疑似正解データを利用した修辞構造解析器の改善

小林 尚輝 † 平尾 努 § 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
{kobayasi@lr., kamigaito@lr., oku@}pi.titech.ac.jp
{tsutomu.hirao.kp, masaaki.nagata.et}@hco.ntt.co.jp

1 はじめに

文書はつながりのある言語単位（節や文，段落など）から構成されており，言語単位間の関係を明らかにすることを談話構造解析という．談話構造が計算機により自動的に解析できるようになれば，要約や翻訳など，文書を対象とした下流タスクの性能向上が期待できる．修辞構造理論 [1] は談話構造を表現する手法の一つであり，Elementary Discourse Unit (EDU) と呼ばれる節相当の談話単位を終端ノードとした句構造木として文書を表す．非終端ノードは連続する EDU から構成されるテキストスパン（以降，スパン）と対応し，子ノード間の核性（従属関係）ラベル，関係（修辞関係）ラベル[1)]が割り当てられる．多くの修辞構造解析器は，1) 木構造の推定 2) 隣接するスパンの核性ラベルの推定 3) 隣接するスパンの関係ラベルの推定の 3 つの副問題を教師あり学習を用いたモデルで解くことで実現される．

近年，ニューラルネットワークを用いることで修辞構造解析の性能が大きく改善されることが報告されている [2, 3, 4, 5]．スパンを再帰的に分割することでトップダウンに木を構築する Span Based Parser (SBP) [3] は，木の構造および核性ラベルの推定において高い性能を達成しているが，関係ラベルの推定にはまだ改善の余地がある．これは，18 種類の関係ラベルの分類を学習するにはデータサイズが不十分であることが原因である．現存する最大規模のコーパスである RST-DT [6] でさえ 385 文書しかなく，十分な規模とは言い難い．しかし，修辞構造を文書に付与するためには専門知識と時間を要するため，人手でデータを増やすことは容易ではない．

こうした問題を解決するため，ニューラル機械翻訳で提案された逆翻訳による疑似正解データの活用 [7] にヒントを得て，既存の修辞構造解析器を用いて自動的に作成された大規模な疑似正解データを用いて SBP を事前学習 (pre-training) し，RST-DT を用いて追学習 (fine-tuning) することで関係ラベルの推定性能を改善する枠組みを本稿では提案する．また，疑似正解データを大量かつ高品質に獲得するために，図 1 に示されるように複数の解析器が出力する木の間で重複する部分木を疑似正解データとして効率よく抽出するアルゴリズムを提案する．

RST-DT を用いた実験では，提案手法は SBP の関係ラベルの推定精度を大きく向上させ，核性と関係ラベルの両方を考慮した評価指標である Full において世界最高性能を達成した．

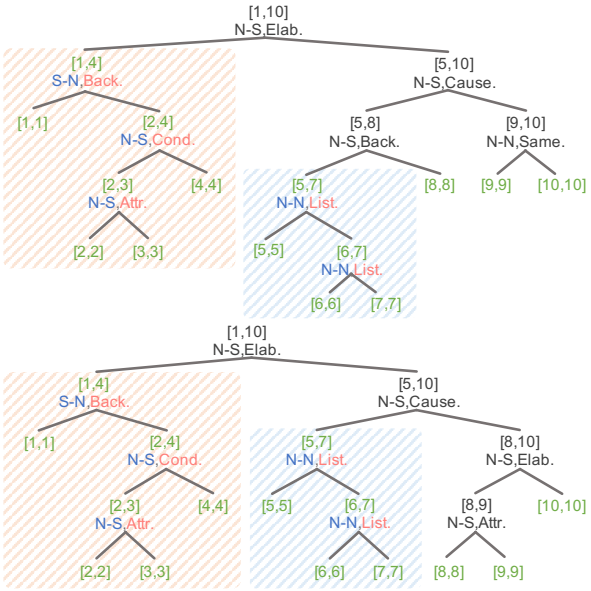

図 1 重複する部分木（網掛け部分）の抽出

2 関連研究

一般に修辞構造解析器は教師あり学習の枠組みで訓練され，解析方法としては遷移型とトップダウン型がある．遷移型では，人手で設計された特徴量を用いる手法 [8] と係り受け解析器から得られた特徴ベクトルを用いる手法 [2] がある．また，BERT の

1) 核性ラベルは Nucleus, Satellite の組み合わせからなる{N-S, S-N, N-N}の 3 通り，関係ラベルは 18 通り存在する．

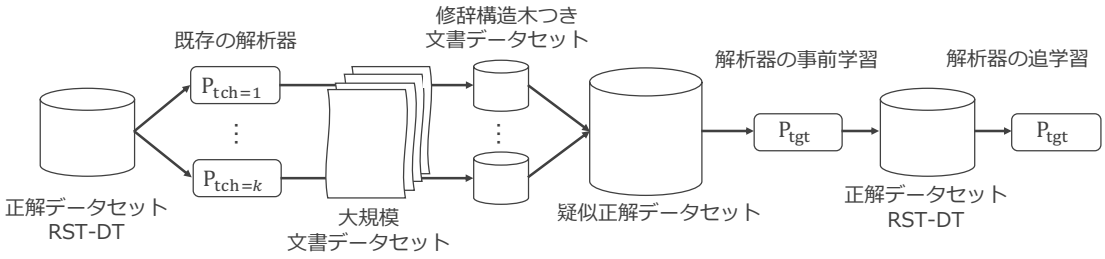

図 2　提案手法における学習の流れ

派生である SpanBERT により埋め込まれた特徴ベクトルを用いる手法 [5] は，現時点で最も良い性能を達成している．トップダウン型では，ポインターネットを利用してスパンを分割し木を構築する手法 [4, 9] とスパンを再帰的に 2 つに分割することで木を構築する手法 [3] がある．

修辞構造解析は RST-DT を正解データとして学習されるが，RST-DT に含まれる文書は 385 文書のみである．したがって，データ不足を解決するための手法が盛んに研究されており，修辞構造が付与された複数言語のデータセットを使う手法 [10, 11]，感情分類を学習したモデルを用いて修辞構造木を構築する手法 [12, 13]，修辞構造に近い特性をもつ複数のタスクを同時学習する手法 [14] がある．しかし，これらの手法は追加のコーパスを必要とする．追加のコーパスを必要としない手法として，既存の 2 つの修辞構造解析器から得られた木の間で合意率が一定のしきい値を超えた木を擬似データとして利用する手法 [15] がある．しかし，この手法の効果は低頻度ラベルに限定的である．

3　提案手法

ラベルなしデータセットに既存の分類器を用いてラベルを付与し，疑似正解データセットとして分類機を再学習する手法に Self-training や Co-training がある．ラベルを付与する際に，複数の分類器の出力間で一致を取ることで疑似正解データセットの質を保証するが，修辞構造解析が文書を対象とするタスクであるため，木の全体における一致が得られる文書は多くない．また，疑似正解データセットには，正解データセットに含まれるラベルの出現頻度における偏りがより強く反映される．

これらの問題点を解決するために提案法は，1) 複数の解析器の出力する木の間で重複する部分木を擬似正解データとすることで信頼性の高いデータを大規模に構築し，2) 疑似正解データで事前学習した解析器を正解データで追学習することで疑似正解データのラベルの偏りが引き起こす影響を緩和する．

提案法の概要を図 2 に示す．まず，正解データをもとに学習された複数の解析器を用いて修辞構造木つき文書データセットを得る．次に，得られた修辞構造木から，重複する部分木を抽出することで疑似正解データセットを得る．最後に，疑似正解データセットによる事前学習および正解データセットによる追学習により解析器を訓練する．

3.1　疑似正解データの獲得方法

複数の解析器が出力した修辞構造木の間で一致するすべての部分木を抽出する線形時間アルゴリズムを Algorithm 1 に示す．[2] 修辞構造木は入れ子状のラベル付きスパンとして表現され，２分木であるため非終端ノードは左右の子スパンを持つ．終端ノードはスパンの長さが 1 であることで判定される．

関数 Agree は与えられたスパンに含まれる部分木が重複した部分木であるかどうかを判定し，*state* にスパンをキーとしてその判定結果 c_{agree} を保持する．任意のスパンが表す部分木が目的の重複した部分木であるかどうかを確認するためには，そのスパンの左右の子スパンが共に重複した部分木であり，かつ，そのスパン自身が全ての修辞構造木に含まれることを確認すれば良い．左右の子スパンに対しては，再帰的に関数 Agree を適用することで，自身が全ての修辞構造木に含まれることは，スパンの出現頻度の辞書 *freq* の値と修辞構造木の数 k の一致により判定する．

次に，重複した部分木を関数 Extract によって抽出し，*subtrees* に追加する．深さ優先で木の探索を行い，各スパンに対して関数 Agree により求めた c_{agree} が真であればそのスパンを *subtrees* に追加し引き返す．これにより，包含関係にある部分木は最大の木のみが抽出される．また，抽出される木の大きさが

2)　ただし，包含関係にある部分木は最大の木のみを抽出する．

Algorithm 1: 重複した構造の判定

入力: 木のルートにあたるスパン *root*
修辞構造木の数 k
ラベル付きスパンの頻度 *freq*
抽出する木の大きさ $l_{\min}, l_{\max}$
出力: 部分木のリスト *subtrees*

```
state ← {}
def Agree(span):
    if Len(span) = 1 then
        return True
    c_left ← Agree(leftChild(span))
    c_right ← Agree(rightChild(span))
    if freq[span] = k then
        c_self ← True
    else
        c_self ← False
    c_agree ← c_left ∧ c_right ∧ c_self
    state[span] ← c_agree
    return c_agree
Agree(root)
subtrees ← []
def Extract(span):
    if Len(span) < l_min then
        return
    else if Len(span) > l_max then
        Extract(leftChild(span))
        Extract(rightChild(span))
    else
        c_agree ← state[span]
        if c_agree then
            Append(subtrees, span)
            return
        else
            Extract(leftChild(span))
            Extract(rightChild(span))
Extract(root)
```

$l_{\min}$ より小さい場合，文書よりも極端に小さな木となるため，学習の役に立たない可能性を考え抽出しない．一方で，$l_{\max}$ より大きい場合は解析器の学習時間が長くなるため，左右の子スパンに対して再帰的に探索を行うことでより小さい木を抽出する．

3.2 解析器の選定

本研究では SBP の性能改善を目的とするため，疑似正解データの構築に利用する解析器には SBP とは異なる特性をもつ Two-Stage Parser (TSP) [8] を使用した．TSP は統計的モデルにおいて最も高性能であり，特に，関係ラベルの正解傾向が SBP とは異なるため，TSP を利用して構築した疑似正解データを事前学習に用いることで，SBP の関係ラベルの推定性能の改善が期待できる．

表 1 疑似正解データセットのサイズ

名称	抽出単位	木の数	スパンの数
DT	-	91,536	8,162,114
ADT	全体	2,142	57,940
AST	部分木	175,709	2,279,275

4 実験

4.1 データセット

正解データセットとして RST-DT [6] を使用した．RST-DT は学習データ 347 文書とテストデータ 38 文書に分割されており，従来研究 [16] に従い学習データから開発データ 40 文書を分割した．EDU に関しては正解の分割を利用した．

疑似正解データセットの作成に用いる文書データとしては CNN コーパス [17] を使用した．EDU の分割には Neural EDU Segmenter [3] を使用し，Stanford CoreNLP toolkit[4] を用いて前処理を行った．作成された疑似正解データセットに含まれる木の数およびスパンの数を表 1 に示す．提案法である部分木を使用する Agreement Sub Tree (AST) の他に，単一の解析器により木を付与した Document Tree (DT)，複数の解析器の出力が完全に一致した木からなる Agreement Document Tree (ADT) を比較する．

4.2 パラメータ

抽出する木の大きさ: 最大 $l_{\max}$ は RST-DT に含まれる最大の木をもとに 240 とし，最小 $l_{\min}$ は 5 から 10 の間で実験を行い，開発データにおいて最も良い性能であった 9 を選択した．

Span Based Parser (SBP)[5]: 基本的なパラメータは従来の値をそのまま利用し，隠れ状態の次元数のみ 500 次元へと変更した．また，事前学習は 5 epochs，追学習は 10 epochs の学習を行った．SBP は文書から段落，段落から文，文から EDU の 3 段階に分けて解析器を学習する D2P2S2E により最も良い性能を達成しているが，本実験では D2E に対応する EDU を葉とした文書の木を構築する解析器のみを学習する．しかし，解析する際に文と段落の境界を優先的に分割する事により D2P2S2E に近い設定となるよう工夫した．解析時は 5 つの解析器のアンサンブルを行うが，大規模データセットを用いた事前学習に

3) https://github.com/PKU-TANGENT/NeuralEDUSeg
4) https://stanfordnlp.github.io/CoreNLP/
5) https://github.com/nttcslab-nlp/Top-Down-RST-Parser

表 2　Micro-averaged F_1 による性能比較

Model	Span	Nuc	Rel	Full
Two-stage Parser [8]	86.0	72.4	59.7	58.8
NNDisParser [2]	85.5	73.1	60.2	59.9
Span Based Parser [3]	87.1	74.6	60.0	59.6
TSP w/ SpanBERT [5]	**87.9**	**75.8**	**63.4**	-
TSP w/ SpanBERT (reproduced)	87.9	75.7	63.3	62.1
SBP+DT	87.4	74.7	62.7	61.7
SBP+ADT	86.9	74.3	60.5	59.7
SBP+AST	87.1	75.0	63.2	**62.6**
Human	88.3	77.3	65.4	64.7

は時間がかかる．そこで，事前学習された 1 つの解析器を初期値として，異なる 5 つのモデルを追学習により学習した．

Two-Stage Parser (TSP)[6]**：** SVM の最適化に dual coordinate descent を用いているため，初期値を変化させることで複数の解析器を得て，疑似正解データセットの作成に用いた．解析器の数 k は 4 とした．

4.3　評価指標

従来研究 [18] に従い，正解スパンと解析器の出力スパン間の micro-averaged F_1 によりスパン (Span)，核性 (Nuc)，関係ラベル (Rel)，核性・関係ラベル (Full) を評価する．

4.4　実験結果

表 2 に評価結果を示す．疑似正解データを用いない SBP と比較して，部分木を利用する AST のゲインが最も大きく，Rel, Full がそれぞれ 3.2, 3.0 ポイントであった．一方，ADT は得られるデータサイズが小さいためほとんど差がなかった．DT も AST と同様にゲインは得られたが，AST ほど大きくはない．さらに，学習時間は表 1 におけるスパンの数に比例するため，AST の 4 倍の学習時間が必要となる．これらの結果より，複数の解析器の出力を部分木での重複を見て疑似正解データとすることは性能改善に有効であり，学習時間の短縮にも役立つ．一方，TSP w/ SpanBERT と比較すると，Nuc, Rel ではやや劣るものの Full では勝り，世界最高性能を達成した．これは，提案手法における核性ラベルと関係ラベルの正解の一貫性が高いことを示している．

AST を対象としデータサイズを変化させた際の性能の変化を図 3 に示す．Span, Nuc には変化がほとんどなく，Rel, Full のみデータサイズに比例して性

6)　https://github.com/yizhongw/StageDP

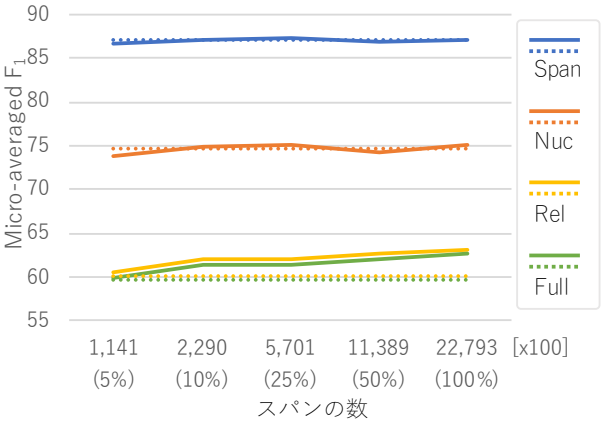

図 3　データサイズによる性能の変化

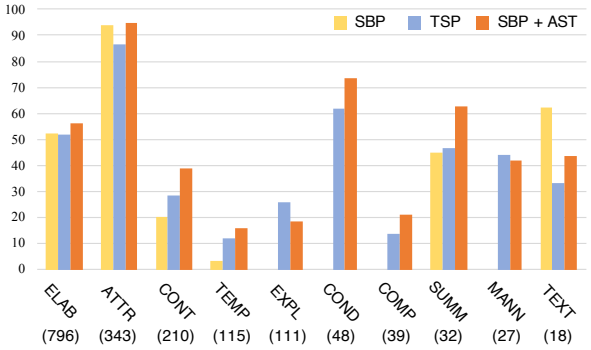

図 4　関係ラベルごとの性能比較

能が向上していることから，データを増やすことで関係ラベルの推定性能はさらに改善することが期待できる．さらに，疑似正解データを構築するための解析器として TSP w/ SpanBERT を用いれば性能改善はより顕著になるものと期待できる．

10 種類の関係ラベルを例としたラベルごとの性能を図 4 に示す．提案した疑似正解データによる学習は，TSP の推定性能が高く SBP の推定性能が低いラベル（例えば EXPL, COND, MANN など）において大きく性能を向上させることを確認した．

5　おわりに

本研究では，ニューラル修辞構造解析器のデータ不足を解消するために．複数の解析器の間で一致する部分木を疑似正解データとし，それを用いて事前学習された解析器を正解データにより追学習する枠組みを提案した．部分木を用いた疑似正解データは木全体を用いる場合と比較して高い性能を達成し，かつ短い時間で学習できる利点を示した．また，疑似正解データの作成に異なる分類器および解析手法に基づく解析器を使用することで，関係ラベルの推定性能を大きく向上させた．

参考文献

[1]W.C. Mann and S.A Thompson. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, USC/ISI, 1987.

[2]Nan Yu, Meishan Zhang, and Guohong Fu. Transition-based neural RST parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 559–570, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[3]Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. Top-down rst parsing utilizing granularity levels in documents. In *Proceedings of the 2020 Conference on Artificial Intelligence for the American*, pp. 8099–8106, New York, America, September 2020.

[4]Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. A unified linear-time framework for sentence-level discourse parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4200, Florence, Italy, July 2019. Association for Computational Linguistics.

[5]Grigorii Guz and Giuseppe Carenini. Coreference for discourse parsing: A neural approach. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pp. 160–167, Online, November 2020. Association for Computational Linguistics.

[6]Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, 2001.

[7]Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics.

[8]Yizhong Wang, Sujian Li, and Houfeng Wang. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 184–188, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[9]Longyin Zhang, Yuqing Xing, Fang Kong, Peifeng Li, and Guodong Zhou. A top-down neural architecture towards text-level parsing of discourse rhetorical structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6386–6395, Online, July 2020. Association for Computational Linguistics.

[10]Chloé Braud, Barbara Plank, and Anders Søgaard. Multi-view and multi-task training of RST discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1903–1913, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[11]Zhengyuan Liu, Ke Shi, and Nancy Chen. Multilingual neural RST discourse parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6730–6738, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[12]Patrick Huber and Giuseppe Carenini. Predicting discourse structure using distant supervision from sentiment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2306–2316, Hong Kong, China, November 2019. Association for Computational Linguistics.

[13]Patrick Huber and Giuseppe Carenini. MEGA RST discourse treebanks with structure and nuclearity from scalable distant sentiment supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7442–7457, Online, November 2020. Association for Computational Linguistics.

[14]Chloé Braud, Maximin Coavoux, and Anders Søgaard. Cross-lingual RST discourse parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 292–304, Valencia, Spain, April 2017. Association for Computational Linguistics.

[15]Kailang Jiang, Giuseppe Carenini, and Raymond Ng. Training data enrichment for infrequent discourse relations. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2603–2614, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[16]Michael Heilman and Kenji Sagae. Fast rhetorical structure theory discourse parsing. *CoRR*, Vol. abs/1505.02425, , 2015.

[17]Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 28, pp. 1693–1701. Curran Associates, Inc., 2015.

[18]Mathieu Morey, Philippe Muller, and Nicholas Asher. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1319–1324, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.