抽出型オラクルを利用した要約の自動評価

平尾努　上垣外英剛　永田昌明
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
{hirao.tsutomu,kamigaito.hidetaka,nagata.masaaki}@lab.ntt.co.jp

1　はじめに

現在の要約自動評価のデファクトスタンダードは，ROUGE [8] や Basic Elements (BE) [4] であり，要約を N グラムや依存構造木における主辞と修飾語のタプルといった小さなユニットの集合としてとらえ，人間が用意した参照要約とシステム要約との間でこれらのユニットがどの程度一致するかを評価する．

これらの手法は人間の評価結果との相関が高く優れた自動評価法であるが，要約システムのエラー分析に向いているとはいえない．なぜなら，多くの要約システムが要約を生成する際に原文書から抽出するユニットと評価の際に使用する N グラムやタプルというユニットとの間には粒度という点で大きな違いがあるからである．要約システムは「文」や「根付き部分木」という意味を構成する単位を抽出することに対し，ひとつひとつの N グラムやタプルはそれ単独で意味を構成するものではない．よって，ある N グラムの抽出に成功あるいは失敗したことが，要約中で意味のある構成単位の抽出に成功あるいは失敗したことに直接はつながらない．

2000 年代はじめまでの要約システムの多くは文抽出によるシステムであったため，参照要約も人間が自由に生成した要約ではなく，原文書中の文を人手で抽出することで作成されることが多かった．よって，システムの評価もいわゆる分類問題と同様に F 値や正解率 [10, 2]，またこれらを改良した relative utility [11] スコアなどが用いられた．こうした評価指標を用いる場合，システムが要約生成の基本とするユニットと評価で利用するユニットが同じであるため，システムのエラー分析は比較的容易である．しかし，参照要約を文抽出で生成するにはコストがかかる．特に，複数文書要約のように原文書中の文の数が数百以上に及ぶ場合には現実的ではない．

本稿では，ROUGE や BE といった自動評価指標を最大化する原文書中の有意味なユニット (文や節) の組合せである抽出型オラクル要約 [3] を利用した要約自動評価法を提案する．多くの要約システムは原文書中の文やその一部分をユニットして要約を生成するため，システムが抽出したユニットがオラクルに含まれるか否かで容易に要約を評価できる．また，システムが抽出に成功あるいは失敗した有意味なユニットが容易にわかるのでシステムのエラー分析も容易になる．

なお，抽出型オラクルを用いれば，従来の F 値や正解率を用いることが可能となる．しかし，本稿では複数の参照要約が与えられることを活かし，オラクル中のユニットをピラミッド [9] を模した疑似ピラミッドにより重み付けし，その重みを考慮したうえでシステム要約を評価する．

2　抽出型オラクル要約

オラクル要約とはある要約の評価関数を最大化するシステム要約であり，以下の式で定義される．

$$\begin{aligned} \mathcal{O} &= \arg\max_{S \subseteq \mathcal{S}} f(R, S) \\ &\text{s.t.} \quad \ell(S) \leq L \end{aligned} \tag{1}$$

$\mathcal{S}$ は原文書から得たユニット (文や節) の集合，S はその部分集合である要約，R は参照要約であり，関数 f は ROUGE や BE に代表される要約の自動評価関数である．$\ell(\cdot)$ は，要約の長さを返す関数であり，L はあらかじめ与えられる要約長である (単語数が用いられることが多い)．なお，ROUGE や BE は，任意の要約とそれに対応する参照要約集合を引数とする関数として定義されているが，本稿ではユニットの重み付けのため，任意の要約とひとつの参照要約を引数とする関数として扱う．

評価関数 f として，ROUGE$_n$，BE を用いる場合，オラクル生成は整数計画問題として定式化され，MIP ソルバを用いればオラクルを得ることができる．詳細は文献 [3] を参照されたい．

3 抽出型オラクルを用いた自動評価法

抽出型オラクルは原文書のユニットの組合せであり，多くの要約システムも原文書の何らかのユニットを抽出して要約を生成している現状では，システム要約中のユニットがオラクルに含まれているか否かでシステムを評価することが可能となる．ただし，本稿では以下に詳述するとおり，オラクル中のユニットに対して重みを付けたうえで評価スコアを決定する．

3.1 疑似ピラミッドによるユニットの重み付け

一般的に独立に生成した複数の参照要約で言及されている内容はより重要であると考えることは自然であろう．人間が要約を評価をする際に用いられるピラミッド [9] はこの考えに基づいており，複数の参照要約に出現する内容 (Summary Content Unit:SCU) に重みを高く与える．たとえば，K 個の参照要約に出現する SCU の重みは K となることに対し，ひとつの参照要約にしか出現しない SCU の重みは 1 となる．しかし，SCU を機械的に性能良く文から抽出し，同じ意味をもつ SCU をまとめあげ頻度を数えることは現状では困難である．

一方，参照要約から得た抽出型オラクルは原文書中のユニットの集合であるため複数のオラクル間で同一のユニットがあるかどうかは一目瞭然である．よって，オラクル中のユニットを SCU と考え，そのスコアをピラミッドスコアの考えに基づき決定する．いま，K 個のオラクルが与えられたとき，ユニット o_j に対するスコア w_j を以下の式で定義する．

$$w_j = N(o_j) \tag{2}$$

$N()$ は，o_j が何個のオラクルに出現したかを返す関数である．つまり，オラクル中のユニットは最大 K，最小 1 のスコアをとる．これは，原文書中のすべてのユニットに対し，0 から K までのスコアを与えることであり[1]，Radev らによる relative utility スコアの一種とみなすこともできる．

[1] どのオラクルにも選択されなかったユニットのスコアは 0 と考えれば，原文書中のすべての文に対してスコアが割り当てられるため，粗い relative utility と考えることが可能である．

3.2 ユニットの対応関係の決定

いま，システム要約から得たユニット集合を U，u_i をその要素とし，K 個のオラクルから得たユニット集合を O，o_j をその要素とする．基本的には u_i について O のメンバであるか否かを判定し，真となった数を評価スコアとすればよい．しかし，原文書に冗長性があると U の要素には全く同じ，あるいは非常によく似たユニットが含まれることがある．この場合，単純にスコアを計算すると冗長な要約が過剰に高いスコアを獲得してしまう．この問題を解決するため，U の要素と対応する O の要素はただひとつだけという制約のもと，以下の割当問題を解くことで評価スコアを決定する．

$$\begin{aligned} \text{maximize} & \sum_{i=1}^{|U|}\sum_{j=1}^{|O|} g(u_i, o_j) w_j x_{i,j} \\ \text{s.t.} & \sum_{j=1}^{|O|} x_{i,j} = 1 \quad \forall i \\ & \sum_{i=1}^{|U|} x_{i,j} \leq 1 \quad \forall j \\ & x_{i,j} \in \{0,1\} \quad \forall i,j \end{aligned} \tag{3}$$

なお，関数 $g(,)$ は以下の式で定義する．

$$g(u,o) = \begin{cases} 1 & \text{sim}(u,o) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$g(,)$ は u が O の要素であるかどうかを判定する関数なので単純には u と O の要素が完全に一致するかどうかを判定すればよい．しかし，システムが前処理段階でトークンを書き換えたり，文圧縮などを行うことを考慮すると完全一致による判定は厳しすぎる．よって，u と u の類似度を計算し，それがある閾値 τ 以上である場合に u が O の要素であると判定する．

式 (3) の 1 番目の制約は u_i 対して割り当てられるオラクルのユニット o はただひとつであることを保証し，2 番目の制約は o_j に割り当てられるシステムのユニット u は最大でひとつであることを保証する．この割当問題は MIP ソルバ，ハンガリアンアルゴリズム [5] を用いることで解くことができる．

3.3 疑似ピラミッドスコア

式 (3) の割当問題を解くことでシステム要約のスコアを得ることができる．i 番目のシステムのユニット

表 1: 実験データの詳細

年	人手評価	要約長	トピック	参照要約	システム
2003	Coverage	100	30	4	16
2004	Coverage	100	50	4	17
2006	Pyramid	250	20	4	22

に対応するオラクルのユニットのインデックスを $a(i)$ とすると，擬似ピラミッドスコアは，以下の式で定義できる．

$$\textsc{PseudoPyramid}(U, O) = \sum_{i=1}^{|U|} g(u_i, o_{a(i)}) w_{a(i)} \quad (5)$$

しかし，このスコアは正規化されていないため，異なるデータセット間で比較するには不適切などの問題がある．そこで，このスコアをユニット抽出による要約が取りうる最大スコアで割ることで正規化する．オラクル中のユニット o_j には長さ (単語数) とスコア w_j が与えられている．よって，参照要約を同じ長さ制約のもとでスコアを最大となるユニットの組合せをみつければよい．これは，ナップサック問題であり以下の整数計画問題として定式化される．

$$\begin{aligned} &\text{maximize} \sum_{j=1}^{|O|} w_j z_j \\ &\text{s.t.} \sum_{j=1}^{|O|} \ell(o_j) \le L. \quad \forall j \\ &z_j \in \{0, 1\} \quad \forall j \end{aligned} \quad (6)$$

z_j は o_j を選択するか否かをあらわる 0/1 変数，w_j は 3.1 節で定義した o_j の重みである．上記問題の最適解を $\textsc{Upper}(O)$ とし，擬似的ピラミッドスコアを以下の式で再度定義する．

$$\textsc{PseudoPyramid}(U, O) = \frac{\sum_{i=1}^{|U|} g(u_i, o_{a(i)}) w_{a(i)}}{\textsc{Upper}(O)} \quad (7)$$

4 検証実験

4.1 実験設定

実験には DUC (Document Understanding Conference) 2003，2004，2006 のデータセットを用いた．それぞれのデータセットの詳細を表 4.1 にまとめる．DUC-2006 に関しては人手評価指標として Responsiveness も採用しているが，本稿では Pyramid のみを対象とする．DUC-2003，2004 と DUC-2006 の大きな違いは要約の単語数制限である．前者が 100 単語に対し後者は 250 単語で 2 倍以上の違いがある．また，DUC-2003，2004 の人手評価指標には Coverage スコア，DUC-2006 では Pyramid スコアが採用されている．参照要約はすべてのデータセットで 4 である．トピック数は DUC-2004 が最大で 50，DUC-2006 が最低で 20 である．システム数は DUC-2006 がやや多く 22 で，DUC-2003，2004 は 16，17 でほぼ同数である．

なお，式 (4) の sim(,) は，以下の式で定義した．

$$\text{sim}(u, o) = \frac{\text{LCS}(u, o)}{\ell(o)} \quad (8)$$

LCS は，u と o の間の最長共通部分列を返す関数である．sim(,) に対する閾値 τ は 0.6 に設定した．

また，疑似ピラミッドを生成する際に必要となるオラクルは $\textsc{Rouge}_2$，BE の双方を試し，比較対象としてもこれらを用いた．オラクル生成および評価の基本ユニットには，節相当である Elementary Discourse Unit (EDU) を用いた．文献 [7] では，EDU と SCU の類似性が指摘されており，基本ユニットを EDU に設定することは自然であろう．

なお，RST Discoursee Treebank [1] のデータを用いて，文字，単語列を入力とする両方向 LSTM [6] に基づき貪欲法で探索を行う EDU セグメンタを訓練し，原文書中の文，システム要約の文を EDU へと分割した．

4.2 結果と考察

実験結果を表 2 に示す．表 2 より，DUC-2003，2004 では評価関数として BE を採用した提案手法が非常に高い相関を獲得した．提案法は Rouge や BE に基づき生成した抽出型オラクルとシステム要約との間で一致する EDU の数に基づきシステム要約を採点する．オラクル要約が Rouge，BE を最大化する EDU の組合せであることを考慮すると Rouge，BE 相当の相関が得られることは妥当な結果である．

一方，DUC-2006 では相関は Rouge，BE よりも劣る結果となった．この原因は EDU セグメンタの質の低下に起因するオラクルの質の低下にあると考える．RST Discourse Treebank における EDU 境界認定の F 値は 0.9 程度である．RST Discourse Treebank は The Wall Street Journal の一部であるため，異なる通信社のデータを利用している DUC のデータでも同等の性能が発揮されているとは限らない．DUC は New York Times, Associated Press などのデータを利用し

表 2: 自動評価法と人間の評価との間の相関．r はピアソンの積率相関係数，ρ はスピアマンの順位相関係数，τ はケンドールの順位相関係数を示す．PP(B) は BE によるオラクルを用いた提案手法，PP(R) は ROUGE$_2$ によるオラクルを用いた提案手法である．

		r	ρ	τ
2003	ROUGE$_2$	.872	.832	.667
	BE	.911	.856	.683
	PP(R)	.894	.800	.617
	PP(B)	**.944**	**.876**	**.733**
2004	ROUGE$_2$	.928	.814	.662
	BE	.934	.863	.721
	PP(R)	.912	.806	.647
	PP(B)	**.951**	**.931**	**.809**
2006	ROUGE$_2$	**.882**	**.874**	**.714**
	BE	.883	.837	.680
	PP(R)	.765	.763	.576
	PP(B)	.864	.817	.645

ており，年によってそれも異なる．よって，DUC-2006 ではライティングスタイルの違いなどによりセグメンタの性能が特に劣化しているのではないかと考える．

また，PP(R) と PP(B) を比較すると明らかに PP(B) が PP(R) よりも良い．これもオラクルの質によるものと考える．EDU という文よりも小さな単位に分割すると参照要約に対してバイグラムの一致率が高くなるようにそれらを組み合わせると短い EDU が数多く選択され，情報が断片化されたオラクルとなってしまう．一方，BE では依存構造のタプルを一致するように EDU を選択するため，バイグラムのような情報の断片化は少ない．

これら実験結果から，提案法はオラクルを構成せするユニットの質に依存することがわかった．今後，EDU セグメンタの性能を汎用性をもたせたうえで向上させることが大きな課題であろう．

5 おわりに

本稿では，EDU を基本単位として，参照要約から抽出型オラクルを生成し，システム要約の EDU がオラクル要約に含まれる割合で評価する新しい自動評価法を提案した．DUC-2003，2004，2006 のデータセットを用いて提案手法の有効性を評価したところ，DUC-2003，2004 において提案手法は従来法より人間の評価結果との間の相関が高いことがわかった．しかし，DUC-2006 では従来法よりも劣る結果となった．

参考文献

[1] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proc. of the SIGDIAL01*, pages 1–10, 2001.

[2] Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. Extracting import sentences with support vector machines. In *COLING*, pages 342–348, 2002.

[3] Tsutomu Hirao, Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. Enumeration of extractive oracle summaries. In *EACL*, pages 386–396, 2017.

[4] Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. Automated summarization evaluation with basic elements. In *LREC 2006*, pages 889–902, 2006.

[5] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL/HLT*, pages 260–270, 2016.

[7] Junyi Jessy Li, Kapil Thadani, and Amanda Stent. The role of discourse units in near-extractive summarization. In *SIGDIAL*, pages 137–147, 2016.

[8] Cin-Yew. Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. of Workshop on Text Summarization Branches Out*, pages 74–81, 2004.

[9] Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL 2004*, pages 145–152, 2004.

[10] Miles Osborne. Using maximum entropy for sentence extraction. In *ACL-02 Workshop on Automatic Summarization*, pages 1–8, 2002.

[11] Dragomir R. Radev and Daniel Tam. Summarization evaluation using relative utility. In *CIKM*, pages 508–511, 2003.