テキストセグメンテーションによる教師なし修辞構造解析

小林尚輝† 平尾努§ 中村健吾§ 上垣外英剛† 奥村学† 永田 昌明§
†東京工業大学 §NTT コミュニケーション科学基礎研究所

1 はじめに

文書を木構造として表現する修辞構造理論 (Rhetorical Structure Theory)[11] では，隣接するテキストスパン (節相当の単位である談話基本単位，Elementary Discourse Unit (EDU) そのものや EDU 系列) 間の修辞関係の同定を階層的に繰り返すことでボトムアップに修辞構造木と呼ばれる木を構築する．こうした木を自動的に得るための修辞構造解析技術は自然言語処理の重要な基盤技術であり，文書要約 [7]，評判分析 [2]，文書分類 [6] などの様々な応用タスクでその有効性が示されている．

一般的に修辞構造解析器は，教師有り機械学習アルゴリズムを利用して実装され，木構造のアノテーションが与えられたコーパスを訓練データとしてパラメタを学習する．近年，shift-reduce 法において，アクションを決定するための重みパラメタを訓練データから学習した解析器が良い性能を達成している [10, 12]．しかし，同じ木構造を構築する構文解析タスクとは異なり，修辞構造解析では，修辞構造木のアノテーションが与えられたコーパスは数少ないうえにそのサイズも小さい．最も大規模なコーパスである RST Discourse Treebank (RST-DTB)[3] でさえ 385 文書しかない．教師有り学習に基づく手法は訓練データへの依存性が強いため，性能向上のためにはより多くのアノテーション付きデータが必要であり，さらに，解析したい文書のドメインが訓練データと異なる場合には十分な性能が出ないという問題もある．

本研究ではこうした問題を解決するため，教師なし修辞構造解析を提案する．修辞構造木が 2 分木として表現できること，つまり，テキストスパンを 2 分割することを繰り返すことで得られることに注目し，これをテキストセグメンテーションの問題と捉える．そして，**トップダウン**，**ボトムアップ**両方の解析法を提案する．トップダウン解析法は，分割の各ステップにおいて分割スコアが最大となる箇所でスパンを分割することを貪欲法で再帰的に行い，修辞構造木を構築する．ボトムアップ解析法は，修辞構造木全体での分割スコアが最大となるように動的計画法でボトムアップに木を構築する．RST-DTB を用いて評価実験を行ったところ，ボトムアップ解析法よりもトップダウン解析法の方が性能が良く，スパンスコアは約 0.8 を上回る結果を達成した．当然，既存の教師有り解析器よりもスコアは劣るものの十分高いスコアであり，その有効性が明らかとなった．

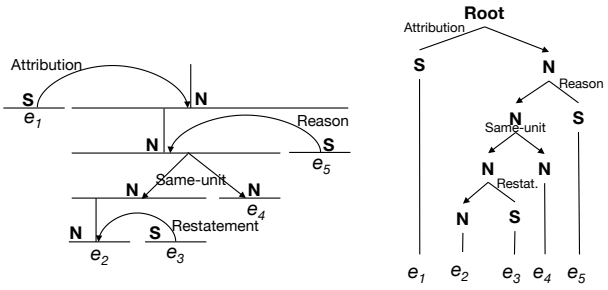

図 1: 修辞構造木の例

2 修辞構造木

修辞構造木は，それを構成する最小の談話基本単位 EDU の系列，つまり，テキストスパンを修辞関係により結合し，より大きなテキストスパンを構成するという操作を再帰的に繰り返すことによって得られる木である．結合される 2 つのスパンの一方は重要な情報を持つ核 (Nucleus)，もう一方はそれを補足する衛星 (Satellite) となり，衛星から核への有向エッジに修辞関係ラベルが付与される．ただし，並列構造のような等価な関係を結ぶ場合には，2 つのスパンがともに核となる．図 1 左に WSJ_2363 の以下の文の修辞構造木を示す．

> [Interprovincial Pile Line Co. said]$_{e_1}$ [it will delay a proposed two-step, 830 million dollar]$_{e_2}$ [(US$705.6 million]$_{e_3}$ [expansion of its system]$_{e_4}$ [because Canada's output of crude oil is shrinking.]$_{e_5}$

たとえば，衛星である EDUe_3 が核である e_2 を Restatement という修辞関係で修飾し，結合されたスパン e_2, e_3 全体が核となる．修辞構造木は文脈自由文法を元にした句構造木，つまり，非終端記号が核か衛星を表すラベル，エッジが修辞関係，終端記号が EDU という句構造木とみなすことができる (図 1 右)．

3 テキストセグメンテーションによる修辞構造解析

図 1 から明らかなように修辞構造解析は，テキストスパン (EDU 系列) を再帰的に 2 分割することで得ることができる．例では，スパン $e_{1:5}$ を e_1 と $e_{2:5}$ に分割し，$e_{2:5}$ をさらに $e_{2:4}$ と e_5 に分割している．こうしてテキストスパンを分割していき，最終的に分割したスパンすべてが EDU そのものになった時点で手続きを停止すれば修辞構造木が構築される．本節では，このスパンの分割にテキストセグメンテーション法を利用した教師なし修辞構造解析について述べる．なお，本研究では教師なしで木構造を構築するため，スパンの核，衛星の推定，スパン間の修辞関係の推定は行わないことに注意されたい．

3.1 TextTiling

TextTiling[4] はもっとも単純なテキストセグメンテーション法の 1 つであり，文書を章，節，段落といった，より小さなセグメントへと分割する手法である．TextTiling では，文書を単語の系列 $w_{1:N}$ として，k 番目の単語 w_k の直後がセグメントの境界であるかどうかを前後の文脈から決定する．

具体的には，(1) 幅（単語数）b の窓を用いて前後の文脈に当たる範囲を表現し，その類似度を計算する．左右の窓は，それぞれに含まれる単語頻度に基づきベクトルとして表現される（ここではそれぞれ $\overrightarrow{w_{k-b:k}}$, $\overrightarrow{w_{k+1:k+b}}$ とする）．この 2 つのベクトルのコサイン類似度を求める，(2) k を 1 から $N-1$ まで動かし，類似度が極小かつ閾値以下となる単語の直後で文書を分割する．

修辞構造理論では文書の最小構成単位が EDU であるため，TextTiling における単語を EDU とみなせば同様に EDU 系列を分割することができる．ただし，複数の単語からなる EDU をベクトルとして表現するため，単語頻度に基づく高次元でかつスパースなベクトルを採用すると類似度がすべて似た値となってしまい，問題となる．そこで，EDU e のベクトル表現は EDU 中の単語の分散表現 $\overrightarrow{w}$ の加重平均で表現し，SIF[1] を用いて式 (1) によって求める．

$$\overrightarrow{e} = \sum_{w \in e} \frac{a}{p(w) + a} \overrightarrow{w} \tag{1}$$

ここで $p(w)$ は単語の出現確率であり，パラメタ a は式 (2) で計算され，頻出単語の重みを減少させる．

$$a = \frac{1 - \alpha}{\alpha Z} \tag{2}$$

α, Z はそれぞれハイパーパラメタの定数と総単語数である．

単語の分散表現 $\overrightarrow{w}$ は，ELMo[9] と GloVe[8] で得た分散表現を結合したものを用いる．

$$\overrightarrow{w} = [\overrightarrow{w}_{\mathrm{E}}; \overrightarrow{w}_{\mathrm{G}}] \tag{3}$$

3.2 トップダウン修辞構造解析

任意のテキストスパンに TextTiling 法を適用し，類似度が最小となる EDU の直後でスパンを分割することを貪欲法で再帰的に繰り返すことで修辞構造木を得る．ただし，文書には段落，文という階層構造があらかじめ与えられていることに着目し，本研究では文書を段落系列，段落を文系列，文を EDU 系列として捉える．そして，各階層について独立に系列の 2 分割を再帰的に繰り返し，木を得る．ここで，段落，文のベクトル表現はそれらに含まれる EDU ベクトルの平均とする．

本研究で提案する修辞構造解析手法では，TextTiling と同様，候補となるスパン（EDU，文，段落）に対して，その左右に幅 b の窓を設けて類似度を計算し，もっとも類似度が低くなるスパンの直後で分割を行う．i 番目の EDU から j 番目の EDU までで構成されるスパンにおいて k 番目の EDU($i \leq k < j$) の直後が分割の候補である場合，その左右の文脈を表すスパンベクトル $\overrightarrow{s_\ell}, \overrightarrow{s_r}$ は，幅 b をパラメタとして以下の式で定義される．

$$\overrightarrow{s_\ell} = \begin{cases} \overrightarrow{e_{k-b:k}} & (k - i > b) \\ \overrightarrow{e_{i:k}} & (\text{otherwise}) \end{cases}$$

$$\overrightarrow{s_r} = \begin{cases} \overrightarrow{e_{k:k+b}} & (j - k > b) \\ \overrightarrow{e_{k:j}} & (\text{otherwise}) \end{cases}$$

これを用いて $\overrightarrow{s_\ell}$ と $\overrightarrow{s_r}$ の類似度を以下の式で定義する．

$$\mathrm{Sim}_{\cos}(\overrightarrow{s_\ell}, \overrightarrow{s_r}) = \frac{\overrightarrow{s_\ell} \cdot \overrightarrow{s_r}}{\|\overrightarrow{s_\ell}\| \|\overrightarrow{s_r}\|} \tag{4}$$

なお，スパン $e_{i:j}$ のベクトル表現 $\overrightarrow{e_{i:j}}$ は，式 (1) により求まる EDU ベクトルを平均したベクトルとして式 (5) に従って求める．

$$\overrightarrow{e_{i:j}} = \frac{1}{j-i} \sum_{\hat{i} \in (i,\ldots j)} \overrightarrow{e_{\hat{i}}} \tag{5}$$

3.3 ボトムアップ修辞構造解析

貪欲法では分割の各ステップで左右の文脈の類似度が最小となる EDU でスパンを分割するため，木全体での分割の最適性は保証されない．そこで，分割の最適性を保証するため，すべての可能な木の組合せの中から総分割スコアを最大化する修辞構造木を動的計画法を用いて得る．

任意のスパン $e_{i:j}$(i 番目の EDU から j 番目の EDU までの系列) を $k(i \leq k \leq j-1)$ 番目の EDU の直後で分割することを考える場合，その分割スコアの最大値 $V[i][j]$ は i 番目の EDU から k 番目の EDU で構成されるスパンの分割スコアの最大値 $V[i][k]$，k 番目の EDU と $k+1$ 番目の EDU の間でスパンが分割されるスコア $P_{\text{split}}(i,k,j)$，$k+1$ 番目の EDU から j 番目の EDU で構成されるスパンの分割スコアの最大値 $V[k+1][j]$ を用いて以下の式で定義される．

$$V[i][j] = \begin{cases} 1 & i = j \\ \max\limits_{i \leq k < j} V[i][k] \times P_{\text{split}}(i,k,j) \times \\ \qquad V[k+1][j] & \text{otherwise} \end{cases} \tag{6}$$

k 番目の EDU と $k+1$ 番目の EDU の間の分割スコア $P_{\text{split}}(i,k,j)$ は TextTiling と同様に左右の文脈の類似度に基づいて算出する．ここでは，「左右のスパンが意味的に分離しているなら，スパン間の類似度は小さくなる」と仮定し，以下の式で定義する．

$$P_{\text{split}}(i,k,j) = 1 - \{\text{Sim}_{\cos}(\overrightarrow{s_\ell}, \overrightarrow{s_r}) + 1\}/2 \tag{7}$$

文，段落，文書それぞれの階層に対して解析対象となるスパンを与え，三角行列を用いた CKY 構文解析アルゴリズムのように，DP テーブルをボトムアップに埋めていけば可能なすべての木から総分割スコアが最大となる修辞構造木を獲得できる．

4 実験

4.1 実験設定

提案法の有効性を検証するため，RST-DTB データセットを用いて評価実験を行った．RST-DTB は訓練セット 347 件，テストセット 38 件に分割されているが，開発セットが明示的に与えられていない．そこで，文献 [5] に従い，訓練セットのうち 40 件を開発セットとして用いた．提案法には窓幅 b，ELMo あり/なし，SIF によるベクトル算出時の主成分の除去の有無といったパラメタがある．これらは開発セットを用いてチューニングした．3 節で述べたとおり教師なし修辞構造解析は木の形しか推定することができないので，評価にはスパンスコアのマクロ，マイクロ F 値を用いた．スパンスコアは，正解の修辞構造木から得られるすべてのスパン，つまり，木のすべての非終端記号が支配するスパンの集合を G とし，推定した修辞構造木におけるすべてのスパンの集合を P として，精度 ($= |G \cap P|/|P|$)，再現率 ($= |G \cap P|/|G|$) を求め，F 値を計算する．また，参考として現在最も優れた修辞構造解析器 [10] のスコアも示す．

手法	w/o ELMo	w/o PC	窓幅 (b) 1	2	5	full
Topdown (Greedy)	√	√	.657	.665	.654	.652
	√		.636	.629	.609	.593
		√	.666	**.669**	.649	.651
			.653	.618	.592	.576
Hie. Topdown	√	√	.836	.837	**.838**	.833
	√		.826	.826	.821	.818
		√	.824	.822	.830	.831
			.826	.818	.818	.810
Bottomup (DP)	√	√	.600	.609	.604	.583
	√		.598	.621	.624	**.631**
		√	.599	.608	.601	.577
			.599	.602	.612	.619
Hie. Bottomup	√	√	.791	.800	.791	.790
	√		.795	**.803**	.796	.796
		√	.792	.796	.788	.787
			.793	.785	.769	.770
WLW17	−	−	.873			

表 1: 開発セットに対するマクロ F 値

4.2 結果と考察

表 1 に提案法と Wang らの方法 (WLW17) の開発セットにおけるスパンスコアを示す．表より，貪欲法，動的計画法とも段落，文という階層構造を考慮することでスコアが大幅に向上しており，修辞構造解析におけるこれらの情報の重要性がわかる．また，貪欲法と動的計画法を比較すると，貪欲法の方が良い結果となった．ELMo を利用するか否か，SIF において第一主成分を除去するか否かという点に関しては，双方ともに用いない方が良い傾向にあり，窓幅によるスコアの違いも大きくはないが 2 以上の場合が 1 よりは良い傾向にある．提案手法間を比較すると，階層構造を利

手法	macro F	micro F
Greedy	.648	.624
Hie. Greedy	.823	.799
DP	.615	.588
Hie. DP	.790	.767
WLW17	.881	.860

表 2: テストセットに対するマクロ，マイクロ F 値

用した貪欲法 (ELMo，主成分除去なし) のスコアが最も高く，0.838 である．

それぞれの手法に対して，パラメタを開発セットに基づき決定し，テストセットで評価した結果を表 2 に示す．開発セットでの結果よりスコアはやや低くなっているものの手法間の差に変化はない．階層構造を利用した貪欲法 (ELMo, 主成分除去なし) のスコアは 0.823 である．このスコアは，教師あり学習に基づく Wang らの手法より低いことは当然であるものの初期の教師あり学習を用いた解析器に近い．

動的計画法で得た木は分割の最適解であるにもかかわらず，貪欲法よりも劣る結果となった．これは，我々の直感に反する．本研究では，スパンを 2 分割する処理を再帰的に繰り返すことで修辞構造木が得られることに着目し，分割のスコアとして左右のスパンの分散表現に基づく類似度を利用した．しかし，この類似度は正しい修辞構造木を得ることに有望であるものの，それ単体で十分ではなかったことが原因と考えられる．たとえば，分散表現だけでなくその他の特徴も考慮してスパン間の類似度を計算できればより性能が向上し，動的計画法を用いる効果が発揮されると考える．

5 まとめ

本研究では，テキストスパンを 2 分割することを繰り返すことで修辞構造木が構築できることに着目し，貪欲法によりトップダウンに木を構築する手法と動的計画法によりボトムアップに木を構築する手法の，2 つの教師なし修辞構造解析手法を提案した．スパンの分割は，テキストセグメンテーション問題と捉え，TextTiling 法を改良した手法を適用した．RST-DTB データセットを用いて提案手法の性能を評価したところ，スパンスコアの F 値はトップダウン解析法の方が優れており，その値は 0.8 を上回った．この結果より，我々の教師なし修辞構造解析手法の有用性が示された．

参考文献

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *CoLR*, 2017.

[2] Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. Better document-level sentiment analysis from rst discourse parsing. In *EMNLP*, 2015.

[3] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *SIGDIAL*, 2001.

[4] Marti A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 1997.

[5] Michael Heilman and Kenji Sagae. Fast rhetorical structure theory discourse parsing. *CoRR*, 2015.

[6] Yangfeng Ji and Noah A. Smith. Neural discourse structure for text categorization. In *ACL*, 2017.

[7] Daniel Marcu. Improving summarization through rhetorical parsing tuning. In *Very Large Corpora Workshop*, 1998.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[9] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *ACL-HLT*, 2018.

[10] Yizhong Wang, Sujian Li, and Houfeng Wang. A two-stage parsing method for text-level discourse analysis. In *ACL*, 2017.

[11] William C. Mann and Sandra A. Thompson,. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, Vol. 8, No. 3, pp. 243–281, 1988.

[12] Nan Yu, Meishan Zhang, and Guohong Fu. Transition-based neural rst parsing with implicit syntax features. In *ICCL*, 2018.