ラグランジュ緩和による複数文書要約の高速求解

西川 仁[†]　平尾 努[‡]　牧野 俊朗[†]　松尾 義博[†]
[†]NTTサイバースペース研究所　[‡]NTTコミュニケーション科学基礎研究所
{nishikawa.hitoshi, hirao.tsutomu, makino.toshiro, matsuo.yoshihiro}@lab.ntt.co.jp

概要

本稿では，ナップサック問題に基づく要約モデルに冗長性を制限する制約を加えた新しい要約モデルを提案する．加えて，提案する要約モデルの，ラグランジュヒューリスティックを用いたデコード法を提案する．提案する要約モデルをROUGEによって評価したところ，提案する要約モデルは，その最適解において最大被覆問題以上の性能を持つことを示す．また，速度においては，最大被覆問題の最適解とROUGEにおいて同等の近似解を高速に得られることを示す．

1　はじめに

現在の文書要約技術の多くは文を単位にした処理を行っている [11]．具体的には，まず入力された文書集合を文分割器を用いて文集合に変換する．次に，文集合から，要約長を満たす文の組み合わせを，要約としての善し悪しを与える何らかの基準に基づき選び出す．最後に，選び出された文に適当な順序を与えることによって要約は生成される．

近年の文書要約は最大被覆問題の形で定式化されることが多い [2, 4, 10, 12]．これは，入力文書集合に含まれるユニグラムやバイグラムといった単位を，与えられた要約長を満たす文の集合によって出来る限り被覆することによって要約を生成するものである．最大被覆問題は要約モデルとして高い能力を持つことが実証されている [12]．一方，その計算複雑性はNP困難である [8]．そのため，入力文書集合が大規模になった場合，最適解を求める際に多大な時間を要する恐れがある．

個別の文に重要度を与え，与えられた要約長内で文の重要度の和を最大化するナップサック問題として文書要約を定式化した場合，動的計画ナップサックアルゴリズムを用いると擬多項式時間でナップサック問題の最適解を得ることが出来る [5, 8]．しかし，個別の文にスコアを与え，文のスコアの和を最大化する形式であるため，最大被覆問題とは異なり冗長な要約を生成する恐れがある．

本稿では，動的計画法によって擬多項式時間で最適解を得られるナップサック問題の性質を活かし，ナップサック問題としての要約モデルに，要約の冗長性を制限する制約を陽に加えた要約モデル（**冗長性制約付ナップサックモデル**）を提案する．この制約を加えることで要約の冗長性を制限することが出来るが，再び最適解の求解は困難となる．

そこで，本稿では，ラグランジュヒューリスティック [3, 13] を用いて冗長性制約付ナップサックモデルの近似解を得る方法を提案する．ラグランジュヒューリスティックはラグランジュ緩和によって得られる緩和解から何らかのヒューリスティックを用いて実行可能解を得るもので，集合被覆問題において良好な近似解が得られることが知られている [13]．具体的には，まず，上で述べた冗長性を制限する制約をラグランジュ緩和し，目的関数に組み込む．次に，この目的関数の最適解，すなわち緩和解を動的計画法を用いて得る．最後に，緩和解からヒューリスティックを用いて実行可能解を得る．

本稿の新規性，貢献を以下にまとめておく．

- 我々の知る限り，本稿で提案する要約モデルおよびデコーディングアルゴリズムは，自動要約研究の文脈において初めてのものである．
- 提案する要約モデルは，最適解において，最大被覆問題を用いた要約モデルに対して，ROUGE[9]において同等以上の性能を持つことを示す．
- 提案するデコード法によって得られる，提案する要約モデルの近似解は，最大被覆問題の最適解とROUGEにおいて同等であることを示す．また，最大被覆問題のソルバによる求解に比べ高速に解が得られることを示す．

以下，2節では関連研究について述べる．3節では提案する要約モデルについて述べる．4節では，デコードのためのアルゴリズムについて述べる．5節では提案手法の性能を実験によって検証する．6節では本稿についてまとめる．

2　関連研究

最大被覆問題による要約モデルはFilatovaらによって提案された [2]．Filatovaらはこれを貪欲法 [6] で解いた．高村らはこれに整数計画問題としての厳密な定式を与え，近似解法と分枝限定法による結果を報告している [12]．最大被覆問題に基づく要約モデルの応用としては，レビュー文書の要約 [10] や音声文書の要約 [4] がある．

平尾らは単一文書要約をナップサック問題として定式化した場合，これを動的計画ナップサックアルゴリズムで擬多項式時間で解けることを示した [5].

依存構造解析や統計的機械翻訳の分野では，ラグランジュ緩和を用いることで，表現力の高いモデルを利用する場合でも最適解を高速に探索する手法が提案されている [1, 7].

3 要約モデル

n 文の入力および，それらに含まれる m 個の何らかの単位を考える．この単位はユニグラムやバイグラムなどである．$\mathbf{x}$ を文 i が要約に含まれる際に $x_i = 1$ となる決定変数を要素とするベクトルとする．$\mathbf{z}$ を単位 j が要約に含まれる際に $z_j = 1$ となる決定変数を要素とするベクトルとする．$\mathbf{w}$ を単位 j に対する何らかのスコア w_j を要素とするベクトルとする．行列 $\mathbf{A}$ の要素 a_{ji} を文 i が単位 j を含んでいる数とする．$\mathbf{l}$ を文 i の長さ l_i を要素とするベクトルとする． K を要約長とする．このとき，最大被覆問題は以下のよう定式化される．

$$\max_{\mathbf{z}} \quad \mathbf{w}^\top \mathbf{z} \tag{1}$$
$$s.t. \quad \mathbf{A}\mathbf{x} \geq \mathbf{z} \tag{2}$$
$$\mathbf{x} \in \{0,1\}^n \tag{3}$$
$$\mathbf{z} \in \{0,1\}^m \tag{4}$$
$$\mathbf{l}^\top \mathbf{x} \leq K \tag{5}$$

この問題の制約 (2) を $\mathbf{A}\mathbf{x} = \mathbf{z}$ とし，また制約 (4) を 0 と 1 からなるベクトルでなく 0 を含む自然数からなる $\mathbf{z} \in \{\mathbb{N}^0\}^m$ とすると，問題はナップサック問題となり，動的計画ナップサックアルゴリズムによって擬多項式時間 $O(nK)$ で最適解を求めることができる．

一方，冗長性を除去する制約として働いていた制約 (4) が緩和されたことによって，冗長な要約が生成される恐れが高まる．そこで，以下のように，陽に冗長性を制限する問題を考える（冗長性制約付ナップサックモデル）.

$$\max_{\mathbf{z}} \quad \mathbf{w}^\top \mathbf{z} \tag{6}$$
$$s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{z} \tag{7}$$
$$\mathbf{x} \in \{0,1\}^n \tag{8}$$
$$\mathbf{z} \in \{z_j | \mathbb{N}^0 \cap [0, r_j]\}^m \tag{9}$$
$$\mathbf{l}^\top \mathbf{x} \leq K \tag{10}$$

この問題は制約 (9) によって，要約に含まれる単位 j の数が r_j 個までと制限されている[1]．この制約を与えるベクトル $\mathbf{r}$ により，要約の冗長性を制限することができるが，再び最適解の求解は困難となる[2].

[1]冗長性を制限する制約を，単位に対してではなく文に対して与えることもできる．例えば，ある単位を要約に 3 つまでしか含めないという制約は，その単位を含む文を要約に 3 つまでしか含めないという制約としても記述できる．そのときこの問題は排他制約付ナップサック問題 [14] となる.

[2]冗長性制約付ナップサックモデルも動的計画法によって擬多項式時間にて最適解を求めることが可能である．しかし，そのオーダは $O(nK\prod_{j=1}^{m} r_j)$ となり，文数より語彙の数が多い $n \ll m$ の条件下では事実上指数オーダとなるため，最適解の求解は困難である.

そこで，求解を困難にする原因である制約 (9) をラグランジュ緩和した次の問題を考える．

$$\max_{\mathbf{z}} \quad \mathbf{w}^\top \mathbf{z} + \boldsymbol{\lambda}(\mathbf{r} - \mathbf{z}) \tag{11}$$
$$s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{z} \tag{12}$$
$$\mathbf{x} \in \{0,1\}^n \tag{13}$$
$$\mathbf{z} \in \{\mathbb{N}^0\}^m \tag{14}$$
$$\mathbf{l}^\top \mathbf{x} \leq K \tag{15}$$

式 (11) では，制約 (9) が破られている場合には，非負のラグランジュ乗数 $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)$ によって目的関数にペナルティを与える．すなわち，要約の中に単位 j が $r_j + 1$ 個以上含まれている際には，その単位の重み w_j がラグランジュ乗数 λ_j によって低下する．これにより，次に動的計画ナップサックアルゴリズムによって式 (11) が解かれる際に単位 j が要約に含まれづらくなり，結果として制約が満たされる可能性が高まる．ラグランジュ乗数は，この問題のラグランジュ双対問題 $L(\boldsymbol{\lambda}) = \min_\lambda \{\max_{\mathbf{z}} \mathbf{w}^\top \mathbf{z} + \boldsymbol{\lambda}(\mathbf{r} - \mathbf{z})\}$ は劣勾配法によって解くことで得る.

緩和する制約 (9) は不等式制約であるため，等式制約の場合 [1, 7] のように厳密解を得ることはできない．そこで，以下に述べる手続きで近似解を得る.

4 デコード

本稿では，以下の手続きによって式 (11) の近似解を得る.

1. ラグランジュ乗数を 0 に初期化する.
2. 以下の手続きを既定の回数だけ繰り返す.
   (a) 動的計画ナップサックアルゴリズムで式 (11) の最適解を得る.
   (b) (a) で得た最適解が制約を満たしているときは 3. へ．そうでなければ最適解からヒューリスティックを用いて実行可能解を得る.
   (c) (b) で得た実行可能解がこれまでの下界を上回るものであれば下界を更新する.
   (d) ラグランジュ乗数を更新する.
3. 下界を出力して終了する.

動的計画ナップサックアルゴリズムを図 1 に示す．K は要約長，$\mathbf{l}$ は各文の長さを格納したベクトル， n は入力される文の数， $\mathbf{s}$ は各文のスコアである．各文のスコア $\mathbf{s}$ は，各文が含む単位の数を格納した行列 $\mathbf{A}$ と各単位のスコア $\mathbf{w}$ ，およびラグランジュ乗数 $\boldsymbol{\lambda}$ から計算できる.

動的計画ナップサックアルゴリズムで得た解，すなわち緩和解が制約を満たしていればその時点で得られ

```
INPUT K, l, n, s
SET x = ∅
FOR j = 0 to K
  T[0][j] = 0
ENDFOR
FOR i = 1 to n
  FOR j = 0 to K
    T[i][j] = T[i − 1][j]
    U[i][j] = 0
  ENDFOR
  FOR j = l[i] to K
    IF T[i − 1][j − l[i]] + s[i] ≥ T[i][j] THEN
      T[i][j] = T[i − 1][j − l[i]] + s[i]
      U[i][j] = 1
    ENDIF
  ENDFOR
ENDFOR
j = K
FOR i = n to 1
  IF U[i][j] = 1 THEN
    x_i = 1
    j = j − l[i]
  ENDIF
ENDFOR
OUTPUT x
```

図 1: 動的計画ナップサックアルゴリズム

ている最良の下界を出力して終了する．制約に違反しているときは以下のようにラグランジュ乗数 $\boldsymbol{\lambda}$ を更新する．

$$\lambda_j \leftarrow \max\left(\lambda_j + \alpha\frac{\mathrm{UB}-\mathrm{LB}}{||\mathbf{d}||^2}, 0\right) \tag{16}$$

ここで，UB は緩和解の中で最も値が低いもの，すなわち上界であり，LB は実行可能解のうち最も値が高いもの，すなわち下界である．$\mathbf{d}$ はラグランジュ双対問題の劣勾配である．α は探索を制御するパラメタである．

緩和解から実行可能解を取得するヒューリスティックとして，本稿では以下の手続きを用いた．

1. 冗長性制約に違反している単位を含む文の中で，文スコアを文の長さで割った値が最も小さいものを要約から取り除く．これを要約が冗長性制約を満たすまで繰り返す．
2. 冗長性制約を満たす要約が得られたら，元問題からその要約の文，長さを取り除いた部分問題を生成し，この部分問題を貪欲法 [6] で解く．

5 実験

本節では提案手法を評価するための実験について述べる．ここでは，以下の 2 点を評価する．

- **要約の品質** ： 生成される要約の品質を評価する．品質の評価には要約の自動評価尺度である ROUGE[9] を用いる．
- **処理時間**：要約の生成に要する時間を計測する．要約に要した時間は，5.1 で述べる 30 文書集合全ての要約を求めるまでにかかった時間とした．

実験では，以下の 4 手法を比較する．

- **最大被覆モデル (MCKP)**：式 (1) を目的関数，式 (2) から式 (5) を制約として，ソルバを用いて解を求める．ソルバには `lp_solve`[3]を用いた．
- **ナップサックモデル (DPKP)**：式 (6) を目的関数，式 (7)，(8)，(10) を制約として，図 1 に示す動的計画法で解を求める．
- **冗長性制約付ナップサックモデル (RCKP)**：提案手法の最適解を出力するもの．DPKP に制約 (9) が追加されたもの．式 (6) を目的関数，式 (7) から式 (10) を制約として，ソルバを用いて解を求める．ソルバには MCKP と同様に `lp_solve` を用いた．
- **冗長性制約付ナップサックモデル＋ラグランジュヒューリスティック (RCKP-LH)**：提案手法の近似解を出力するもの．式 (11) を目的関数として，ラグランジュヒューリスティックによって近似解を求める．繰り返し回数として，10 回の場合と 100 回の場合をそれぞれ評価した．

DPKP と RCKP-LH のデコーダはそれぞれ Perl で実装した．全てのプログラムは， CPU として Intel Xeon X5560 (Quad Core) 2.8GHz CPU × 2 ，64Gbyte のメモリを搭載した計算機上で動作させた．

5.1 コーパス

コーパスとして，TSC-3[4]を用いる．TSC-3 コーパスは複数文書要約のための評価セットとなっており，30 セットの文書集合からなる．それぞれの文書集合には 2 つの要約長の参照要約が付与されているが，本稿では短い要約長の参照要約のみを用いた．

5.2 パラメタの設定

実験のため，以下に示す 3 つのパラメタを推定する必要がある．それぞれ以下に示す方法でパラメタを設定した．

- α：劣勾配法によるラグランジュ乗数の更新幅 α は，ラグランジュ乗数の更新回数の逆数とした．
- $\mathbf{r}$：単位毎に許す冗長性を調整することができる．そこで，各単位 j が入力文書集合に含まれる回数を tf_j としたとき，その 2 乗根を切り下げた値 $r_j = \lfloor\sqrt{\mathrm{tf}_j}\rfloor$ を許容する冗長性とする．

[3]http://lpsolve.sourceforge.net/
[4]http://lr-www.pi.titech.ac.jp/tsc/tsc3.html

表 1: ROUGE による評価の結果

	ROUGE-1	ROUGE-2
MCKP	0.456	0.185
DPKP	0.450	0.205
RCKP	0.482	0.218
RCKP-LH(10)	0.451	0.207
RCKP-LH(100)	0.459	0.210

表 2: 要約に要した時間

	時間（秒）
MCKP	893551.7
DPKP	7.6
RCKP	2437.4
RCKP-LH(10)	68.6
RCKP-LH(100)	564.1

- $\mathbf{w}$: 各単位 j は内容語とし，重みはそれらの入力文書集合中での頻度 tf_j とした．

α は RCKP-LH のみが用い，$\mathbf{r}$ は RCKP および RCKP-LH が用いる． $\mathbf{w}$ は全ての手法で共通である．

5.3 結果

ROUGE による評価結果を表 1 に示す．ROUGE-1，ROUGE-2 ともに，RCKP が他の全ての手法に対して有意に高い値を得た．いずれの尺度においても，他の 4 手法間において有意差は認められなかった．

要約に要した時間を表 2 に示す．`lp_solve` による MCKP の 30 文書集合の要約には 1 週間以上の時間を要した．DPKP は，擬多項式時間オーダを反映して高速な求解が可能であった． `lp_solve` による RCKP のデコードも，MCKP に比べれば高速である．RCKP-LH は内部で DPKP を繰り返し解くものであるため，DPKP の計算時間に RCKP-LH の繰り返し回数をかけた時間に近いものになっている．

5.4 考察

実験により，ROUGE において RCKP は MCKP 以上の性能を持つことが示された．MCKP は，目的関数上，幅広い単位を含む要約を指向する．それに対し RCKP は文書集合中に頻出する単位であれば一定の冗長性を要約に許す．人間が作成した参照要約は，文書集合の主題となる単位をある程度重複して含んでいると考えられる．そのため，RCKP は MCKP に対して優位性があったものと考えられる．

6 おわりに

本稿では冗長性制約付ナップサックモデルを提案した．これはナップサック問題に基づく要約モデルに冗長性を制限する制約を加えた要約モデルである．ROUGE によってその性能を評価し，また要約に要する時間を報告した．実験の結果から，提案するモデルは ROUGE において最大被覆問題と同等の性能を持っており，速度においては最大被覆問題を上回る速度でデコードできることを示した．

参考文献

[1] Yin-Wen Chang and Michael Collins. Exact Decoding of Phrase-based Translation Models through Lagrangian Relaxation. In Proc. of EMNLP, 2011.

[2] Elena Filatova and Vasileios Hatzivassiloglou. A Formal Model for Information Selection in Multi-Sentence Text Extraction. In Proc. of Coling, 2004.

[3] Salim Haddadi. Simple Lagrangian heuristic for the set covering problem. European Journal of Operational Research. 97:200–204, 1997.

[4] Ryuichiro Higashinaka, Yasuhiro Minami, Hitoshi Nishikawa, Kohji Dohsaka,Toyomi Meguro, Satoshi Kobashikawa, Hirokazu Masataki, Osamu Yoshioka, Satoshi Takahashi, and Genichiro Kikui. Improving HMM-based Extractive Summarization for Multi-Domain Contact Center Dialogues. In Proc. of SLT, 2010.

[5] 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, 24(2):223–231, 2009.

[6] Samir Khuller, Anna Moss and Joseph Naor. The budgeted maximum coverage problem, Information Processing Letters, 70(1):39–45, 1999.

[7] Terry Koo and Michael Collins. Efficient Third-order Dependency Parsers. In Proc. of ACL, 2010.

[8] Bernhard Korte and Jens Vygen. Combinatorial Optimization (Third Edition). Springer-Verlag, 2008.

[9] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In Proc. of the Workshop on Text Summarization Branches Out, 2004.

[10] Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo and Genichiro Kikui. Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering. In Proc. of Coling, 2010.

[11] 奥村学, 難波英嗣. テキスト自動要約. オーム社, 2005.

[12] 高村大也, 奥村学. 最大被覆問題とその変種による文書要約モデル. 人工知能学会論文誌, 23(6):505–513, 2008.

[13] Shunji Umetani and Mutsunori Yagiura. Relaxation heuristics for the set covering problem. Journal of the Operations Research Society of Japan, 50:350–375, 2007.

[14] Takeo Yamada, Seiji Kataoka and Kohtaro Watanabe. Heuristic and exact algorithms for disjunctively constrained knapsack problem. IPSJ Journal, 43:2864–2870, 2002.