高速な精度保証付き圧縮型要約

坂上 晋作　平尾 努　西野 正彬　永田 昌明
NTT コミュニケーション科学基礎研究所
{sakaue.shinsaku, hirao.tsutomu}@lab.ntt.co.jp
{nishino.masaaki, nagata.masaaki}@lab.ntt.co.jp

1 はじめに

文書要約において盛んに研究されている手法の一つに，抽出型要約 (extractive summarization) がある．抽出型要約では，与えられた文書を各文を要素とする集合と見做し，要約長の制限を満たすように文をいくつか取り出すことで，要約を作成する．こうした方法は，高速に要約を作成できる上に，非常に高い ROUGE スコアを達成し得ることが知られている [6]．抽出型要約のための有効な手法の一つに，単調劣モジュラ最大化に対する貪欲法を用いた方法がある [10]．この手法には，(1) 劣モジュラ性をもつ様々な目的関数に対し適用可能であること，(2) 貪欲法により高速に要約が得られること，(3) 貪欲法の近似精度を理論的に保証できること，という 3 つの利点がある．

一方で，抽出型要約には次のような課題がある．今，要約すべき文書が多くの長い文を含んでいるとする．こうした文は，重要な部分と冗長な部分を同時に持っている場合が多い．この状況で抽出型要約を用いると，冗長な部分を含んだ長い文をそのまま抽出して要約に含めてしまうため，要約長の制限が厳しい場合には，十分な情報を含んだ要約を得ることができない．

こうした問題を解決するのが，圧縮型要約 (compressive summarization) である．圧縮型要約では，各文が何らかの木構造（例えば，依存構造木）を持っていると考え，その構造を保存するように文を圧縮し，圧縮された文からなる要約を作成する．こうすることで，冗長性を含む長い文から重要な部分のみを取り出せるため，要約長を抑えつつ十分な情報を含んだ要約を作成できる．圧縮型要約では，文の構造を保存するという制約も考慮しなければならないため，要約を得るために解くべき問題は，抽出型要約の場合よりも複雑になる．圧縮型要約の既存研究としては，整数線形計画法（integer linear programming, 以下 ILP）を用いる手法 [1] や，貪欲法と動的計画法による劣モジュラ最大化を用いる手法 [12] などがあるが，抽出型要約の場合の劣モジュラ最大化に対する貪欲法 [10] のような，(1) 様々な目的関数への適用可能性，(2) 貪欲法を用いることによる高速性，(3) 近似精度の保証，の 3 つの利点を持つ手法は今まで知られていなかった．本稿では，これらの 3 つの利点を兼ね備えた，劣モジュラ最大化に対する貪欲法に基づく圧縮型要約法を提案する．実験では，提案法が ILP に基づく手法と同程度の目的関数値や ROUGE_1 スコアを達成する一方で，100 から 400 倍程度高速に動作することを確認する．

2 問題設定

要約すべき文書（群）が与えられた時，その文書内の全てのチャンク（単語列）の集合を V とする．また，$f : 2^V \to \mathbb{R}$ を単調劣モジュラな $f(\emptyset) = 0$ を満たす目的関数とする．ただし，任意の $A, B \subseteq V$ に対して $f(A \mid B) \coloneqq f(A \cup B) - f(B)$ とすると，単調性と劣モジュラ性はそれぞれ次のように表される．

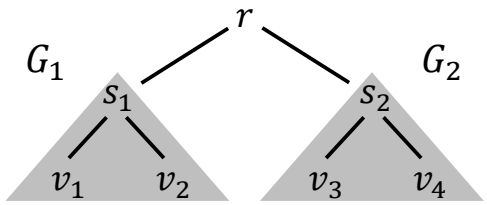
$$f(B) \geq f(A) \qquad \forall A \subseteq B,$$
$$f(\{v\} \mid A) \geq f(\{v\} \mid B) \qquad \forall A \subseteq B, \forall v \notin B.$$

要約生成のために用いられる様々な目的関数は，上記の性質を満たすことが知られている [11].

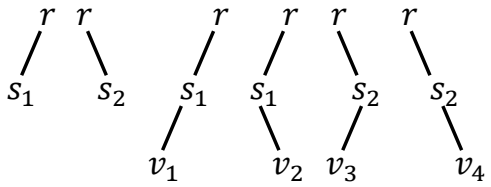
任意の正整数 M に対し，$[M] \coloneqq \{1, \ldots, M\}$ とする．今，N 文からなる文書を要約することを考える．まず，各文に対して，[3, 4] などの方法を用いてチャンク間の依存関係を表した木構造（以下，チャンクの依存構造木）を構築する．各 $i \in [N]$ について，第 i 文のチャンクの依存構造木を $G_i = (V_i, E_i)$ と書き，その根を $s_i \in V_i$ と書く．この時，$V_i \subseteq V$ が第 i 文のチャンクの集合となり，E_i の各枝がそれらの依存関係を表す．さらに，文書全体に対する木構造 $T \coloneqq (\{r\} \cup V, E)$ を定義する．ただし，r は便宜上導入されたダミーの根であり，V, E は以下のように定義される．

$$V \coloneqq V_1 \cup \cdots \cup V_N,$$
$$E \coloneqq \{E_1 \cup \{(r, s_1)\}\} \cup \cdots \cup \{E_N \cup \{(r, s_N)\}\}.$$

ここで，V が要約すべき文書のチャンク全体の集合と一致していることに注意する．与えられた要約 $A \subseteq V$ に対し，$A \cup \{r\}$ が T 上で r を根とする部分木を成しているならば，$A \cap V_i$ は G_i 上で s_i を根とする部分木



(a) 文書の木構造 $T = (V, E)$



(b) パスの集合 $\mathcal{P}$

図 1: (a) はチャンクの依存構造木 G_1, G_2 とダミーの根 r からなる，文書に対する木構造 T．(b) は T から得られる r と各頂点を結ぶパスの集合 $\mathcal{P}$．

になるため，各文におけるチャンクの依存関係が保存されていることになる．そこで以下では，T 上の r を根とする部分木を探すことで，要約を得ることを考える．また，c_v をチャンク $v \in V$ の長さ（文字数や単語数）とする．最終的に出力される要約 $A \subseteq V$ の長さは，ある一定の値 C 以下でなければならない．以上より，圧縮型要約は次のような制約付き劣モジュラ最大化として定式化できる．

$$\begin{aligned} &\underset{A \subseteq V}{\text{maximize}} && f(A) && \text{(問題 1)} \\ &\text{subject to} && \sum_{v \in A} c_v \leq C, \\ & && A \cup \{r\} \text{ は } T \text{ 上の根付き部分木.} \end{aligned}$$

上記の問題は T 上の部分木制約を保つため，単純な予算制約付き劣モジュラ最大化 [8] に比べて複雑な問題となっている．そのため，単縦な手法（例えば，根を始点として貪欲に頂点を順次追加する，など）では精度保証のある解を得ることができない．

3 提案法

提案法では，問題 1 を劣モジュラ予算制約付き劣モジュラ最大化問題 [7] として定式化し直す事で，部分木制約を直接扱うことを回避する．こうして得られた新たな問題に対する貪欲法を考えることで，理論保証付きの高速な圧縮型要約法を得る．

3.1 問題の再定式化

P を $v \in V$ と r をつなぐ T 上の全てのパス集合とする．ここで，$v \in V$ と $p \in P$ には一対一対応の関係があり，$|V| = |P|$ となることに注意する．また，$V_p \subseteq V$ を p に含まれる頂点の集合として定義し，任意の $X \subseteq P$ に対して $V_X \coloneqq \bigcup_{p \in X} V_p$ とする．これらの定義を用いると，問題 1 は以下のようなパス集合上の劣モジュラ最大化問題として定式化できる．

$$\begin{aligned} &\underset{X \subseteq P}{\text{maximize}} && F(X) \coloneqq f(V_X) && \text{(問題 2)} \\ &\text{subject to} && L(X) \coloneqq \sum_{v \in V_X} c_v \leq C. \end{aligned}$$

上記の P 上の集合関数 F, L は劣モジュラ性を持つため，この問題は劣モジュラ予算制約付き劣モジュラ最大化問題と見做すことができる．

Algorithm 1 貪欲法

```
Q ← P, X ← ∅
while Q ≠ ∅ do
    p = argmax_{p'∈Q} F({p'}|X) / L({p'}|X)
    if L(X + {p}) ≤ C then
        X ← X + {p}
    end if
    Q ← Q − {p}
end while
p̂ = argmax_{p'∈P} F({p'})
return Y = argmax_{X'∈{X,{p̂}}} F(X')
```

3.2 貪欲法

本節では，問題 2 に対する貪欲法を考える．以下では，P 上の和集合，差集合をとる演算を $+$，$-$ で表す．さらに，任意の $X, Y \subseteq P$ に対して $F(X \mid Y) \coloneqq F(X + Y) - f(Y)$，$L(X \mid Y) \coloneqq L(X + Y) - L(Y)$ と定義する．

問題 2 に対する貪欲法を Algorithm 1 に示す．Algorithm 1 は概ね通常の予算制約付き劣モジュラ最大化に対する貪欲法 [8] と同じであるが，Step 4 で $\frac{F(\{p\}|X)}{L(\{p\}|X)}$ を最大化する p を見つける必要がある点が，従来の方法と異なる．すなわち，$L(\cdot)$ が単なるコストの線形和ではなく，集合関数として表されることを考慮した貪欲法となっている．実装の際には，[8] 記載のものと類似した方法を用いて目的関数値の評価回数を削減する．この手法によって $Y \subseteq P$ が得られれば，$V_Y \subseteq V$ が問題 1 に対する解となる．

劣モジュラ予算制約付き劣モジュラ最大化に対する既存法 [7] は単なる貪欲法ではなく，より計算コストのかかる手法を用いている．一般の劣モジュラ予算制約付き劣モジュラ最大化に対しては，貪欲法の近似保証は現状知られていないが，問題 2 に対しては，問題の特殊性を利用することで，次のような近似保証を得ることができる．

定理 1. 各 $i \in [N]$ に対し，G_i の葉の数を ℓ_i とし，$\ell \coloneqq \max_{i\in[N]} \ell_i$ とする．$Y \subseteq P$ を Algorithm 1 の出力とし，$X^* \subseteq P$ を 問題 2 の最適解とすると，$F(Y) \geq \frac{1}{2}(1-e^{-1/\ell})F(X^*)$ が成り立つ．

r から各 $v \in V$ への枝を持つスターグラフ T を考えることで，問題 1 は予算制約付き単調劣モジュラ最大化問題を特殊ケースとして含んでいることがわかる．この問題の場合 $\ell = 1$ となるため，上記の結果は予算制約付き単調劣モジュラ最大化に対する $\frac{1}{2}(1-1/e)$ 近似 [8] を特殊ケースとして含んだものとなっている．

4 目的関数

[11] にある通り，様々な要約生成のための目的関数や要約の評価関数は，単調性と劣モジュラ性を持つ．本稿では，提案した貪欲法により圧縮型要約生成と圧縮型オラクル生成を行う．以下，それぞれで用いる目的関数について説明する．

4.1 要約生成のための目的関数

与えられた文書から要約を生成するための目的関数として，次の 2 つを考える．

被覆関数： M を要約すべき文書に含まれる異なり語の数とし，それらは $j \in [M]$ によって番号づけられているとする．また，w_j（$j \in [M]$）を第 j 番目の単語の重みとする．要約 $A \subseteq V$ が与えられた時，それを評価する被覆関数は以下のように書ける．

$$f(A) \coloneqq \sum_{j=1}^{M} w_j z_j.$$

ただし，$z_j \in \{0,1\}$ は j 番目の単語が A に含まれているか否かを表す二値変数である．

報酬付き被覆関数： 上記の被覆関数を目的関数として用いた場合，得られる要約は過度に圧縮された多数の文からなる傾向があり，結果として可読性の低下を招いてしまう．そのような傾向への対策として，より文の数が少ない要約に対して報酬を与えるような目的関数が，[12] によって提案されている．具体的には，要約 A が与えられた時，報酬付き被覆関数は以下のように書ける．

$$f(A) \coloneqq \sum_{j=1}^{M} w_j z_j + \gamma \left(\sum_{v\in A} c_v - \sum_{i=1}^{N} b_{s_i} \right).$$

ただし，$b_{s_i} \in \{0,1\}$ は第 i 文が A に含まれるか否かを表す二値変数であり，$\gamma \geq 0$ は被覆度と報酬値の割合を調整するパラメーターである．

4.2 オラクル生成のための目的関数

要約の評価関数を最大化することで得られた要約は，オラクル要約と呼ばれる [5, 6]．オラクル要約を求めることは，要約システムの性能の限界の把握や，システムの訓練データの作成において必要不可欠である．本稿の実験では，オラクル要約生成のための要約の評価関数として ROUGE を用いる（ROUGE の単調劣モジュラ性については [11] 参照）．

ROUGE： ROUGE [9] は要約評価において幅広く使われる指標であり，人手評価と高い相関を持つことが知られている．具体的には，$R_1, \ldots, R_K \subseteq V$ を人手によって得られた要約（参照要約）とし，$n_e(A)$ を n-gram e が要約 $A \subseteq V$ に表れた回数を返す関数とすると，ROUGE$_n$ は以下のように書ける．

$$f(A) \coloneqq \frac{\sum_{k=1}^{K} \sum_{e\in R_k} \min\{n_e(S), n_e(R_k)\}}{\sum_{k=1}^{K} \sum_{e\in R_k} n_e(R_k)}.$$

5 実験

提案法を要約タスクに適用し性能を評価する．また，比較手法として ILP に基づく手法を用いる．目的関数は上述の 3 つを用い，それぞれの場合について，達成された近似比，ROUGE$_1$ スコア，実行時間を比較する．

5.1 実験設定

DUC-2004 をデータセットとして用いて実験を行う．各文のチャンクの依存構造木は次のようにして構築する．まず，Stanford parser [2] によって単語間の依存関係を求める．それらの関係をもとに，Filippova らの手法 [3, 4] を用いてチャンクの依存構造木を構築する．また，得られる要約の可読性を高めるため，特定の関係を持つチャンクたちは 1 つの頂点としてまとめて扱う．さらに，c_v を $v \in V$ の単語数，$C = 100$ とし，100 単語以下の要約を作成を目指す．被覆関数で用いられる単語重み w_j（$j \in [M]$）は，DUC-2003 のデータセットを用いてロジスティック回帰 [13] によって計算する．また，報酬付き被覆関数のパラメーター γ は 0.9 とする．

実験は CPU: Intel Xeon E5-2620 v4 2.10GHz 32GB RAM を用いて行う．提案法は C++ で実装し `GCC version 4.8.5` を用いてコンパイルする．ILP 法は `CPLEX ver. 12.5.1.0` を用いて整数線形計画問題を解く．

表 1: 近似比，ROUGE_1，実行時間の比較結果．各数値は DUC-2004 に含まれる 50 トピック間での平均値．

目的関数	手法	近似比	ROUGE_1	実行時間
被覆関数	提案法	0.964	**0.347**	**1.34 (ms)**
	ILP	**1.00**	0.346	231 (ms)
報酬付き被覆関数	提案法	0.967	**0.334**	**1.44 (ms)**
	ILP	**1.00**	0.332	552 (ms)
ROUGE_1	提案法	0.985	0.468	**0.759 (ms)**
	ILP	**1.00**	**0.494**	92.1 (ms)

5.2 実験結果

表 1 に実験結果を示す．近似比については，提案法は常に 95% 以上を達成している．DUC-2004 のデータセットではチャンクの依存構造木の葉の数の最大値 ℓ は約 22 であり，定理 1 の近似保証の値は 2.2% 程度となる．すなわち，提案法の経験的な近似精度は，理論的に保証される値よりもはるかに良くなる傾向がある．この乖離は，理論上の近似比が最悪ケースの入力を想定して導出されているのに対し，現実的な問題の入力は最悪ケースからは程遠いことに起因している．ROUGE_1 スコアについては両手法とも同程度の値を達成している．実行時間については，目的関数が被覆関数，報酬付き被覆関数，ROUGE_1 それぞれの場合について，提案法は ILP 法よりも 約 170 倍，380 倍，120 倍高速である．特に，目的関数が報酬付き被覆関数の場合は，ILP 法の速度が著しく低下するのに比べ，提案法では殆ど低下していない．以上の結果をまとめると，提案法は目的関数値や ROUGE_1 スコアの面では ILP 法と同程度の性能を発揮しつつ，実行時間に関してはおよそ 100 から 400 倍高速に動作する．

6 おわりに

本稿では，劣モジュラ最大化に対する貪欲法を用いた圧縮型要約法を提案した．提案法は，（１）任意の単調劣モジュラ関数に適用可能であり，（２）貪欲法によって高速に要約を得ることができ，（３）理論的に近似精度を保証することができる，という３つの利点を持つ．実験では，提案法が ILP 法と同程度の目的関数値や ROUGE_1 スコアを達成しつつ，100 から 400 倍程度高速に動作することを確認した．

参考文献

[1] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 481–490. ACL, 2011.

[2] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pp. 449–454. European Language Resources Association, 2006.

[3] Katja Filippova and Yasemin Altun. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1481–1491. ACL, 2013.

[4] Katja Filippova and Michael Strube. Dependency tree based sentence compression. In *Proceedings of the 5th International Natural Language Generation Conference*, pp. 25–32. ACL, 2008.

[5] Tsutomu Hirao, Masaaki Nishino, and Masaaki Nagata. Oracle summaries of compressive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 275–280. ACL, 2017.

[6] Tsutomu Hirao, Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. Enumeration of extractive oracle summaries. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 386–396. ACL, 2017.

[7] Rishabh Iyer and Jeff Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pp. 2436–2444. Curran Associates Inc., 2013.

[8] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 420–429. ACM, 2007.

[9] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization Branches Out*, pp. 74–81, 2004.

[10] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 912–920. ACL, 2010.

[11] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520. ACL, 2011.

[12] Hajime Morita, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Subtree extractive summarization via submodular maximization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1023–1032. ACL, 2013.

[13] Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1776–1782. Morgan Kaufmann Publishers Inc., 2007.