれ，異なる 3 名の要約筆記者が short，long の 2 種の字数制限に従って要約を作成している．ここでは，単一文書要約のデータを A1，A2，A3 とよび，複数文書要約のデータを B1，B2，B3 とよぶこととする．これら，A1〜A3，B1〜B3 に対して被験者 1 名が元テキストとの対応付けを行った．なお，4.2 節で説明したたとおり，本論文ではある 1 つの要約文に対して最適な対応文集合を 1 つ選択することを目標とするので，正解データを作成する際には，対応文としてふさわしい文が他にあった場合でもその中のただ 1 つしか，正解データとして対応付けていない．ただし，今回のデータにおいて複数の正解対応文を持つような例が全体に占める割合は，2%程度であり，実験結果への影響は小さい．

要約と元テキストを対応付けることで作成されたデータ (以降，これを抜粋と呼ぶ) に関して，文数，文字数，要約に対する長さ比の調査結果を**表 3** に示す．なお，カッコ内の数値は要約率（要約の長さ/元テキストの長さ ×100）を表す．ただし，要約における「文」と抜粋における「文」が異なる場合があることに注意されたい．

まず，単一文書要約タスクのデータからみる．short

の場合，文を単位とした要約率は，抜粋が約 30%，要約が 25%であり，抜粋は要約の 1.2 倍程度の長さである．文字を単位とした要約率では，抜粋が 33%，要約は 20%であり，抜粋は要約の 1.6〜1.8 倍の長さである．long の場合，文を単位とした要約率は，抜粋が約 50〜52%，要約が約 42〜46%であり，抜粋は要約の 1.1〜1.26 倍程度の長さである．文字を単位とした要約率では，抜粋が約 55〜57%，要約は 40%であり，抜粋は要約の約 1.4 倍前後の長さである．

次に，複数文書要約タスクのデータをみる．short の場合，文を単位とした要約率は，抜粋が要約が約 10〜14%，要約が約 14〜18%で，抜粋は要約の 1.2〜1.4 倍程度の長さである．文字を単位とした要約率では，抜粋が約 19〜24%，要約が約 9.5%で，抜粋は要約の 2〜2.5 倍の長さである．long の場合，文を単位とした要約率は，抜粋が要約が約 25〜29%，要約が約 18〜25%で，抜粋は要約の 1.2〜1.4 倍程度の長さである．文字を単位とした要約率では，抜粋が約 33〜37%，要約が約 19〜22%で，抜粋は要約の 1.7〜1.9 倍の長さである．

タスクの違いによる特徴としては，複数文書要約タスクのデータの方が要約率が低い．特に short では顕著である．これは，タスクで設定された字数の制限が厳しいためであり，より多くの情報を含むよう 2 章で説明した文短縮，文結合，それらの併用を多用して要約を作成するからであると考える．

また各データについて，要約の 1 文に対して元テキストの何文が対応付けられたかを調べた結果を**表 4** に示す．カッコ内の数値はそれぞれの対応関係の全体に占める割合を表している．双方のデータ共に 1 対 1 の対応関係が多くを占めているが，複数文書要約のデータの方が 1 対 2 以上の対応付けが多い．次に，元テキストの 1 文が要約の何文に対応付くかを調べた結果を**表 5** に示す．単一文書要約データでは，ほぼ，元テキストの 1 文が要約文でも 1 文に対応しており，文分割が行われていないことを示す．これに対して複数文書要約データでは全体に占める割合は少ないが，元テキストの 1 文が要約では 2 文に対応する例や 3 文に対応する例がみられる．

表 4 要約文の 1 文に対応する元テキスト文数の分布
Table 4 The distribution of aligned original sentences corresponding to one summary sentence.

		1	2	3 以上
A1	Short	167 / (0.770)	49 / (0.226)	1 / (0.005)
	Long	283 / (0.773)	73 / (0.199)	10 / (0.027)
A2	Short	157 / (0.762)	46 / (0.223)	3 / (0.015)
	Long	299 / (0.817)	59 / (0.161)	11 / (0.022)
A3	Short	198 / (0.846)	34 / (0.145)	2 / (0.009)
	Long	359 / (0.890)	39 / (0.097)	5 / (0.012)
B1	Short	295 / (0.833)	45 / (0.127)	14 / (0.040)
	Long	530 / (0.869)	65 / (0.107)	15 / (0.025)
B2	Short	156 / (0.667)	58 / (0.248)	20 / (0.085)
	Long	312 / (0.698)	104 / (0.233)	31 / (0.069)
B3	Short	191 / (0.705)	62 / (0.229)	18 / (0.066)
	Long	392 / (0.797)	76 / (0.154)	24 / (0.048)

表 5 元テキストの 1 文に対応する要約文数の分布
Table 5 The distribution of aligned summary sentences corresponding to one source sentence.

		1	2	3 以上
A1	Short	268 / (1.000)	0	0
	Long	458 / (0.994)	2 / (0.006)	0
A2	Short	258 / (1.000)	0	0
	Long	440 / (1.000)	0	0
A3	Short	272 / (1.000)	0	0
	Long	450 / (1.000)	0	0
B1	Short	406 / (0.974)	11 / (0.026)	0
	Long	660 / (0.964)	22 / (0.032)	2 / (0.004)
B2	Short	317 / (0.975)	8 / (0.025)	0
	Long	550 / (0.945)	31 / (0.053)	1 / (0.002)
B3	Short	364 / (0.989)	4 / (0.011)	0
	Long	583 / (0.965)	16 / (0.025)	5 / (0.010)

5.2 実験の設定

提案手法の有効性を検証するため，前節で述べたコーパスを用いて評価実験を行った．DTP を用いる場合と用いない場合（ベースライン）の双方に対して以下に示す類似尺度を組み合わせ，比較した．なお，ベースラインの場合には，類似度がある閾値を超えた文を対応文として採用した．閾値は，0.05〜1 まで 0.05 刻みで変化させた．

BOW テキストを単語集合としてとらえ，コサイン距離（式 (1)）を用いて類似度を計算する．対象単語は自立語のみとし，w_t には TF を用いた．

N-gram BOW の拡張として，素性に bigram を加えた場合（2-gram），さらに trigram を加えた場合（3-gram）のコサイン距離を計算する．

TREE 文を文節に基づく依存構造木としてとらえ，TreeKernel [3] を用いて文間の類似度を計算する．

WSK テキストを単語列としてとらえ，WSK を用いて類似度を計算する．単語の組合せ数 d は，2〜4 まで変化させた．

ESK テキストを単語と意味カテゴリからなるノード列としてとらえ，ESK を用いて類似度を計算する．意味カテゴリは日本語語彙大系[13] に従った．d は WSK と同一条件で変化させた．

なお，TREE はベースラインのみで用いた．WSK，ESK の減衰パラ―メタ λ に関しては，Leave-one-out 法で最適値を決定し，評価を行った．

評価指標には F 値を用いた．F 値は，システムが出力した対応文のうち正解であった対応文の数を a としたとき，以下の適合率（P）と再現率（R）の調和

表 6 ベースラインでの対応付けの評価結果（単一文書要約データ）
Table 6 Evaluation results w/o DTP (single documents).

		ESK			WSK			BOW	2-gram	3-gram	TREE
		(d=2)	(d=3)	(d=4)	(d=2)	(d=3)	(d=4)				
Short	A1	0.951	0.941	0.907	0.957	**0.958**	0.955	0.906	0.952	0.948	0.386
	A2	0.938	0.928	0.901	0.943	0.948	**0.954**	0.916	0.945	0.950	0.322
	A3	0.927	0.922	0.906	0.949	0.944	**0.951**	0.875	0.926	0.926	0.436
Long	A1	0.951	0.949	0.935	0.959	0.956	0.959	0.916	**0.961**	0.959	0.418
	A2	0.961	0.968	0.958	**0.973**	0.972	0.972	0.940	0.966	0.972	0.476
	A3	0.961	**0.967**	0.966	0.961	0.964	0.966	0.926	0.961	0.962	0.547

表 7 DTP を用いた場合の対応付けの評価結果（単一文書要約データ）
Table 7 Evaluation results with DTP (single documents).

		DTP(ESK)			DTP(WSK)			DTP	DTP	DTP
		(d=2)	(d=3)	(d=4)	(d=2)	(d=3)	(d=4)	(BOW)	(2-gram)	(3-gram)
Short	A1	**0.966**	0.960	0.958	0.957	0.950	0.946	0.955	0.952	0.952
	A2	**0.973**	0.970	0.966	0.955	0.955	0.943	0.959	0.957	0.956
	A3	0.952	0.955	0.955	0.960	0.961	0.960	**0.964**	0.962	0.960
Long	A1	0.947	0.951	0.956	0.956	0.955	0.950	**0.960**	0.951	0.949
	A2	0.975	0.975	**0.976**	0.974	0.973	0.958	0.972	0.973	0.975
	A3	0.965	0.965	0.961	0.968	**0.969**	0.966	0.962	0.960	0.960

表 8 DTP を用いたことによる対応付け性能の改善幅（単一文書要約データ）
Table 8 Effectiveness of DTP (single documents).

		ESK			WSK			BOW	2-gram	3-gram
		(d=2)	(d=3)	(d=4)	(d=2)	(d=3)	(d=4)			
Short	A1	+0.015	+0.019	+0.051	±0	−0.008	−0.009	+0.049	±0	+0.008
	A2	+0.035	+0.042	+0.065	+0.012	+0.007	−0.011	+0.043	+0.012	+0.006
	A3	+0.025	+0.033	+0.049	+0.011	+0.017	+0.009	+0.089	+0.036	+0.034
Long	A1	−0.004	+0.002	+0.021	−0.003	−0.001	−0.009	+0.044	−0.010	−0.010
	A2	+0.014	+0.007	+0.018	+0.001	+0.001	−0.014	+0.032	+0.007	+0.003
	A3	+0.004	−0.002	−0.005	+0.007	+0.005	±0	+0.036	−0.001	−0.002

平均（$((1+\beta^2)/(1/\mathrm{P}+\beta^2/\mathrm{R})$）で表される．ただし，$\beta=1$ として計算した．

適合率（P）$= a/$システムが出力した対応文の数

再現率（R）$= a/$正解の対応文の数

5.3 単一文書要約データでの評価結果

表 6 にベースラインでの評価結果，**表 7** に DTP を用いた場合の評価結果を示す．また，**表 8** には，DTP を用いたことによる改善幅を示す．

表 6 より，文を基本単位として類似度を計算する場合には，単語の系列パターンと見なし，WSK を用いることが有効であることが分かる．単語の組合せ数 d への依存は小さく，どの d においても高い性能である．続いて，ESK，2-gram，3-gram が同程度の成績で，BOW はやや劣り，TREE は非常に悪い成績である．ESK の場合，short では，$d=2$ の成績が最も良く，d を増やすに従い，性能が劣化している．long では，$d=3$ の場合が最も良い．ESK は単語とその意味ラベルを素性として類似度を計算するので，d を増やすと素性数が膨大になり，このような性能劣化が起こると考える．ただし，BOW よりはおおむね良い成績であることを考えると，文などの短いテキストを対象としたときには，語順を考慮できる類似尺度が適していることが分かる．また，TREE の成績が悪い原因は，すべての部分木を考慮してカーネルの値を計算することにある．部分構造に対して制約を設けずカーネルの値を計算すると素性数が膨大になり，性能の低下を招くことは Convolution Kernel の欠点としてよく知られている[11]．次に，表 7 をみると，DTP（ESK）が他よりも若干良い成績であり，DTP（WSK）と同様，d に依存せず高い性能である．よって，DTP を用いる場合，それを単なる単語列として扱うよりも，単語と意味ラベルからなるノード列として扱う方が良いことが分かる．

また，表 8 より，short の場合には DTP（WSK）（$d=3,4$）を除くすべての手法で性能が改善されていることが分かる．long の場合には，やや性能が低下する場合もみられるが，多くの場合，DTP を用いることで性能が改善されていることが分かる．BOW の場合，その傾向は特に顕著である．すなわち，テキスト間の類似尺度に何を用いるのであれ，文を DTP

表 9 ベースラインでの対応付けの評価結果（複数文書要約データ）
Table 9 Evaluation results w/o DTP (multiple documents).

		ESK			WSK			BOW	2-gram	3-gram	TREE
		(d=2)	(d=3)	(d=4)	(d=2)	(d=3)	(d=4)				
Short	B1	0.609	0.580	0.520	0.547	0.504	0.402	0.576	**0.644**	0.638	0.127
	B2	0.608	0.622	0.616	0.660	0.660	0.622	0.590	0.668	**0.680**	0.161
	B3	0.674	0.683	0.667	0.712	0.697	0.633	0.654	**0.733**	0.729	0.158
Long	B1	0.674	0.670	0.602	0.627	0.588	0.485	0.655	**0.714**	0.711	0.223
	B2	0.717	0.742	0.740	**0.769**	0.767	0.740	0.690	0.751	0.761	0.236
	B3	0.775	0.793	0.793	**0.821**	0.820	0.803	0.768	0.805	0.817	0.280

表 10 DTP を用いた場合の対応付けの評価結果（複数文書要約データ）
Table 10 Evaluation results with DTP (multiple documents).

		DTP(ESK)			DTP(WSK)			DTP	DTP	DTP
		(d=2)	(d=3)	(d=4)	(d=2)	(d=3)	(d=4)	(BOW)	(2-gram)	(3-gram)
Short	B1	**0.741**	0.731	0.695	0.725	0.714	0.719	0.719	0.725	0.728
	B2	0.721	0.697	0.697	**0.735**	0.725	0.733	0.710	0.720	0.721
	B3	0.776	**0.786**	0.766	0.776	0.782	0.776	0.748	0.768	0.760
Long	B1	0.793	0.788	0.795	0.792	0.789	0.787	0.784	**0.797**	**0.797**
	B2	**0.805**	0.801	0.783	0.794	0.792	0.798	0.797	0.797	0.794
	B3	0.838	0.817	0.838	**0.848**	0.840	0.838	0.828	0.835	0.830

表 11 DTP を用いたことによる対応付け性能の改善幅（複数文書要約データ）
Table 11 Effectiveness of DTP (multiple documents).

		ESK			WSK			BOW	2-gram	3-gram
		(d=2)	(d=3)	(d=4)	(d=2)	(d=3)	(d=4)			
Short	B1	+0.132	+0.151	+0.175	+0.178	+0.210	+0.317	+0.143	+0.081	+0.096
	B2	+0.113	+0.075	+0.081	+0.075	+0.065	+0.111	+0.120	+0.052	+0.041
	B3	+0.102	+0.103	+0.099	+0.064	+0.085	+0.143	+0.094	+0.035	+0.031
Long	B1	+0.119	+0.118	+0.193	+0.165	+0.201	+0.302	+0.129	+0.083	+0.086
	B2	+0.088	+0.059	+0.043	+0.025	+0.025	+0.058	+0.107	+0.046	+0.033
	B3	+0.063	+0.024	+0.045	+0.027	+0.020	+0.035	+0.060	+0.030	+0.013

集合としてとらえ，各 DTP に最も類似する DTP を持つ文を対応文とすることは有効であることを示している．ただし，単一文書要約データでは，DTP を用いたことによる改善幅は最大でも 9 ポイント弱にとどまっている．

5.4 複数文書要約データでの評価結果

表 9 にベースラインでの評価結果，**表 10** に DTP を用いた場合の評価結果を示す．また，**表 11** には，DTP を用いたことによる改善幅を示す．

表 9 より，全体的に表 6 よりも成績が低いことが分かる．short の場合には N-gram の成績が良く，次いで $d = 2$ の WSK，ESK の順で成績が良い．$d = 3, 4$ の ESK，WSK ともデータによっては，BOW よりも成績が低い場合がある．long の場合，データ B1 に関しては N-gram の成績が良いが，それ以外では，$d = 2$ の WSK の成績が良い．また，データ B1 を除けば，どのような d に対しても，ESK，WSK ともに BOW よりも成績が良い．単一文書要約データの場合と同じく short，long とも TREE の成績は非常に悪い．このことから，組み合わせる単語数を適切に選択できれば，単語の並びを意識した手法が有効であることが分かる．

次に，表 10 をみると，short，long の双方において，$d = 2, 3$ の DTP（ESK）がやや成績が良い．次いで $d = 2, 3$ の WSK，N-gram の順で成績が良い．また，$d = 4$ の DTP（ESK），DTP（WSK），DTP（BOW）はほぼ同等の成績である．表 11 によれば，N-gram を用いた場合の改善幅がやや小さいが，その他の手法では，short では 10 ポイント以上改善されている．単一文書要約データの結果である表 8 と比較すると，複数文書要約データでは改善幅が大きく，DTP を導入した効果がよく分かる．

単一文書要約データと比較して改善幅が大きい理由は，複数文書要約では，文短縮，結合，分割が多用されるので，元テキストの 1 文がそのまま利用されることが少ない．このため，文をその部分構造である DTP へ分解しなければうまく対応をとれないことにあると考える．

また，要約率による成績の違いをみると，単一文書要約のデータでは顕著な差はないが，複数文書要約の

データでは高い要約率の方が成績が良い．単一文書要約のデータでは低い要約率でも高い要約率でも元テキストの表現がそのまま用いられることが多い☆のに対して，複数文書要約のデータでは高い要約率の方が元テキストの表現がそのまま使われる箇所が多いことが関係している．

表 10 をみると，DTP を用いることによって，ESK は WSK よりも成績が良くなる傾向にある．これは，文を DTP という短いテキストに分解したことによって，語の意味ラベルがより有効に働いたことを示唆している．

5.5 パラメータに関する考察

ここでは，カーネルの 2 つのパラメータ（d と λ）について，DTP（ESK）と DTP（WSK）の F-measure に対してどのような影響を及ぼすかを議論する．

図 3 に DTP（ESK），DTP（WSK）のそれぞれについて，d を 2〜4，λ を 0.05〜1 まで変化させた際の F-measure の変化を示す．DTP（ESK）は，$d = 2$ の場合，$0.15 \leq \lambda \leq 0.9$ の区間では F-measure は，比較的安定しており，高い成績であるが，λ が 0.15 未満，0.95 以上の場合には大きく成績が落ちる．$d = 3$ でも，それはほぼ同様の傾向にあるが，安定性は $d = 2$ の場合より劣る．$d = 4$ では，F-measure の安定区間は狭まり，全体的にも $d = 2, 3$ と比較して成績が悪い．

一方，DTP（WSK）では，λ を大きくすると，$d = 2$ については，F-measure が減少する傾向にあり，$d = 3, 4$ については，$\lambda = 0.2, 0.25$ 付近でピークを持ちその後減少する傾向にある．$d = 3$ の場合が，他の場合よりもやや成績が良い．DTP（ESK）と比較すると全体的に成績は悪い．

上記より，DTP（ESK）（特に $d = 2, 3$ の場合）は DTP（WSK）と比較して，λ の影響が小さく，高い成績であることが分かる．よって，組合せ数 d を 2 か 3 に設定した ESK は，DTP 間の類似尺度として WSK より適していると考える．

6. おわりに

本論文では，同じ内容を表す異なるテキストが与えられた際に文間の対応付けを行う手法を提案した．提案手法は，文を依存構造木における経路集合ととらえ，各経路に最も類似する経路を有する文を対応文とすること，経路を単語と意味カテゴリを要素として持つノード列として表現し，ESK を用いて類似度を計算

☆ 単一文書要約の作成の際には，文書から重要箇所を抜き出し，それらを用いて要約を作成するというガイドラインが設けられている．

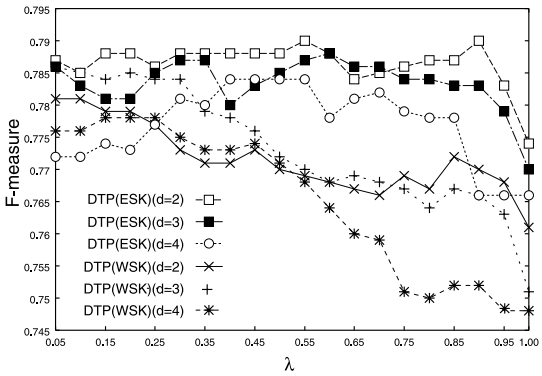

図 3 λ を変化させた場合の F 値の変化（B3-short）
Fig. 3 F-measure on various λ (B3-short).

することを特徴とする．TSC の単一文書要約データ，複数文書要約データに対し人間が対応付けを行った結果を正解として提案手法を評価した結果，単一文書要約のデータに関しては F 値 0.95〜0.97，複数文要約データに関しては F 値 0.72〜0.83 という高い性能を示し，従来の対応付け手法より優れていることを確認した．

謝辞 Cabocha を公開してくださっている工藤拓氏に感謝いたします．また，NTCIR，TSC の運営に関わられたすべての皆様に感謝いたします．

参考文献

1) Banko, M., Mittal, V., Kantrowitz, M. and Goldstein, J.: Generating Extraction-Based Summaries from Hand-Written Summaries by Aligning Text Spans, *Proc. 4th Conference of the Pacific Association for Computational Linguistics* (1999).
2) Cancedda, N., Gaussier, E., Goutte, C. and Renders, J.-M.: Word-Sequence Kernels, *Journal of Machine Learning Research*, Vol.3, No.Feb, pp.1059–1082 (2003).
3) Collins, M. and Duffy, N.: Convolution Kernels for Natural Language, *Proc. Neural Information Processing Systems* (*NIPS'2001*) (2001).
4) Hearst, M.-A.: Texttiling: Segmenting Text into Multi-paragraph Subtopic Passages, *Computational Linguistics*, Vol.23, No.1, pp.33–64 (1997).
5) Jing, H. and McKeown, K.: The Decomposition of Human-Written Summary Sentences, *Proc. 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp.129–136 (1999).
6) Kudo, T. and Matsumoto, Y.: Japanese De-

pendency Analysis using Cascaded Chunking, *Proc. 6th Conference on Natural Language Learning*, pp.63–69 (2002).
7) Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C.: Text Classification using String Kernel, *Journal of Machine Learning Research*, Vol.2, No.Feb, pp.419–444 (2002).
8) Marcu, D.: The Automatic Construction of Large-scale Corpora for Summarization Research, *Proc. 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp.137–144 (1999).
9) Okumura, M., Fukusima, T. and Nanba, H.: Text Summarization Challenge 2 - Text summarization evaluation at NTCIR Workshop 3, *HLT-NAACL 2003 Workshop: Text Summarization* (*DUC03*), pp.49–56 (2003).
10) Papineni, S., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pp.62–66 (2002).
11) Suzuki, J., Hirao, T., Sasaki, Y. and Maeda, E.: Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data, *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, pp.32–39 (2003).
12) 麻生英樹，津田　宏，村田　昇：パターン認識と学習の統計学，岩波書店 (2003).
13) 池原　悟，宮崎正弘，白井　諭，横尾昭男，中岩浩巳，小倉健太郎，大山芳史，林良　彦：日本語語彙大系，岩波書店 (1999).
14) 工藤　拓，松本裕治：部分木を素性とする Decision Stumps と Boosting Algorithm の適用，情報処理学会研究会（SIG-NL-158），pp.55–62 (2003).
15) 平尾　努，賀沢秀人，磯崎秀樹，前田英作，松本裕治：機械学習による複数文書からの重要文抽出，自然言語処理，Vol.10, No.1, pp.81–108 (2003).
16) 竹内和広，松本裕治：自動文節対応付けを用いた要約中の文再構成操作の調査，自然言語処理，Vol.9, No.3, pp.87–108 (2002).

(平成 16 年 12 月 6 日受付)
(平成 17 年 9 月 2 日採録)

平尾　努（正会員）

1995 年関西大学工学部電気工学科卒業．1997 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同年 NTT データ通信株式会社（現，株式会社 NTT データ）入社．2000 年より日本電信電話株式会社 NTT コミュニケーション科学基礎研究所に所属．2002 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．自然言語処理の研究に従事．言語処理学会，ACL 各会員．

鈴木　潤（正会員）

1999 年慶應義塾大学理工学部数理科学科卒業．2001 年同大学院理工学研究科計算機科学専攻修士課程修了．同年日本電信電話株式会社入社．2005 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．現在，NTT コミュニケーション科学基礎研究所に所属．主として自然言語処理，機械学習に関する研究に従事．ACL，言語処理学会各会員．

磯崎　秀樹（正会員）

1983 年東京大学工学部計数工学科卒業．1986 年同工学系大学院修士課程修了．同年日本電信電話（株）入社．1990〜1991 年スタンフォード大学ロボティクス研究所客員研究員．現在，NTT コミュニケーション科学基礎研究所知識処理研究グループリーダ．博士（工学）．平成 15 年度情報処理学会論文賞・山下記念研究賞受賞．人工知能・自然言語処理の研究に従事．電子情報通信学会，人工知能学会，言語処理学会，ACL 各会員．

前田 英作（正会員）

1984年東京大学理学部卒業．1986年同大学院修士課程理学系研究科修了．同年日本電信電話株式会社入社．現在，NTTコミュニケーション科学基礎研究所．メディア情報研究部アンビエントインテリジェンス研究グループリーダ，知能統合オープンラボ統括．工学博士．1995〜1996年ケンブリッジ大学（英国）客員研究員．パターン認識，統計的機械学習，自然言語処理，生物情報処理，環境知能などの研究に従事．IEEE，電子情報通信学会，日本バイオインフォマティックス学会各会員．