

潜在意味を捉え制約付き差分進化を用いた組合せ最適化による複数文書要約

Topic-based Multi-document Summarization using Differential Evolution for Combinatory Optimization of Sentences

重松 遥 小林 一郎
Haruka Shigematsu Ichiro Kobayashi

お茶の水女子大学大学院人間文化創成科学研究科理学専攻

Advanced Sciences, Faculty of Sciences, Graduate School of Humanities and Sciences, Ochanomizu University

As a general method for multi-document summarization, there is a method to obtain the most proper combination of important sentences in the documents, avoiding redundancy in a generated summary. The exact algorithms, i.e., integer programming, for optimal combination are often used for proper combination of sentences. They have however a problem in terms of calculation costs. On the other hand, as an optimization method to obtain quasi optimum solution in real time, it is reported that evolutionary computation is useful. In this context, we employ differential evolution (DE) known as superior to other evolutionary computation algorithms in terms of calculation costs and the accuracy of solution, and apply it to multi-document summarization. Besides, under an assumption that multiple topics are included in a document, latent topics in documents are extracted by means of latent Dirichlet allocation, and a summary is generated by considering the latent topics.

1. はじめに

近年、大量の文書データと接する機会の増加にともない、文書要約技術の必要性が高まっている。文書要約の一手法としては、要約生成問題を文の組合せ最適化問題として帰着させる方法がある。最適化手法としては、動的計画法や分岐限定法などの厳密解法を用いた研究が多い。しかし、厳密解法には、要約対象とする文書集合の大きさに従って、計算時間が膨大に膨れ上がってしまうという問題が存在する。一方、厳密解を追求せず実用的な時間で近似解を求める最適化手法として、進化的アルゴリズムの有効性が報告されている。そのような背景を踏まえて、本研究では、進化的アルゴリズムの中でも解の精度や計算時間の点で優れているとされている差分進化アルゴリズムを用いて組合せ最適化を行う要約文生成を行う。また、文書中には複数のトピックが含まれているという仮定の下に、文書内の潜在トピックを潜在的ディリクレ配分法を用いて抽出し、各トピックの内容を万遍なく含むような文の組合せを要約文として生成する。

2. 関連研究

組合せ最適化に基づく文書要約手法において、最適化手法に分岐限定法や動的計画法などの厳密解法を用いることが多い[1, 2, 3, 4]。しかし、厳密解法はNP困難に属し、最適解を得られる反面、解を求めるためには膨大な計算時間を必要とする。そこで西川ら[5]は計算時間の問題を回避するため、制約に対してラグランジュ緩和を施し、目的関数の中に制約を組み込むことで高速な近似解の求解を実現している。

一方、近似解を求める最適化手法として、進化的アルゴリズムも有効であることが報告されている。Petkovic[6]、Nieminenら[7]は、近似解法である遺伝的アルゴリズム[8]と、厳密解法である動的計画法、分岐限定法を比較したところ、遺伝的アルゴリズムの方がコストパフォーマンスに優れているとの結果

を得ている。また、進化的アルゴリズムには遺伝的アルゴリズムの他にも粒子群最適化[9]や差分進化[11]など多くのアルゴリズムがあり、Chandrasekarら[10]による進化的アルゴリズムの比較実験では、遺伝的アルゴリズムや粒子最適化に比べ、差分進化が解の精度や計算速度の点で優れたアルゴリズムであることを示している。

組合せ最適化に基づく文書要約手法において、進化的アルゴリズムを用いる研究は近年徐々に研究され始めている。Nandhiniら[12]は、文の重要度、読みやすさ、類似度などを考慮した組合せ最適化に遺伝的アルゴリズムを用い、他手法よりも安定した精度が出ることを示した。また、Rasimら[13]は、文書を構成している総文の平均を取ることで、文書の内容を総括するような代表文を生成し、この代表文と類似して、なおかつ冗長が少ない文の組合せを差分進化を利用して求めた。しかし、Rasimらの論文をもとに自身で追試実験を行ったところ、要約長制限を大きく逸脱した要約が生成されるなど結果に不十分な点があり、十分な考察がなされていないことが分かった。

文の組合せ最適化においては、文の重要度の決め方が大切とされる。一般に、文の重要度は、その文中に含まれる単語の重要度から計算されることが多い。単語の重要度の決め方には、従来からの手法であるtf-idfの指標を用いた方法に加え、近年、文書中に潜在トピックを考慮してトピックの観点から単語の重要度を定める手法の有効性が示されている。文書内のトピックの抽出には、Bleiら[14]によって提案された潜在的ディリクレ配分法(LDA:Latent Dirichlet Allocation)が多く用いられ、文書要約の研究だけでなく、情報検索や情報推薦など様々な応用に適用されている。文書要約においては、Murray[15]やAroraら[16]は、潜在トピックに基づいて重要文を抽出するのにLDAを用いている。また、Gaoら[18]や北島ら[17]は、文の類似度グラフ作成するためにLDAを用いた手法を提案している。両者とも高い精度を実現しており、要約生成におけるLDAの有用性を示している。

上述したことを踏まえて、本研究では、重要文の判定に潜在トピックを適用し、重要文の組合せ最適化手法に、進化的アルゴリズムの中でも計算効率の良いとされる差分進化アルゴリズムを用いた複数文書要約手法を提案する。

連絡先: 小林一郎, お茶の水女子大学大学院人間文化創成科学研究科理学専攻, 〒112-8610 東京都文京区大塚 2-1-1, koba@is.ocha.ac.jp

3. 潜在的ディリクレ配分法

本研究では、複数文書内の潜在的トピックを確率的に求めるトピックモデルとして潜在的ディリクレ配分法 (LDA)[14] を使用する。LDA は、文書はいくつかの話題 (トピック) が混合されて作られているという仮定の下、そのトピックの確率分布を導き出す手法である。各トピック t は単語分布ベクトル ϕ_t で表され、各文書 d はトピック分布ベクトル θ_d で表される。ベクトル ϕ_t において高い確率が割り振られた単語ほど、そのトピックの特徴を表す単語となり、ベクトル θ_d によって、文書の中にどのような比率でトピックが含まれているのかを推定することができる。

4. 差分進化

差分進化 (DE: Differential Evolution)[11] は進化的アルゴリズムの一種で、個体群を用いて確率的な多点探索を行う最適化アルゴリズムである。決められた世代数の中で、適合度を最大 (または最小) にするように個体群を進化させていくことで近似解を得ることができ、アルゴリズムの容易さ、計算速度の高速性、計算精度の高さから、最適化問題において有力な手法として注目されている。以下に、一般的な DE アルゴリズムを示す。

Step 1. 初期化。初期個体をランダムに N 個生成し、初期集団 $G(0) = \{P_1(0), P_2(0), \dots, P_N(0)\}$ を構成。

Step 2. 終了判定。予め設定した最大世代数 g_{max} に達していたら終了。

Step 3. 突然変異。各個体 $P_i(g)$ に対して、3 個体 $P_a(g), P_b(g), P_c(g)$ を、 $P_i(g)$ 及び互いに重複しないように個体群 $G(g)$ から選択する。そして、突然変異ベクトル $Q_i(g)$ を基底ベクトル $P_a(g)$ および差分ベクトル $P_b(g) - P_c(g)$ から以下のように求める。

$$Q_i(g) = P_a(g) + F(P_b(g) - P_c(g)) \quad (1)$$

ここで、 F は差分の調整パラメータである。

Step 4. 交叉。親ベクトル $P_i(g)$ と突然変異ベクトル $Q_i(g)$ を交叉し、子ベクトル $R_i(g)$ を生成する。

Step 5. 生存者選択。親ベクトル $P_i(g)$ と子ベクトル $R_i(g)$ を比べ、良い方を次世代に残す。

Step 6. Step 2. に戻る。

5. 差分進化を用いた文書要約

n 文から成る文書集合を要約する場合、文の組合せは、各文を要約として抽出するとき 1、しないとき 0 として長さ n の二値ベクトルで表される。差分進化を用いた文の組合せ最適化においては、各個体を文の組合せと捉え、要約長の制約を加味しながら、適合度 (= 重要度) が高い個体を次世代に残していくことで、要約として良い文の組合せを探す。

5.1 適合度関数の定義

なるべく文書集合内の重要な内容を多く含み、なおかつ内容の冗長が少ない個体 P_i を高く評価するような適合度関数を考える。ここでは、文書内の潜在トピックを考慮した 3 つの適合度関数 $f(\cdot)$ を提案する。

5.1.1 提案 1

提案 1 では、文の重要度と被覆度の積を適合度とすることで、重要な内容を多く含んだ、内容が網羅されている文の組合せを高く評価する関数を定義する。

$$f(P_i) = \frac{|W_i|}{V} \sum_{s=1}^n b_s p'_{i,s} \quad (2)$$

$|W_i|$ は個体 P_i を構成している単語の種類数、 V は要約対象文書セットを構成する単語の種類数を表し、 $\frac{|W_i|}{V}$ は、個体 P_i がどれだけ文書セットの単語を被覆しているのかを指す。

b_s は文 s の重要度を表し、LDA によって抽出されたトピックを考慮して以下の式で求める。

$$b_s = \sum_{t=1}^k b_{ts} \quad (3)$$

ここで、 b_{ts} は各トピック $t (t = 1, \dots, k)$ における文 s の重要度を表し、全てのトピックにおける重要度の総和によって、文の重要度を決定する。 b_{ts} は、以下の式で定義する。

$$b_{ts} = \frac{\sum_{w=1}^V \phi_{tw} y_{sw}}{\sqrt{|W_s|}} \cdot \theta_t \quad (4)$$

ϕ_{tw} はトピック t における単語 w の重要度、 y_{sw} は文 s に単語 w が含まれるとき 1、含まれないとき 0 の二値変数を表す。また、文長を考慮した評価を行うべく、文 s を構成する単語の重要度を総和したものを文 s の単語数 $|W_s|$ の平方根の逆数で割る。ここで、トピックの重要度は、文書セット内で多く含まれているトピックほど重要度が高いとの考えにより、文書セット中のトピック t の比率 θ_t を掛ける。

5.1.2 提案 2

提案 2 では、提案 1 で示した適合度関数 (2) の文の重要度 b_s の求め方を変更した方法を試す。ここでは、文は文書内に存在する各トピックの代表文に類似しているほど重要であると、以下のように定義する。

$$b_s = \max_{t=1,2,\dots,k} \{sim(w_{ts}, O_t)\} \quad (5)$$

O_t はトピック t の代表文を表し、これらはベクトル $O_t = [o_{t1}, o_{t2}, \dots, o_{tV}] (t = 1, 2, \dots, k)$ として表される。ここで O_t には、LDA で抽出したトピックごとの単語分布 ϕ_t を使用する。 w_{ts} はトピック t における文 s のベクトルを表し、 $w_{ts} = \sum_{j=1}^V o_{tj} x_{sj}$ によって求める。ここで、 x_{sj} は、文 s に単語 j が入っているとき 1、そうでないとき 0 の二値変数となる。

$sim(a, b)$ はベクトル a, b 間のコサイン類似度を表し、 $\max_{t=1,2,\dots,k} \{sim(w_{ts}, O_t)\}$ によって、 k 個の代表文のうち、最も類似している代表文とのコサイン類似度を文 s の重要度とする。コサイン類似度は以下の式で求められる。

$$sim(a, b) = \frac{\sqrt{\sum_{j=1}^V a_j^2 \cdot \sum_{j=1}^V b_j^2}}{\sum_{j=1}^V a_j b_j} \quad (6)$$

5.1.3 提案 3

提案 3 では、文の重要度を求めるのに式 (5) を用い、内容の冗長性削減を、被覆度を掛けるのではなく組合せ内の文の類似

度で割ることで行う適合度関数を定義する．

$$f(P_i) = \frac{\sum_{s=1}^{n-1} \sum_{r=s+1}^n (b_s + b_r) p'_{i,s} p'_{i,r}}{\sum_{s=1}^{n-1} \sum_{r=s+1}^n \text{sim}(w_s, w_r) p'_{i,s} p'_{i,r}} \quad (7)$$

w_s は、文 s の単語ベクトルであり、 $w_s = [w_{s,1}, w_{s,2}, \dots, w_{s,V}]$ と表される．ここで、 $w_{s,a}$ は、文 s における単語 a の重要度を表し、tf-idf 値によって求める．

$$w_{s,a} = \text{tf}_{sa} \times \log\left(\frac{n}{n_a}\right) \quad (8)$$

tf_{sa} は単語 a が文 s に含まれる割合を表し、 n は総文数、 n_a は総文のなかで単語 a が含まれる文の数を表す．そして、 $\sum_{s=1}^{n-1} \sum_{r=s+1}^n \text{sim}(w_s, w_r) p'_{i,s} p'_{i,r}$ により、個体 P_i において選択されている文同士のコサイン類似度の総和を求める．

5.2 差分進化を用いた文書要約の流れ

DE によって得られた最終世代目のベスト個体をシステム要約として生成する．通常の DE において実数値ベクトルで表されている個体を二値ベクトルに変換する作業の追加や、要約長の制約を加味した生存者選択などの改良点がある．以下、改良 DE の手順を詳細に説明する．

5.2.1 初期集団生成

DE では世代 $g = 0, 1, \dots, g_{max}$ の中で N 個の個体からなる集団 $G(g)$ を進化させていく．ここで、世代 g の i 番目の個体 $P_i(g)$ は以下のおく．

$$P_i(g) = [p_{i,1}(g), p_{i,2}(g), \dots, p_{i,n}(g)]$$

初期集団 ($g = 0$) は、予め与える必要があり、多様性に富んだ個体を用意するために、以下の式で個体 $P_i(0)$ の s 番目の要素を求める．

$$p_{i,s}(0) = 10 - 20(1 - \text{rand}_{i,s})^{1/(n+1)} \quad (9)$$

各要素ごとにランダム値 $0 \leq \text{rand}_{i,s} \leq 1$ を求め、 $[-10, 10]$ の間の値を求める．ここで n は、 $[-10, 10]$ の値の出現確率を操作するパラメータであり、 n が大きいほど出現確率が -10 側に偏る．

5.2.2 突然変異

突然変異ベクトル Q_i を求める一般的な式は、式 (1) であるが、解の精度を高めるため、新たな式を提案している研究が多数ある．本研究では Rasim ら [13] が提案した以下の式を用いる．

$$Q_i(g) = P_a(g) + F \cdot (P_{best}(g) - P_b(g)) + F \cdot (P_{best}(g) - P_c(g)) \quad (10)$$

$P_a(g), P_b(g), P_c(g)$ は、個体 $P_i(g)$ を除いた集団 $G(g)$ のなかからランダムに選んだ個体である．また、 P_{best} は、集団 $G(g)$ の中で最も良い個体を表す．

5.2.3 交叉

親ベクトル $P_i(g)$ と突然変異ベクトル $Q_i(g)$ を交叉率 $CR(g)$ で交叉させ、子ベクトル $R_i(g)$ を生成する．ここで、子ベクトルの各要素 $r_{i,s}(g)$ は以下のルールによって、親ベクトルの要素 $p_{i,s}(g)$ または突然変異ベクトルの要素 $q_{i,s}(g)$ を継承する．

$$r_{i,s}(g) = \begin{cases} q_{i,s}(g) & (\text{if } \text{rand}_{i,s} \leq CR(g) \text{ or } s = s_{rand}) \\ p_{i,s}(g) & (\text{otherwise}) \end{cases}$$

s_{rand} はランダムに選ばれた $1, 2, \dots, n$ のいずれかの値で、 s_{rand} 番目の要素は必ず突然変異ベクトルの要素を取るようにすることで、子ベクトルが親ベクトルと同等になることを防ぐ．

また、世代が進むにつれ集団は良いものとなってくるため、子ベクトルを生成する際は親ベクトルの要素を多く取り入れた方がよい．そこで、交叉率を世代が経つにつれ徐々に減らしていく．

$$CR(g) = CR(0) \cdot \text{sigm}(g_{max}/(2 \cdot (g + 1))) \quad (11)$$

$\text{sigm}(\cdot)$ はシグモイド関数であり、世代が 0 から g_{max} に近づくにつれ徐々に交叉率を減らし、親ベクトルの要素が強い子ベクトルを生成するようにする． $CR(0)$ は初期世代の交叉率であり、予め与えておく．

5.2.4 生存者選択

親ベクトル $P_i(g)$ と子ベクトル $R_i(g)$ を評価し、次世代の生存者 $P_i(g+1)$ を選択する．ここで、適合度を評価するためには個体が二値ベクトルである必要があるため、 $p_{i,s}(g)$ を 0 より大きければ 1、小さければ 0 として二値化する．そして、要約長制約を加味した選択を行うため、以下のルールに基づき次世代の生存者を選択する．

- どちらも制約を満たしている場合、適合度が大きい方を選択
- どちらかが制約を満たしていない場合、いかなる場合も制約を満たしている方を選択
- どちらも制約を満たしていない場合、制約を大きく違反していない方を選択

6. システム要約評価実験

6.1 実験仕様

本実験では、要約評価ワークショップ DUC04 の Task2 で使用されたデータセットを用いる．データセットには、話題の異なる 50 の文書セットが用意されており、1 文書セットあたり 10 個のニュース記事から成っている．各文書セットに対して、長さ 665 バイト以内の要約を 10 回生成し、ROUGE-1 値を用いて 10 個の要約の平均精度を測る．ROUGE-1 値は、ストップワードを含めた評価 “with” と含めない評価 “without” についてそれぞれ求める．実験環境は、OS は Ubuntu 12.04.3、CPU は AMD FX(tm)-8120 1.4GHz を用いた．

LDA の設定は、トピックの推定にギブスサンプリングを利用し、反復回数は 100 回、ハイパーパラメータ α, β はそれぞれ 0.1 に設定した．文書集合内のトピック数の推定にはパープレキシティを用いた．

改良 DE は、最大世代 $g_{max} = 10000$ 、個体数 $N = 50$ として実験を行う．細かなパラメータの設定は、初期個体のパラメータは $n = 5$ 、差分パラメータは Rasim ら [13] を参考にし $F = 0.45$ 、初期交叉率は $CR(0) = 0.7$ と設定した．

6.2 結果と考察

表 1 に、提案手法、他手法の精度を示す．ここで、5.1.1 節の適合度関数を用いた手法を Topic-DE_{fit1}、5.1.2 節の適合度関数を用いた手法を Topic-DE_{fit2}、5.1.3 節の適合度関数を用いた手法を Topic-DE_{fit3} とする．他手法においては、Topic-OPT は、著者らの先行研究で提案した手法 [19] であり、Topic-DE_{fit1} と同様の文の重要度や被覆度を用いて、文の組合せ最適化手法に厳密解法ソルバー CPLEX を適用したものである．

表 1: DUC'04 各手法の精度

手法	with	without	計算時間 (秒)
Topic-DE _{fit1}	0.345	0.249	458
Topic-DE _{fit2}	0.337	0.232	447
Topic-DE _{fit3}	0.287	0.145	451
他手法	with	without	計算時間 (秒)
Topic-OPT	0.389	0.326	9548
CLASSY	0.382	0.309	-

また, CLASSY は DUC'04 で最も精度の高かった手法であり, 要約手法の指標とされる。

提案手法 3 つを比較すると, with, without 共に, 文の重要度に Topic-DE_{fit1} が最も高く, 次いで Topic-DE_{fit2}, 最も ROUGE-1 値が低くなったのが Topic-DE_{fit3} となった。Topic-DE_{fit1} と Topic-DE_{fit2} の比較より, トピックを考慮した文の重み付けの際には, 代表文との類似度をとるよりも, 単語の重要度の総和による重み付けの方が有効であることが分かった。また, Topic-DE_{fit2} と Topic-DE_{fit3} の比較では, 冗長性を考慮する際に, 組合せ内の類似度よりも, 組合せがどれほど文書セットを被覆しているかを考慮する方が良い評価となることが分かった。

さらに, Topic-OPT と比較してみると, 計算時間は最適化手法に DE を用いたことにより, 約 9500 秒から約 450 秒へと著しく削減でき, 文書集合の大きさに関わらず安定した計算時間で要約を出力できた。一方, ROUGE-1 値は下がった。Topic-OPT と Topic-DE_{fit1} の精度に差が出てしまった原因としては, Topic-OPT では目的関数において各文に対して重要度と被覆度を求めていたが, Topic-DE_{fit1} では, 組合せに対して重要度と被覆度を求めていたため, 適合度関数に更なる工夫が必要だったのではないかと考えられる。

7. おわりに

本研究では, 計算時間削減のために最適化手法に差分進化を用いた複数文書要約の提案を行った。文の評価には, トピックごとの文の重要度の総和を文の重要度とする定義と, トピックの代表文との類似度を重要度とする定義を用いたが, 実験の結果, 前者の方が適した評価ができていたことが分かった。また, 内容の冗長性を考慮する際, 組合せ内の類似度を測るよりも, 組合せがどれだけ文書セット全体を被覆しているのかという被覆度を考慮したほうが効果的であった。厳密解法に基づく手法との比較したところ, 差分進化の計算速度の速さが示されたが, 差分進化の世代数が十分でなかったこともあり, 精度の点で劣っていた。そこで今後は, 世代数を増やした実験と共に, 精度向上のための適合度関数の改善を課題とする。

参考文献

- [1] 岡崎直観, 松尾豊, 石塚満: 関連する複数新聞記事からの重要文抽出法, 第 3 回 MYCOM 資料, 2002.
- [2] Ryan McDonald: A Study of Global Inference Algorithms in Multi-Document Summarization, Proc. of the 29th European Conference on Information Retrieval, pp557-564, 2007.
- [3] 平尾努, 鈴木潤, 磯崎秀樹: 最適化問題としての文書要約, 人工知能学会論文誌 vol.24, pp.223-231, 2009.

- [4] 高村大也, 奥村学: 施設配置問題による文書要約のモデル化, 人工知能学会論文誌, Vol. 25, No. 1, pp. 174-182, 2010.
- [5] 西川仁, 平尾努, 牧野俊朗, 松尾義博: ラグランジュ緩和による複数文書要約の高速求解, 言語処理学会論文誌, Vol. 18, pp. 1071-1074, 2012.
- [6] Dusan Petkovic: Dynamic Programming Algorithm vs. Genetic Algorithm: Which is Faster?, Research and Development in Intelligent Systems XXVII, pp 483-488, 2011.
- [7] K Nieminen, S Ruuth, I Maros: Genetic algorithm for finding a good first integer solution for MILP, 2003.
- [8] Holland, J.H.: Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence, University of Michigan Press, 1975
- [9] J. Kennedy and R. C. Eberhart: Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, Vol. 1498 of Lecture Notes in Computer Science, pp. 1942-1948, 1995.
- [10] Chen, N, Vapnik, J. P: Performance Comparison of GA, DE, PSO and SA Approaches in Enhancement of Total Transfer Capability using FACTS Devices, Journal of Electrical Engineering & Technology, Vol. 7, No. 4, pp. 493-500, 2012.
- [11] Storn, R. and Price, K.: Minimizing the Real Functions of the ICEC96 Contest by Differential Evolution, in Proc. of the International Conference on Evolutionary Computation, pp. 842-844, 1996.
- [12] K. Nandhini and S. R. Balasundaram, Use of Genetic Algorithm for Cohesive Summary Extraction to Assist Reading Difficulties, Applied Computational Intelligence and Soft Computing Volume 2013 Article ID 945623, 11 pages, 2013.
- [13] Rasim M. Alguliev, Ramiz M. Aliguliyev, Chingiz A. Mehdiyev: Sentence selection for generic document summarization using an adaptive differential evolution algorithm, Swarm and Evolutionary Computation 1(4), pp. 213-222, 2011.
- [14] David M. Blei, Andrew Y. Ng, Michael I. Jordan: Latent Dirichlet Allocation, Journal of Machine Learning Research 3, pp. 993-1022, 2003.
- [15] Kenton W. Murray, Summarization by Latent Dirichlet Allocation: Superior Sentence Extraction through Topic Modeling, A senior thesis for Bachelors degree, Princeton University, 2009.
- [16] Rachit Arora, Balaraman Ravindran: Latent Dirichlet Allocation Based Multi-Document Summarization, Proceedings of the second workshop on Analytics for noisy unstructured text data, pp. 91-97, 2008.
- [17] 北島理沙, 小林一郎: トピックを考慮したグラフによる複数文書要約への一考察, 第 19 回言語処理学会年次大会, pp. 504-507, 2013.
- [18] Dehong Gao, Wenjie Li, You Ouyang, Renxian Zhang, LDA-Based Topic Formation and Topic-Sentence Reinforcement for Graph-Based Multi-document Summarization, Lecture Notes in Computer Science Volume 7675, pp 376-385, 2012.
- [19] 重松遥, 小林一郎, 潜在トピックの比率に基づく文書要約手法の提案, 4N-9, 第 26 回人工知能学会全国大会, 山口, 2013.