論文 / 著書情報
Article / Book Information

題目(和文)	修辞構造解析器の高度化に関する研究
Title(English)	
著者(和文)	小林尚輝
Author(English)	Naoki Kobayashi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12389号, 授与年月日:2023年3月26日, 学位の種別:課程博士, 審査員:奥村 学,熊澤 逸夫,中山 実,篠崎 隆宏,船越 孝太郎
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12389号, Conferred date:2023/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

修辞構造解析器の高度化に関する研究

東京工業大学
工学院 情報通信系
情報通信コース
博士論文

指導教員 教授 奥村 学
小林 尚輝

2023年3月

目次

図目次

表目次

第 1 章

序論

1.1 研究背景

近年の言語処理技術の発展により，それまで難しいとされていた機械翻訳や対話ボット，英文校正などの自然言語を入力とする幅広いタスクは，数文程度の短い入力であれば高品質な出力が得られるほどに進展した．さらなる展望として，議事録や新聞，web 上に存在するテキストなど，より長い文書を対象とした要約や分類などのタスクで性能の改善が期待される．これを実現するためには，文書を構成する文の間にある隠れた構造や意味関係の正しい理解が必要不可欠である．そこで，単語の並びから文の構造や単語の関係性を決定する構文解析のように，文の並びから文書についてさまざまな課題を考える文脈解析の重要性が高まっていると考えられる．本稿では文脈解析の中で，文書構造を解析する修辞構造解析に取り組む．

修辞構造解析は，文間の意味的構造を元に文書の構造を解析する研究課題である．修辞構造解析は，Mann and Thompson (1987) によって提案された修辞構造理論に基づいて，文書構造を入れ子状のスパン，つまり木構造によって表現する．修辞構造解析では，まず文書を構成する最小単位である談話単位 (Elementary Discourse Unit: EDU) を定義し，文書を EDU の系列へと分割する．そして，それらの間で関係性のある EDU やその系列であるスパンが結合し，より大きなスパンとなることで木構造を構築する．隣り合うスパンの間には，従属関係を表す核性ラベルと修辞関係ラベルが付与される．核性ラベルは核 (Nuclues: N) および衛星 (Satellite: S) からなる N-S， S-N，N-N の 3 通りがあり，N が付与されたスパンを S が付与されたスパンが修飾する，また，修辞関係ラベルは Elaboration, Attribution, Back-ground など 18 種類から構成され，隣り合う二つのスパンの関係性を表す．

修辞構造木を解析する修辞構造解析器は，主に以下の 3 つの分類器と，テキストスパンを分類器の入力となる特徴量ベクトルへと変換するエンコーダから構成され，それらは修辞構造が人手により付与された文書からなるコーパスを用いた教師あり学習によって学習される．

- 木の構造を推定するためにスパンの分割または結合する分類器，
- 二つの隣接するスパン間の核性ラベルを推定する分類器，
- 二つの隣接するスパン間の修辞関係ラベルを推定する分類器．

修辞構造解析により得られる文書構造は，文書を対象とした分類や要約，感情分類などの応用タスクで活用されている．特に，文書要約では修辞構造を活用した研究が多くあり，文書中にある重要な文の抽出や重要度のランク付けに用いられている．例えば，要約に含める文の選択を修辞構造木を利用したナップサック問題へと帰着して解く手法 (Hirao et al., 2015) がある．また，近年のニューラルモデルでは，修辞構造をグラフ構

造として入力し，埋め込み表現へ取り入れて文のランク付けに活用する手法 (Ishigaki et al., 2019; Xu et al., 2020; Kwon et al., 2021) がいくつか提案されている．これらのような，前処理で解析した修辞構造木を応用タスクの入力とする，いわゆるパイプライン処理では，前処理の解析性能が後段にある応用タスクの性能に強く影響するため，高性能な解析器が求められる．

修辞構造解析器を構築する上で考慮する必要があるのは以下の 3 つと考えられる．

一つ目は木を構築するための解析アルゴリズムに関する取り組みである．修辞構造解析では，構文解析と同様，トップダウン（下向き），ボトムアップ（上向き）両方のアプローチが試みられており，解析方法としては，CKY 法や Shift-Reduce 法，スパン分割法が用いられている．初期のいくつかの手法 (Joty et al., 2013; Li et al., 2016) では CKY 法が用いられたが，木を構成するノード数 n に対して $O(n^3)$ の計算量が必要となり計算コストが問題とされている．そのため，多くの手法 (Feng and Hirst, 2014; Wang et al., 2017; Yu et al., 2018) はより高速な Shift-Reduce 法による上向きの解析を行なっている．また，本研究を初めとした下向きの解析手法としてスパン分割法に基づく手法 (Kobayashi et al., 2020; Koto et al., 2021; Zhang et al., 2020) も提案されている．

二つ目は解析器の入力となる特徴量に関する取り組みである．従来の人手により設計されたルールに基づく特徴量に代わり，事前学習されたニューラルモデルを用いた埋め込みベクトルが一般的となった．初期の研究 (Joty et al., 2013; Feng and Hirst, 2014; Wang et al., 2017) においては EDU を構成する単語のほかに，文の句構造木や係り受け木，品詞などの統語情報，EDU に含まれる単語の数や文書中の位置などのルールといった，人手で選択された特徴量が活用された．ニューラルネットワークを用いた研究 (Ji and Eisenstein, 2014; Li et al., 2016; Braud et al., 2017) でも単語の埋め込み表現のほかに従来と同様の統語情報などが併用された．近年の事前学習済みの言語モデルを特徴量獲得に活用した研究 (Guz et al., 2020; Nguyen et al., 2021; Zhang et al., 2021) では従来の統語情報やルールを必要とせずとも高い解析性能を達成している．

三つ目は解析器の学習に用いるデータセットとその拡張方法に関する取り組みである．修辞構造解析では，修辞構造のアノテーションコストの大きさが原因でデータサイズが十分でない問題がある．そこで，異なるタスクの同時学習や疑似正解データセットの作成が行われている．Braud et al. (2016, 2017) は，修辞構造が付与された複数の異なる言語のデータセットによる同時学習や修辞構造解析に類似したタスクの同時学習により学習に用いるデータセットの拡張に取り組んでいる．また，Jiang et al. (2016); Huber and Carenini (2020) は既存の解析器やそれに類する識別器などを活用して，テキストデータに人手を介することなく修辞構造を付与する方法で大規模な疑似正解データセットの作成に取り組んでいる．

1.2 本研究の貢献

本研究では，修辞構造解析における解析性能の改善のために，スパン分割に基づく下向き修辞構造解析の提案と部分木による疑似正解データセットを活用した修辞構造解析の改善を通して貢献する．以降では，それぞれの取り組みに関する本研究の貢献を整理する．

1.2.1 スパン分割に基づく下向き修辞構造解析

本研究では，スパン分割に基づく下向きの解析手法を修辞構造解析へと導入し，さらに文書の持つ階層構造のひとつである段落を用いて解析空間を階層化することで修辞構造解析の性能改善に貢献する．

解析によって得られた修辞構造木を入力として活用する応用タスクでは，修辞構造木から変換して得られる

係り受け木が多く活用されるが，係り受け木における長期の依存関係を正しく獲得するためには，修辞構造木の上部における解析性能が重要である．しかし，従来の修辞構造解析器は Shift-Reduce 法による上向きの解析が一般的であり，上向きの解析手法では解析の誤りが伝播して木の上部の解析性能が低下する懸念がある．とくに，修辞構造解析では文書を対象とするため，文を対象とする構文解析と比較して木を構成するノード数が多く，解析誤りの伝播を制御するのは難しい．解析誤りの伝播を緩和する方法の一つとして，解析空間を分割する方法が考えられ，従来手法として文内と文間による分割がある．これは，EDU を終端ノードとした文の構造と文を終端ノードとした文書の構造を別々に解析したのち，それらを結合することで EDU を終端とした文書の構造を得る方法である．

本研究では，木の上部の解析性能を改善するために，木の上部から下向きに解析を行う解析手法であるスパン分割に基づく解析手法を修辞構造解析に導入する．これは，文の句構造解析に向けて提案された解析手法であり，文に対応するスパンから解析を始め，スパンが単一の単語になるまで再帰的にスパンを分割することで句構造木を構築する．本研究では文書に対応するスパンから解析を始め，スパンが単一の EDU になるまで再帰的に分割することで修辞構造木を構築する．また，本研究では解析誤りの伝播を緩和するための解析空間の分割において，文書の持つ階層構造の一つである段落に着目し，段落と文を用いた 3 層の階層化を提案する．

実験では，修辞構造木および係り受け木の二つの構造において評価を行い，ベースライン手法となる従来の上向き解析と比較して，下向きの解析手法は特に構造の推定において性能改善を示し，段落を利用した階層化についても有効性が示された．一方で提案した解析器は修辞関係ラベルの推定性能に改善の余地があることも明らかになった．

1.2.2 部分木による疑似正解データセットを活用した修辞構造解析

本研究では，複数の解析器の間で一致する部分木を活用したデータ拡張を提案し，大規模かつ信頼性のある疑似正解データセットの構築により修辞構造解析の性能改善に貢献する．

解析器の学習は，人手によって文書に修辞構造木が付与されたデータセットが必要となる．しかし，文書を対象とした修辞構造木の付与は，多くの文からなる文書を扱うことや構造および核性・修辞関係ラベルの付与に専門性を必要とすることからアノテーションコストが大きく，人手によるデータセットの拡張は容易ではない．特に，ニューラルネットワークを利用した解析器の学習には大量の学習データを必要とするが，修辞構造解析における最大のコーパスである RSTDT でさえ 385 文書しかない．そこで，人手のアノテーションを必要とせず自動的に疑似正解データセットを獲得する研究が行われている．疑似正解データセットの作成では，修辞構造の付与されていないテキストコーパスを対象に既存の解析器や従属関係を判別する分類器を教師として用いて修辞構造木を付与する．データ拡張手法のひとつである Tri-training では教師となる分類器を複数用意し，それらの間で合意が取れたデータを疑似正解データとして選択することで疑似正解データセットの信頼性を高める．しかし，修辞構造解析では解析対象の文書が多くの EDU から構成されるため，複数の解析器の間で文書全体の一致をとることは難しい．

そこで本研究では，複数の教師解析器の間で一致した部分木を疑似正解データセットとして活用することで大規模かつ信頼性のある疑似正解データセットを構築する．複数の木の間で一致する部分木の抽出には，木の走査 (tree-traversal) に基づく高速なアルゴリズムを提案し，木のノード数 n に対して $O(n)$ の実行時間で部分木を抽出可能とした．

実験では，疑似正解データセットの種類およびサイズによる性能の変化を比較し，部分木を活用した提案手法が解析性能および学習効率においてその他の手法を上回ることを示した．

1.3 本論文の構成

本論文の構成について説明する．2 章では修辞構造解析の関連研究について述べる．まず，修辞構造解析の研究動向を説明したのち，本研究で取り組むスパン分割に基づく下向き修辞構造解析および部分木による疑似正解データセットを活用した修辞構造解析に関する既存研究について説明する．3 章では，一つ目の研究であるスパン分割に基づく下向き修辞構造解析について説明する．この章では，まず解析により得られる修辞構造木を応用する上で重要な係り受け木に着目して既存の解析器の課題を説明する．次に提案手法として，文書の構造の一つである段落を活用した解析空間の分割とスパン分割による下向きの解析手法を修辞構造解析へと適用する方法を述べる．実験では，ベースラインモデルとの比較により特に構造の推定において提案手法が優れていることを示す．第 4 章では，二つ目の研究である部分木による疑似正解データセットを活用した修辞構造解析について説明する．この章では，まず修辞構造解析器の学習の難しさをデータセットの問題点から説明する．その後，提案手法として疑似正解データセットを用いたデータセット拡張において部分木を活用する手法を説明する．実験では，異なる複数の疑似正解データセットの比較により部分木を活用する利点とその性能改善について述べる．5 章では，まず本論文のまとめを行い，各研究における今後の課題を述べる．そして最後に，二つの研究を通して得られた知見を元に修辞構造解析における今後の研究の方向性について議論する．

第 2 章

関連研究

2.1 修辞構造解析の研究動向

自然言語処理において複数の文にまたがった関係性やつながりを考える研究課題として文脈解析がある．例えば，文中にある代名詞のような参照表現を解析する共参照解析や文間の繋がりから段落などの文書構造を決定する文書の構造解析などがある．本研究で取り組む修辞構造解析は文書の構造解析の一つであり，Mann and Thompson (1987) によって提案された修辞構造理論に基づいて木構造によって文書の構造を表現する．解析により得られた修辞構造木は主に文書要約において活用され (Hirao et al., 2015; Kwon et al., 2021)，核性ラベルがテキストスパン間の重要性を表現することから，要約における文選択に役立つ．

本節では，はじめに修辞構造解析に用いられる解析アルゴリズムと分類器の変遷について時系列に沿って説明する．次に修辞構造解析において重要な課題として取り組まれているデータセットの拡張に関する研究について説明する．

2.1.1 解析アルゴリズムおよび分類器の変遷

修辞構造解析は木構造の推定・核性ラベルの推定・修辞関係ラベルの推定の 3 つサブタスクからなり，解析器はそれぞれのサブタスクを機械学習を利用した分類器により学習，推定することで修辞構造木を解析する．また，解析アルゴリズムとして単一の文を対象として文内の構造を解析する句構造解析で培われてきた解析アルゴリズムが修辞構造解析へと適用された．ここでは時系列に従って，解析器に用いられた分類器と解析アルゴリズムについて説明する．

初期の研究では，人手で設計したルールベースによる特徴量を入力とした SVM や Linear-Chain CRF などの分類器が用られ，CKY 法や Shift-Reduce 法による解析アルゴリズムによって解析を行なった．人手で作成されたルールには EDU を構成する単語のほかに，文の句構造木や係り受け木，品詞などの統語情報，EDU の数や文書中の位置，文の句構造や係り受け構造から得られるラベル情報などが用いられた．CKY 法を用いた手法 (Joty et al., 2013, 2015) では，動的計画法によって全体のスコアが最大となる構造を選択することで，与えられたスコア関数のもとで大域最適解を得られる利点があるが，修辞構造解析では文書を構築する EDU の数が大きいため $O(n^3)$ の計算量が問題となる．そこで，より軽量な解析方法として Shift-Reduce 法に基づく解析手法 (duVerle and Prendinger, 2009; Hernault et al., 2010; Feng and Hirst, 2014; Wang et al., 2017) が数多く提案された．Shift-Reduce 法では，スタックとキューの二つを用いて解析状態を保存し，シフト (shift) と還元 (reduce) のアクションを用いた状態遷移により部分木を結合しながら上向きに木を構築する

ことで，入力となる EDU の数に対して線形の計算量で解析を行う．特に Wang et al. (2017) は SVM を分類器とした解析手法の中でも優れた解析性能を達成している．

ニューラルネットワークによる言語処理技術の進展に従って，修辞構造解析でもニューラルベースの解析器 (Ji and Eisenstein, 2014; Li et al., 2016; Braud et al., 2016) が多く研究された．Li et al. (2016) は CKY 法により，Ji and Eisenstein (2014) は Shift-Reduce 法によりそれぞれ解析を行なった．Braud et al. (2016) は Sequence-to-Sequence フレームワークによる木の生成と出力を木構造に制約するためのいくつかのヒューリスティックによる新しい解析手法を提案した．これらのニューラルベースの解析器と従来の SVM を用いた解析器の比較は Morey et al. (2017) が詳しく，適切な評価の上ではそれらの手法にほとんど差がないことが示されている．

Morey et al. (2017) 以降のニューラルベースの研究では，初期の SVM を用いた解析器からの性能改善がみられる．Yu et al. (2018) は Shift-Reduce 法によって解析を行い，特徴量としてニューラルベースの文内の係り受け解析器 (Dozat and Manning, 2017) から得られる特徴量ベクトルを活用し大幅な性能改善を示した．また，スパンを結合しながら上向きに木を構築する解析手法である Shift-Reduce 法とは異なり，スパンの分割を繰り返すことで下向きに木を構築する解析手法 (Lin et al., 2019; Zhang et al., 2020; Kobayashi et al., 2020; Koto et al., 2021) が提案された．Lin et al. (2019) は文内の修辞構造を対象として，Pointer-Network (Vinyals et al., 2015) を活用した分割点の推定によって下向きに修辞構造木を構築した．この手法を Zhang et al. (2020) が文書の修辞構造解析へと拡張した．そのほかの下向きの解析方法として，Kobayashi et al. (2020) は句構造解析で提案されたスパン分割法 (Stern et al., 2017) を修辞構造解析へと適用した．

近年，BERT (Devlin et al., 2019) を始めとした大規模なテキストコーパスによって事前学習された大規模言語モデルは幅広い言語処理タスクで性能改善をもたらした．修辞構造解析においても大規模言語モデルを活用した解析手法 (Guz et al., 2020; Guz and Carenini, 2020; Koto et al., 2021; Nguyen et al., 2021; Zhang et al., 2021) がいくつも提案されている．Guz et al. (2020) および Guz and Carenini (2020) は Shift-Reduce 法による解析を行い，それぞれ言語モデルとして RoBERTa (Liu et al., 2020) と SpanBERT (Joshi et al., 2020) を使用した．Koto et al. (2021) はスパンの分割を系列ラベリングの問題として扱うことで下向きの解析手法を提案し，BERT による埋め込み表現を入力に利用して解析を行なった．Nguyen et al. (2021) および Zhang et al. (2021) は Pointer-Network を用いた下向き解析に言語モデルとして XLNet(Yang et al., 2019) を導入し解析を行なった．いずれの手法においても言語モデルが用いられる以前の手法と比較して大幅な性能改善が示された．

2.1.2 データセット拡張に関する研究

修辞構造解析器の学習および評価には，Penn Treebank コーパス (Marcus et al., 1993) の一部を対象に，人手によって修辞構造木が付与された RSTDT コーパス (Carlson et al., 2001) が用いられる．RSTDT コーパスは修辞構造解析を対象とした最大のコーパスであるが，含まれる文書数は 385 件のみしかない．これは修辞構造の付与に専門知識を必要とするため，人手によるコーパスの拡張が容易ではないためである．しかし，特にニューラルベースの手法では学習データの規模が性能に大きく影響するため，データセットの拡張に関して多くの研究が取り組まれている．

データセットの拡張には，大きく分けて二つの方法がある．一つ目は，RSTDT とは異なる追加の修辞構造付きデータセットを活用する方法，二つ目は，修辞構造が付与されていないテキストコーパスに修辞構造を付

与することで疑似正解データセットを構築する方法である．

一つ目の追加の修辞構造付きデータセットを活用する方法では，Braud et al. (2016, 2017) の研究がある．Braud et al. (2016) は修辞構造解析の他に文書やそれを構成する文に対する感情分類や時系列分類，属性分類など，合計 13 通りの分類問題をマルチタスク学習により学習することでデータセットを拡張し，解析器の性能および頑健性の向上に取り組んだ．Braud et al. (2017) はドイツ語やスペイン語など多言語の修辞構造木が付与された教師データセットを活用することによりデータ拡張を行った．しかし，これらの手法は人手により修辞構造が付与されたコーパスを必要とする問題が解決されておらず，データ拡張に限度がある．

二つ目のテキストコーパスに修辞構造を付与することで疑似正解データセットを構築する方法として，Jiang et al. (2016) と Huber and Carenini (2019, 2020) の手法がある．これらの手法はデータセットの拡張に人手によるアノテーションを必要としないため大規模に疑似正解データセットを構築できる利点がある．Jiang et al. (2016) は二つの修辞構造解析器を用いてテキストコーパスに修辞構造を付与する Co-training により得られた疑似正解データセットを用いて関係ラベルの推定性能の改善を行なった．Huber and Carenini (2019) は文書を構成する各 EDU ごとの感情極性スコアとアテンションスコアを Multiple Instance Learning (Angelidis and Lapata, 2018) により獲得し，得られたスコアをもとに CKY 法を用いて核性付きの修辞構造木を自動的に付与した．Huber and Carenini (2020) は CKY 法の代わりに確率的ビーム探索を利用して効率的な木の構築を可能とし，さらに長い文書に対しても修辞構造木を付与した MEGA-DT コーパスを公開した．MEGA-DT コーパスは Guz et al. (2020) の研究において解析器の事前学習に用いられ，木の構造および核性の性能改善を示している．

2.2 本研究と既存手法との差分

本節では本研究で取り組んだ二つの研究について，それぞれ関連する既存手法の概要と提案手法との差分を述べる．

2.2.1 スパン分割に基づく下向き修辞構造解析

まず，一つ目の研究であるスパン分割に基づく下向き修辞構造解析に関して既存研究を説明する．

本研究では，下向きの解析方法としてスパンの分割を繰り返す Stern et al. (2017) の手法を修辞構造解析へと適用した．

Lin et al. (2019) はポインターネットワークを用いて深さ優先順にスパンの分割を行う下向き解析手法を提案したが，文書の構造解析を対象とした本研究とは異なり文内の修辞構造を対象とした研究である．ポインターネットワークは，系列モデルである LSTM (Hochreiter and Schmidhuber, 1997) や GRU (Cho et al., 2014) を用いたデコーダを必要とする．本研究で参考にした Stern et al. (2017) のスパン分割法では，系列モデルを必要としない単純なネットワークによって構築されているため，実装や探索が用意という利点がある．

Zhang et al. (2020) はポインターネットを用いた下向き解析手法を文書の修辞構造解析へと拡張した手法である．彼らの手法はスパンの埋め込みに事前学習済み大規模言語モデルを用いており大幅な性能改善が示されている．提案法では，それらの事前学習済み大規模言語モデルを採用していないため，実験では彼らの報告している大規模言語モデルを用いない場合の性能と比較する．

Koto et al. (2021) の手法はスパンの分割を系列ラベリングとして解くことで，提案法と同じく再帰的にスパンの分割を行う．しかし，与えられたスパンに対して分割点は常に一つであるため，系列ラベリングとして

解く利点は少ない．実装が公開されているため，手元で再実験した結果を含めて提案法との比較を行う．

2.2.2 部分木による疑似正解データセットを活用した修辞構造解析

次に，二つ目の研究である部分木による疑似正解データセットを活用した修辞構造解析に関して既存研究を説明する．

Jiang et al. (2016) は Co-training を利用した疑似正解データセットの構築によって関係ラベル推定の性能改善に取り組んだ．複数の解析器によって出力された木の間で合意をとることによって疑似正解データセットを構築する点で提案手法と類似しているが，提案手法では作成される疑似正解データセットが部分木から構成される点で彼らの手法と異なる．性能改善に関しても，彼らの手法は関係ラベルのみを対象としているが，提案法では関係ラベルに限らず，構造や核性の推定にも僅かではあるが改善できる点が異なる．

Huber and Carenini (2019, 2020) は EDU 単位の感情分類を学習したネットワークを用いる疑似正解データセットの構築方法を提案した．特に，Huber and Carenini (2020) の研究では，CKY 法の代わりに確率的ビームサーチを採用することで長い文書に対しても高速な木の付与を可能とし，大規模な疑似正解データセットを構築した．一方，提案法では学習済みの複数の修辞構造木の間で一致する部分木を疑似正解データセットとして抽出する．部分木の抽出は木の走査に基づく探索によって $O(n)$ で列挙可能であり，$O(n^3)$ を必要とする CKY 法や $O(n^2)$ を必要とするビームサーチよりも高速に動作すると考えられる．また，修辞構造木を構成する要素として，木の構造，核性および関係ラベルの 3 つが挙げられるが，Huber and Carenini (2019, 2020) の研究では，木の構造および核性ラベルのみ付与可能である．彼らの手法により付与される核性ラベルは注意機構の重みから自動的に決定されるが，彼らの実験で示されているように推定性能は高くない．一方，提案法では分類器の学習に教師データを必要とするが，核性および関係ラベルの分布は教師データである RSTDT に従っており，これはデータの質において重要である．

第 3 章

スパン分割に基づく下向き修辞構造解析

3.1　研究概要

本研究では，テキストスパンを再帰的に分割して木を構築するスパン分割法による修辞構造解析と，文書の持つ階層構造である段落および文を利用した解析空間の分割を提案する．

解析により得られた修辞構造木は文書要約や文書分類などの応用タスクに使用される．このとき修辞構造木は係り受け木へと変換して用いられるのが一般的である．修辞構造木から係り受け木への変換は核性ラベルを利用したルールによって行われるが，長期の係り受け関係を正しく変換するためには修辞構造木の上部における解析性能が重要となる．図 3.1 の上段には異なる位置で核性を誤った二通りの修辞構造木を，下段にはそれぞれの修辞構造木から同一のルールによって変換して得られる係り受け木を示す．修辞構造木において誤った核性ラベルは赤色を示され，図から分かるように左の修辞構造木では木の下部において，右の修辞構造木では木の上部において核性ラベルを誤っている．係り受け木も同様に誤った構造を赤色で示すと，修辞構造木の上部において核性を誤った右側の係り受け木が多く誤りを含むと分かる．

しかし，修辞構造解析器において多く用いられる Shift-Reduce 法による上向きの解析手法は，スパンの結合を繰り返して木を構築するため，解析誤りの伝播により木の上部の解析性能が低下する懸念がある．

そこで提案法は，木の上部から下向きに解析を行うことで木の上部における解析性能の改善に取り組む．局

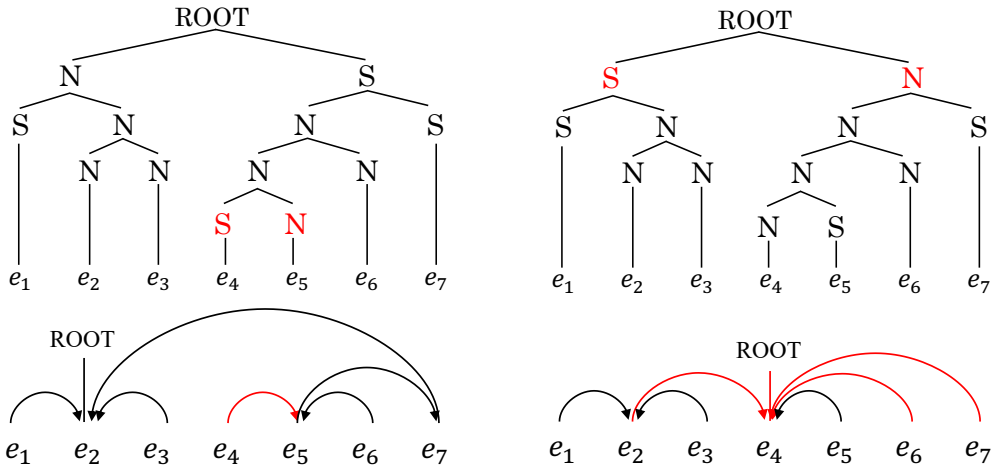

図 3.1　修辞構造木の誤り位置と係り受け木の比較

所的な情報からスパンの結合を繰り返して木を構築する上向き解析に比べて，下向きの解析はより大きな段落やトピックといった大域的な情報が獲得できることが期待できる．解析は文の句構造解析において提案されたスパン分割法を用い，EDU を終端ノードとした文書の修辞構造を解析するためにいくつかの拡張を行う．

また，もう一つの提案手法として，新しい解析空間の分割方法を提案する．修辞構造解析では文書を対象とするため，文を対象とする構文解析と比較して木を構成するノード数が多く，解析誤りの伝播を制御するのは難しい．そこで，解析誤りの伝播を緩和する方法の一つとして，解析空間を分割する方法が考えられ，従来手法として文をもちいた分割による文内と文間の 2 層の階層化がある．提案法では，文書の持つ階層構造の一つである段落に着目し，段落と文を用いた 3 層の階層化を提案する．

実験では，修辞構造木と係り受け木のそれぞれで評価を行い，ベースラインとなる上向きの解析手法と比較した．提案した下向きの解析手法は，特に木の構造，核性の推定に関して高い解析性能を示し，段落を利用した 3 層の階層化によりさらに性能が改善できることが示された．一方で，提案した下向きの解析器は関係ラベルの推定性能に改善の余地があることも明らかになった．

3.2 提案手法

本節では修辞構造木を下向きに解析する方法と段落を用いた解析空間の階層化について説明する．まず，3.2.1 節で段落を用いた解析空間の階層化について説明する．次に，3.2.2 節でスパン分割法に用いた下向き解析手法による修辞構造解析を説明する．

3.2.1 段落を用いた解析空間の階層化

本節では，文書の持つ段落に着目した新しい解析空間の分割について説明する．修辞構造解析では EDU に分解された文書を対象に解析するため，通常，終端ノードは EDU であり木の ROOT にあたるスパンは文書となる．従来の研究では，EDU を終端ノードとした文の修辞構造解析と文を終端ノードとした文書の修辞構造解析の二つに分けて解析を行い，それらを結合することで二段階の階層化を行った．本研究では，EDU を終端ノードとした文，文を終端ノードとした段落，段落を終端ノードした文書の三段階へと階層化する．

それぞれの階層化の例を図 3.2 に示す．図 3.2 左側の (a) は EDU を終端ノードとして文書全体の構造を一つの木で表現する本来の文書の修辞構造解析である．図 3.2 中央の (b) は従来手法の二段階の階層化の例であり，EDU を終端ノードとした文の木が，文を終端ノードとした文書の木に接続される，図 3.2 右側の (c) は提案法である段落を活用した三段階の階層化である．(b) に加えて，段落を用いた階層化が追加されている．

文や段落によって解析空間を分割することは解析誤りの伝播を緩和し，解析に必要な計算量の縮小に寄与するが，本来，修辞構造が文や段落に従った構造であるとは限らない．実際に，RSTDT に含まれる修辞構造木を対象として，性能の上限を後述する RSTParseval の Span によって計測すると，文による分割を行なった場合は 96.3，文と段落の両方による分割を行なった場合は 95.5 となる．この結果から，修辞構造の一部は段落に従わない構造を持っているとわかるが，それは十分少ない割合であると考えられる．

本研究では，それぞれの階層で別々に解析器を学習し，各階層で得た木を結合することで文書全体の修辞構造木を獲得する．そのため，解析器は階層に応じて EDU，文，段落の異なる入力を扱う．したがって，以降はそれらの解析器の入力を解析単位と呼び統一する．

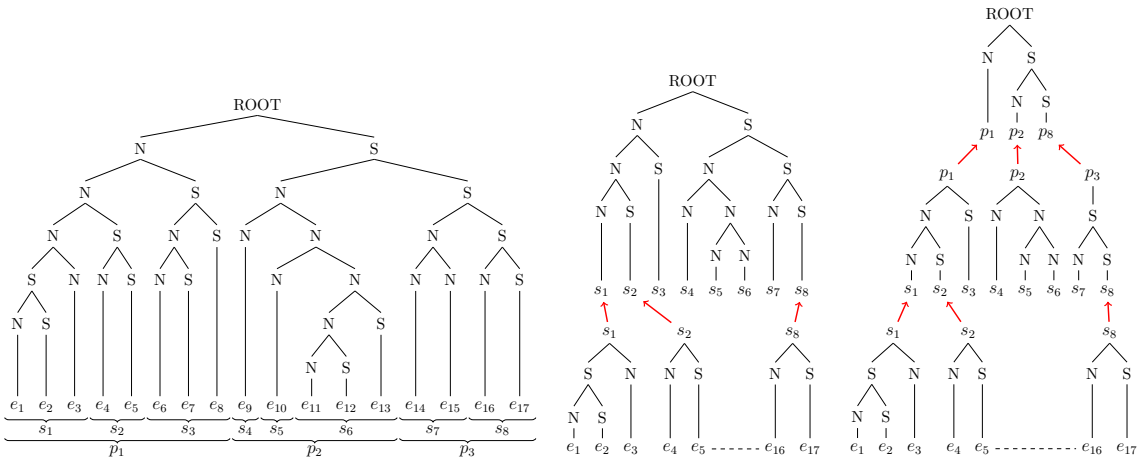

図 3.2　文および段落を活用した修辞構造木の階層化

3.2.2　スパン分割法による下向き解析手法

スパン分割法による下向きの解析は，Stern et al. (2017) によって文内の句構造解析のために提案された．彼らの研究では，単語列に分解された文を一つのスパンとして入力し，そのスパンを再帰的に二分割することで木構造を構築した．

本研究では解析単位である EDU や文，段落の系列を一つのスパンとして入力として再帰的に二分割することで修辞構造を解析する．スパンを分割やラベルの推定には，スパンをベクトルとして表現し，分類器に入力する必要がある．

したがって，まず，解析単位をベクトルで表現した後，解析単位を要素とした任意のスパンをベクトルで表現する方法について説明する．そのあと，それらのスパンの表現ベクトルをもとに，スパンの分割，ラベルの推定を行う方法を説明する．

解析単位のベクトル表現の獲得

解析器の入力となる解析単位を固定長のベクトルで表現する方法について説明する．具体的には，ある解析単位が n 個の単語からなる単語列 $\{w_1, \ldots, w_n\}$ であるとし，ここでは $\{w_1, \ldots, w_n\}$ に対して固定長のベクトル表現 u を獲得する手順について説明する．

まず，各単語をベクトルによって表現する．単語の埋め込み表現には GloVe (Pennington et al., 2014) と ELMo (Peters et al., 2018) を用いる．GloVe は単語の埋め込み表現をその周辺単語から予測する形式で学習する word2vec (Mikolov et al., 2013) を，大域的な単語の共起性を考慮することで改善した手法である．本研究では 300 次元の GloVe ベクトル[*1]を用いた．ELMo は双方向言語モデルを活用して文脈に応じた単語の埋め込みを獲得する．ELMo は文字埋め込み層と 2 層の双方向 LSTM の合計 3 層により構成され，本研究では各層の出力を結合した 3072 次元のベクトルとして単語埋め込みを獲得した．

したがって，解析単位 $\{w_1, \ldots, w_n\}$ に含まれる各単語 w_i に対する埋め込みベクトル表現 $e_i \in \mathbb{R}^{3372}$ を

[*1] `https://nlp.stanford.edu/data/glove.840B.300d.zip`

GloVe と ELMo を用いて以下の式で求める．

$$e_i = [\mathrm{EMB}_{\mathrm{glove}}(w_i); \mathrm{EMB}_{\mathrm{elmo}}(w_i)] \tag{3.1}$$

ここで，$\mathrm{EMB}_{\mathrm{glove}}$ と $\mathrm{EMB}_{\mathrm{elmo}}$ は単語 w_i に対応した GloVe と ELMo のそれぞれのベクトルを返す．$[;]$ はベクトルの結合を表す．

次に単語の埋め込みを元に解析単位に対する固定長のベクトル表現を求める．本研究で提案する解析器は解析単位として EDU のみでなく，より長い系列である文や段落を扱う必要がある．そのため，固定長のベクトルへと変換するためのプーリング処理において，単語の重要度に応じた重みづけを獲得するための Selective Gate (Zhou et al., 2017) を用いた．

したがって，まず単語の埋め込みベクトル e_i を入力として H 次元の隠れ層を持つ前向き，後ろ向きからなる双方向 LSTM (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005) を適用し，それらの出力からベクトル表現 $h_i \in \mathbb{R}^{2H}$ を以下の式によって得る．

$$\overrightarrow{h}_i = \overrightarrow{\mathrm{LSTM}}_{\mathrm{word}}(\overrightarrow{h}_{i-1}, e_i) \tag{3.2}$$
$$\overleftarrow{h}_i = \overleftarrow{\mathrm{LSTM}}_{\mathrm{word}}(\overleftarrow{h}_{i+1}, e_i) \tag{3.3}$$
$$h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i] \tag{3.4}$$

次に，得られたベクトル表現 h_i に対して，各単語の重要度となる $\mathrm{sGate}_i \in \mathbb{R}^{2H}$ を次の式で計算し，それによって重み付けを行なった新しいベクトル表現 h'_i を求める．

$$\mathrm{sGate}_i = \sigma(W_s h_i + U_s s + b_s) \tag{3.5}$$
$$s = [h_1; h_n] \tag{3.6}$$
$$h'_i = h_i \odot \mathrm{sGate}_i \tag{3.7}$$

ここで，$W_s, U_s \in \mathbb{R}^{2H \times 2H}, b_s \in \mathbb{R}^{2H}$ は学習可能なパラメータである．また，シグモイド関数 σ は重みの値域を $[0, 1]$ に制限する．$\odot$ はベクトルの要素積である．

最後に，得られた h'_i に対して系列長に対して平均をとるプーリング処理を行うことで，解析単位を固定長のベクトルによって表現する．

$$u = \frac{1}{n} \sum_{i \in \{1,\ldots,n\}} h'_i \tag{3.8}$$

スパン表現の獲得方法

スパン分割法による解析を行うためには，入力された解析単位の系列に対して，任意のスパンをベクトルで表現する必要がある．したがって，ここでは m 個の解析単位で構成される入力 $\{u_1, \ldots, u_m\}$ に対して，i から j 番目の解析単位を内包するスパン (i, j) をベクトル $u_{i:j}$ によって表現する手順を説明する．

まず，解析単位間の関係性を考慮するために前向き，後ろ向きの LSTM を用いてそれぞれのベクトル表現を獲得する．各 LSTM の隠れ層の次元数は H とした．

$$f_i = \overrightarrow{\mathrm{LSTM}}_{\mathrm{span}}(f_{i-1}, u_i) \tag{3.9}$$
$$b_i = \overleftarrow{\mathrm{LSTM}}_{\mathrm{span}}(b_{i+1}, u_i) \tag{3.10}$$

次に，i から j 番目の解析単位を含むスパンのベクトル表現 $u_{i:j} \in \mathbb{R}^{2H}$ を，Wang and Chang (2016); Ouchi et al. (2020) に従って，スパンの両端にあるベクトル表現を元に以下の式で求める．

$$u_{i:j} = [f_j - f_{i-1}; b_{i-1} - b_j] \tag{3.11}$$

スパンの分割

スパン分割法による下向き解析では，与えられたスパンを再帰的に分割することで木構造を構築する．ここでは，入力されたスパン (i, j) に含まれる分割候補となる各点 k において分割するスコアを求め，最もスコアの高い分割点 $\hat{k}$ を求める．

したがって，分割スコアを得る関数 $s_{\text{split}}(i, j, k)$ を定義する．$s_{\text{split}}(i, j, k)$ はスパン (i, j) を点 k で分割する時のスコアを求める関数であり，Deep Biaffine ネットワーク (Dozat and Manning, 2017) を用いて定義される．

$$s_{\text{split}}(i, j, k) = h_{i:k}^{\top} W_u h_{k+1:j} + v_l^{\top} h_{i:k} + v_r^{\top} h_{k+1:j} \tag{3.12}$$

$$h_{i:k} = \text{MLP}_{\text{left}}(u_{i:k}) \tag{3.13}$$

$$h_{k+1:j} = \text{MLP}_{\text{right}}(u_{k+1:j}) \tag{3.14}$$

ここで，$W_u \in \mathbb{R}^{2H \times 2H}$ および $v_l, v_r \in \mathbb{R}^{2H}$ は学習可能なパラメータである．また，MLP_* は ReLU 関数 (Agarap, 2018) を活性化関数とし，$2H$ 次元の隠れ層を持つ多層パーセプトロンである．

最もスコアの高い分割点 $\hat{k}$ は，$s_{\text{split}}(i, j, k)$ を用いて分割候補点 $k \in \{i, \ldots, j-1\}$ の中から以下の式で求める．

$$\hat{k} = \underset{k \in \{i, \ldots, j-1\}}{\text{argmax}} \; [s_{\text{split}}(i, j, k)] \tag{3.15}$$

ラベルの推定

次に，分割された左右のスパンの間に核性および関係ラベルを推定する．

スパン (i, j) が点 k で分割された時，分割して得られた左右のスパン間に付与されるラベル $\ell \in \mathcal{L}$ のスコアを以下の式で推定する．ただし，ラベル集合 $\mathcal{L}$ は，核性の推定においては $\{\text{N–S}, \text{S–N}, \text{N–N}\}$ の 3 種類，関係ラベルの推定においては 18 種類からなる．

$$s_{\text{label}}(i, j, k, \ell) = v_{\ell}^{\top} \text{MLP}([u_{i:k}; u_{k+1:j}; u_{1:i}; u_{j:n}]) \tag{3.16}$$

ここで，MLP は ReLU 関数を活性化関数とした $2H$ 次元の隠れ層を持つ多層パーセプトロンで，核性と関係の推定にはそれぞれ異なる MLP を利用する．また，$v_{\ell} \in \mathbb{R}^{2H}$ もラベル毎の学習可能なパラメータである．

最も高いスコアを持つラベル $\hat{\ell}$ は，$s_{\text{label}}(i, j, k, \ell)$ が最大となるラベルを以下の式で求める．

$$\hat{l} = \underset{l \in L}{\text{argmax}} [s_{\text{label}}(i, j, k, l)] \tag{3.17}$$

目的関数

最後に，解析器に含まれるパラメータを学習するための目的関数を説明する．

スパン (i, j) に対する正解の分割点およびラベルをそれぞれ k^*, ℓ^* とし，以下のマージン最大化を用いた誤差関数によって，s_{split} と s_{label} がそれぞれ正しい分割点とラベルを推定できるように誤差を求める，

s_{split} は，正解の分割点 k^* と推定した分割点 $\hat{k}$ の間で誤差を計算する．正解と推定した分割点が異なる，つまり $k^* \neq \hat{k}$ である時，正解の分割点におけるスコアがより大きくなるように損失が与えられる．

$$L_{\text{split}} = \begin{cases} \max(0, 1 + s_{\text{split}}(i, j, k^*) - s_{\text{split}}(i, j, \hat{k})) & (k^* \neq \hat{k}) \\ 0 & (\text{otherwise}) \end{cases} \tag{3.18}$$

s_{label} は，正解の分割点 k^* のもとで，分割されたスパン間のラベルを考える．正解と推定したラベルが異なる，つまり $\ell^* \neq \hat{\ell}$ である時，正解のラベルである ℓ^* のスコアがより大きくなるように損失が与えられる．これは核性，関係ラベルのそれぞれで計算される．

$$L_{\text{label}} = \begin{cases} \max(0, 1 + s_{\text{label}}(i, j, k^*, \ell^*) - s_{\text{label}}(i, j, k^*\hat{\ell})) & (\ell^* \neq \hat{\ell}) \\ 0 & (\text{otherwise}) \end{cases} \tag{3.19}$$

これらを正解の木に含まれる全てのスパンに対して計算し，それらの平均を目的関数とする．学習には確率的勾配降下法に基づく最適化手法を用いて，目的関数によって求めた誤差をもとに誤差逆伝播で得られた勾配によってパラメータを学習する．

3.3 実験設定

3.3.1 データセット

解析器の学習および評価には修辞構造解析における標準的ベンチマークデータセットである RSTDT を利用した．RSTDT は学習データ 347 件と評価データ 38 件に分かれており，解析器のハイパーパラメータを決定するために学習データから開発データを分割した．開発データの分割は既存研究である Heilman and Sagae (2015) らに従い，分割後の学習データを 307 件，開発データを 40 件とした．また，従来研究にしたがって，EDU は正解の分割を用いて実験を行なった．

3.3.2 ハイパーパラメータ

LSTM および各 MLP の隠れ層の次元数に用いた H を 250 とし，ドロップアウト率を 0.4 とした．パラメータの最適化は Adam (Kingma and Ba, 2015) を利用し，初期の学習率には 0.01 とした．学習率を減衰させるためエポックごとに学習率を 0.99 倍し，50 エポックまで学習した．学習を安定化させるために勾配クリッピングの値は 5.0 とし，L2 正則化の係数は 0.0001 とした．評価に用いる解析器は，エポック毎に行う開発データによる性能評価において最も性能の高い解析器を選択した．また，選択に用いた評価尺度には後述する OriginalParseval の Full を用いた．

3.3.3 評価指標

修辞構造木の性能を評価するために従来研究で用いられている RSTParseval (Marcu, 2000) のほかに，Morey et al. (2017) らに従って OriginalParseval による評価も行う．どちらの評価においても，多核の N-分木 (N $\geq$ 3) となっている箇所は right-heavy branching tree へと変換[*2]し 2 分木として評価した．

RSTParseval と OriginalParseval はどちらも正解と予測した木のそれぞれに含まれるスパン集合から，一致するスパンの数を元に F 値[*3]を計算する．RSTParseval と OriginalParseval のそれぞれの評価尺度において使用されるスパンの例を図 3.3 に示す．どちらも同一の修辞構造器を表現しているが，核性および関係ラベルが自身のノードが支配するテキストスパンに対するものか，左右の子ノードに対するものかでスパンに違い

[*2] 核性が S-N-S となる N-分木 (N = 3) に関しては left-heavy branching tree へと変換される．

[*3] ただし，本実験では EDU に正解の分割を用いるため，正解および予測した木に含まれる合計のスパンの数は一致するため，Precision，Recall，F 値は同一の値となり，F 値のみで解析器の比較を行う．

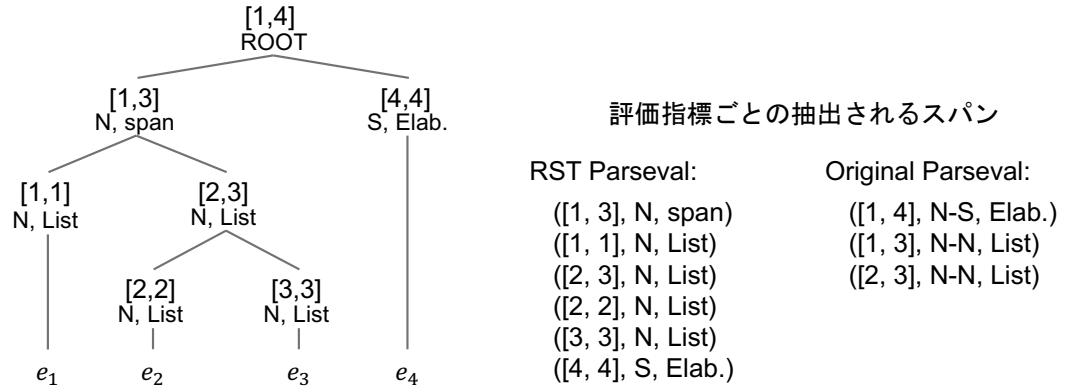

図 3.3　RSTParseval および OriginalParseval で評価に用いるスパンの比較

がある．正解の EDU 分割を用いた場合は，RSTParseval では終端ノードにあたるスパンがかならず正解するため，より正しく性能を表現できるように OriginalParseval の使用が推奨されている．

また，一致するスパンの判定にどのラベルを考慮するかどうかで Span，Nuc.，Rel.，Full の 4 つの評価尺度がある．Span はラベルを考慮せずスパンのみを評価する．Nuc. は核性ラベル付きのスパンによって評価する．Rel. は関係ラベル付きのスパンによって評価する．最後に Full は核性および関係の両方ラベルを付与したスパンによって評価する．

修辞構造木から変換された係り受け木の評価には，Unlabeled Attachment Score (UAS) と Labeled-Attachment Score (LAS) の二つを用いる．どちらも係り受け木に含まれる係り受け関係の親と子が一致するかどうかにより評価を行い，UAS は関係ラベルなし，LAS は関係ラベル込みで評価を行う．

3.3.4　比較手法

比較手法として，上向き解析手法の中でも高い性能を達成している Wang et al. (2017) と Yu et al. (2018) の解析器，および提案法と同じくスパンの分割によって解析を行う下向き解析手法として，Zhang et al. (2021) と Koto et al. (2021) の合計 4 つの解析器を比較手法とする．

Wang et al. (2017) の解析器は，SVM を分類器に用いて，始めに核性付きの構造を推定し，次に関係ラベルの推定を行う二段階の解析を行った．彼らの手法はニューラルネットを用いない手法の中で最も高い性能を達成している．Yu et al. (2018) の解析器は，ニューラルネットワークを利用した解析器であり，文内の係り受け構造を解析するニューラルネットから得られる埋め込み表現ベクトルを活用して大幅な性能改善を達成した．Zhang et al. (2021) の解析器は，ニューラルネットワークを用いた解析器であり，ポインターネットワークを用いて深さ優先順にスパンの分割点を推定し下向きの解析を行う．本実験では，事前学習済み大規模言語モデルを用いていないことから，彼らの実験結果から ELMo を用いた性能を比較する．Koto et al. (2021) の解析器もニューラルネットワークを用いた解析器であり，提案法と同様にスパンを再帰的に二分割するが，彼らはスパンの分割点を系列ラベリングの問題として解いている．

また，提案法における比較として，解析空間の分割による階層化を D2E，D2S2E，D2P2S2E の 3 段階で比較する．D2E，D2S2E，D2P2S2E はそれぞれ図 3.2 の (a)，(b)，(c) に対応しており，階層化を行わない場合，文による階層化を行う場合，文と段落の両方による階層化を行う場合である．

解析手法		RSTParseval				OriginalParseval			
		Span	Nuc.	Rel.	Full	Span	Nuc.	Rel.	Full
Wang et al. (2017)		86.0	72.4	59.6	58.8	72.0	60.5	50.5	48.2
Yu et al. (2018)		85.9	72.5	59.5	58.9	71.8	60.3	49.4	48.4
Zhang et al. (2021)		-	-	-	-	71.8	59.5	47.0	45.9
Koto et al. (2021)		-	-	-	-	73.1	62.3	51.5	50.3
提案法	D2E	86.1	73.1	58.9	58.3	72.3	62.8	47.2	45.9
	D2S2E	86.6	73.4	59.4	59.0	72.7	62.9	48.7	47.7
	D2P2S2E	87.1	74.6	60.0	59.6	**74.1**	**63.7**	48.8	47.9

表 3.1　修辞構造木における性能比較

3.4　実験結果

3.4.1　修辞構造木の性能比較

はじめに，修辞構造木の性能を RSTParseval と OriginalParseval によって比較する．表 3.1 に実験結果を示す．表の上段には，Shift-Reduce 法を用いた上向き解析である Wang et al. (2017) と Yu et al. (2018) の手法，中段には下向き解析である Zhang et al. (2020) および Koto et al. (2021)，下段に提案手法であるスパン分割法を用いた手法の性能をそれぞれ示した．提案法は 5 モデルアンサンブルによる実験結果である．

まず，提案法は，階層化を行わない D2E と比較して，階層化を行う D2S2E，D2P2S2E が高い解析性能となり，さらに段落を活用した D2P2S2E は最も良い結果を示した．この結果から段落を用いた 3 段階の階層化は従来の文のみを利用する階層化よりも優れているとわかる．

上向きの解析手法である Wang et al. (2017) および Yu et al. (2018) と性能を比較すると，Span および Nuc. において RSTParseval と OriginalParseval の両方で大きく性能改善が見られる．さらに，階層化を行わない D2E においても一貫して提案法である下向き解析が既存研究である上向き解析よりも高い解析性能となった．関係を含めた評価尺度である Rel. および Full では従来手法と同等程度の性能であった．

次に，下向きの解析手法である Zhang et al. (2021) および Koto et al. (2021) と性能を比較する．Zhang et al. (2021) は全体として性能が低い．これは，文書に対する分割点の推定順序を推定するポインターネットの学習に RSTDT コーパスが十分な大きさでないためと考えられる．Koto et al. (2021) は Span，Nuc. において提案手法とほとんど同等の性能である．関係ラベルを含めた評価では Koto et al. (2021) のほうが性能が高く，素性として Yu et al. (2018) と同じく文内の係り受け解析によって得られた特徴ベクトルを用いたことが改善の要因である．

一般に，Span，Nuc.，Rel.，および Full はいずれもスパンの一致率に基づく評価尺度であるため，その推定性能にはある程度の相関が見られるのが自然である．表 3.1 の太字で示された D2P2S2E の Span と Nuc. の性能は従来法と比較して最も良い．しかし，関係ラベルを考慮した Rel.，Full では Yu et al. (2018) や Koto et al. (2021) に劣る結果となった．したがって，提案法は関係ラベルの推定に改善の余地があると考えられる．

解析手法		UAS	LAS
Wang et al. (2017)		61.5	47.8
Yu et al. (2018)		61.9	48.4
提案法	D2E	63.9	47.8
	D2S2E	64.0	46.3
	D2P2S2E	64.9	48.5

表 3.2　係り受け木における性能比較

3.4.2　係り受け木の性能比較

次に，係り受け木へと変換した状態で性能比較を行う．表 3.2 に実験結果を示す．比較手法には手元で再現実験を行った Wang et al. (2017) と Yu et al. (2018) の手法を用いた．

D2E，D2S2E，D2P2S2E は順に性能が高くなることから，係り受け木においても段落を用いた階層化は性能改善に役立つことがわかる．

関係ラベルを評価に含めない UAS では，提案法は従来法と比較して最大 3.0 ポイントの改善が得られた．これは，修辞構造木を係り受け木に変換する際に，木の構造と核性ラベルを頼りに親子関係を決定する仕組みであるため，RSTParseval において Span，Nuc. の両方の推定性能が高い提案法は UAS のスコアが高くなったと考える．

一方，関係ラベルを含めた評価では 0.1 ポイントしか改善されておらず，RSTParseval や OriginalParseval の評価と同様に提案法における関係ラベルの改善の余地が示されている．

3.4.3　スパンの長さごとにおける評価

係り受け木の評価である UAS で提案法は性能の改善を示した．長い係り受け関係を正しく解析するためには，修辞構造木の上部を正しく解析する必要があった．そこで，修辞構造木におけるスパンの長さごとに評価した結果を図 3.4 に示す．

比較対象として D2P2S2E の 5 モデルアンサンブルと Wang et al. (2017)，Yu et al. (2018) を用い，評価尺度には係り受け木に変換する上で重要な核性ラベルを考慮した RSTParseval の Nuc. を用いた．

図 3.4 から，短いスパンの間では手法間にほとんど差は見られないが，特に長いスパンにおいて提案法はその他の手法と比較して推定性能が高いことが示された．

3.5　本章のまとめ

この章では，修辞構造木から変換して得られる係り受け木に着目して，木の上部における性能改善を目的として二つの手法を提案した．まず，文書の持つ構造である段落を利用して解析空間を三段階に分割する方法を提案した．次に，文内の句構造解析に用いられるスパン分割法による下向きの解析手法を文書を対象とした修辞構造解析へと適用した．実験では，ベースラインモデルとの比較により構造および核性の推定において提案手法が優れていること，さらに，係り受け木の性能が改善されたことを示した．

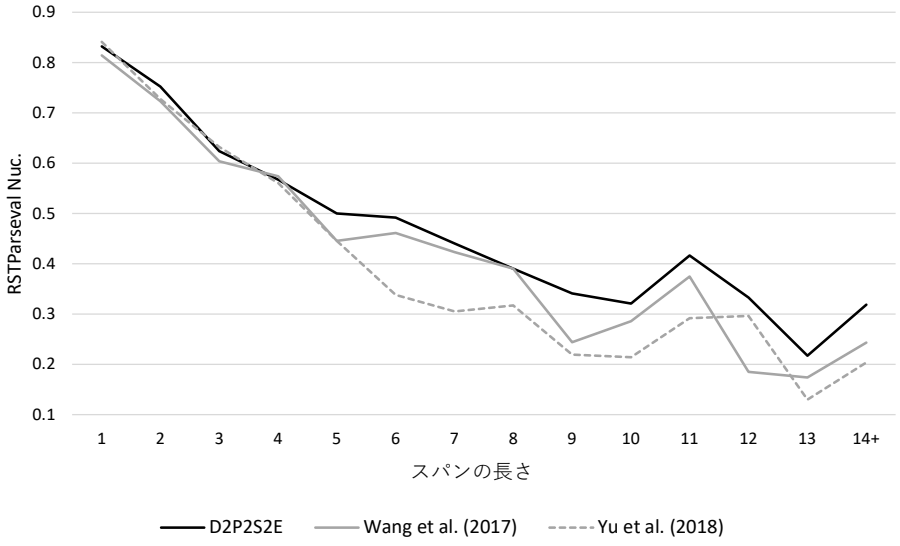

図 3.4　スパンの長さごとの性能比較（RSTParseval Nuc.）

第 4 章

部分木による疑似正解データセットを活用した修辞構造解析

4.1 研究概要

本研究では，部分木を活用した疑似正解データセットを構築し，ニューラルベースの修辞構造解析器の事前学習に利用することで解析性能の向上に貢献する．

3 章で提案したスパン分割に基づく下向きの修辞構造解析器は，木の構造や核性の推定性能で改善を示した一方で，関係ラベルの推定性能では改善の余地が見られた．関係ラベルの学習が難しい理由として，学習に用いる教師付きデータセットである RSTDT の含む文書が 385 件しかないことが挙げられる．また，図 4.1 に示されるように RSTDT に含まれる関係ラベルの出現頻度には大きく偏りがあり，低頻度のラベルは特に学習が難しいことが推測される．

しかし，修辞構造の付与には専門知識を必要とするためアノテーションコストが大きく，人手によるデータセットの拡張は容易ではない．そこで，Self-training や Co-training，Tri-training といった手法を用いて，人手によるアノテーションを必要としない疑似正解データセットを構築することで学習に用いるデータセットを拡張する方法が考えられる．

Tri-training では複数の教師となる分類器の間で合意が得られたデータを疑似正解データとして利用することで品質を向上させる．しかし，単純な分類問題とは異なり，修辞構造解析における出力は木構造であり，文書を対象とするため木のサイズも大きい．複数の教師となる解析器の間で修辞構造木が文書全体において一致するデータは限られ，大規模な疑似正解データセットの構築が難しい．

したがって，提案法は複数の学習済み教師解析器の出力間で，一致する部分木を抽出することで大規模かつ高品質な疑似正解データセットの構築を可能とした．

また，大規模なデータセットを構築する上では，部分木を抽出する際の計算効率も重要となる．そこで，部分木の抽出のために木の走査を元にしたアルゴリズムを提案し，木に含まれるノードの数 n に対して $O(n)$ で一致する部分木の列挙を可能とした．

実験では，部分木を用いた疑似正解データセットによって関係ラベルの推定性能を改善し，かつ，部分木の活用により大幅な学習時間の短縮を示した．また，疑似正解データセットの作成に用いた教師解析器における

図 4.1　RST-DT に含まれる関係ラベルの分布

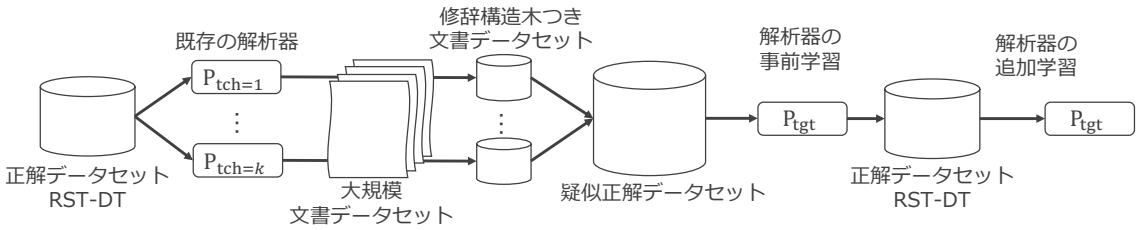

図 4.2　提案法の概要

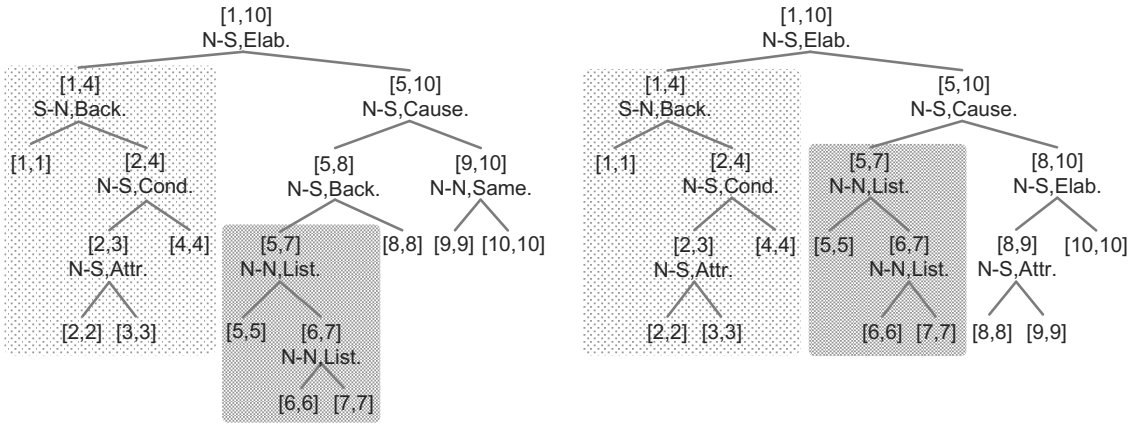

図 4.3　重複する部分木（網掛け部分）の抽出

関係ラベルの解析性能と疑似正解データセットによる性能改善の関係性から，教師解析器と生徒解析器に異なるアーキテクチャに基づく解析器を用いる有効性を示した．

4.2　提案手法

4.2.1　疑似正解データセットの構築および使用手順

疑似正解データセットの構築および使用の流れを図 4.2 に示す．まず正解データセットを用いて教師解析器の学習を行う．この時，出力間で合意を取るために教師解析器を複数用意する．

次に，教師解析木の出力間で合意が得られたデータを疑似正解データとして抽出する．この時，提案法は部分木を活用した疑似正解データの抽出を行う．部分木の抽出方法については 4.2.2 節で説明する．

最後に，得られた疑似正解データセットを生徒解析器の学習に使用する．ここでは，生徒解析器をニューラルベースの手法であると仮定するため，事前学習に疑似正解データセットを活用し，その後，正解データセットで追加学習する．

4.2.2　合意部分木の抽出

本節では，疑似正解データセットを構築するための合意部分木の抽出について説明する．質の高い擬似正解データを得るためには複数の解析器の間で一致する木をそれとすればよい．しかし，修辞構造木は文書から得られる木であるためサイズが大きく，複数の解析器による結果が一致することはまれである．一方で，図 4.3

の網掛けで表されるように，木の全体で一致していなくとも，木の一部では一致することは多い．そこで，本論文では，複数の教師解析器の間で一致した部分木である合意部分木（Agreement SubTree: AST）を疑似正解データとして利用する．

大量の擬似正解データを得るためには，効率的に AST を抽出しなければならない．そこで，AST を効率的に抽出するためのアルゴリズムを提案する．提案するアルゴリズムは木の走査に基づいており，木に含まれるノードの数 n に対して線形の $O(n)$ で動作する．合意部分木を列挙するアルゴリズムはデータマイニング分野で複数の研究 Abe et al. (2002); Zaki (2005); Keselman and Amir (1994) があり，たとえば，一般的な木の集合から合意部分木を得るには Zaki (2005) で提案された最右拡張を用いれば良いが，これは計算量を見積もることができない．一方，本研究では終端ノードが順序も含めて一致する特殊な木を対象とすることから，新たな合意部分木を列挙するアルゴリズムを提案した．

提案するアルゴリズムの目的は，入力された k 個の修辞構造木 (trees) の全てに出現する部分木 (subtrees) を抽出することにある．アルゴリズム 1 に合意部分木を抽出する全体の流れを示す．アルゴリズムは関数 MakeCount，関数 Agreement，関数 FindAST の 3 つから構成される．

アルゴリズム 1: AST の抽出

Input: trees

Output: subtrees

```
count ← MakeCount(trees)
tree ← trees[0]
span ← root(tree)
S ← Agreement(span, count)
subtrees ← FindAST(span, S)
```

アルゴリズム 2: スパンをキーとしての出現数を値にもつ辞書 Count の作成

```
Function MakeCount(trees):
    Count ← dict()
    for tree in trees do
        for span in traversal(tree) do
            if span not in Count then
                Count[span] ← 0
            Count[span] += 1
    return Count
```

はじめに，アルゴリズム 2 に示す関数 MakeCount により出現する全てのスパンの頻度を計算し，スパンをキーとしてその頻度を値とする辞書によって保持する．本アルゴリズムでは，各ノードが内包する先頭と末尾の EDU のインデックスで表される区間とそのノードに付与された核性，関係ラベルの三つ組をスパン (span) として扱い，これを同一性の判定に用いる．ラベルなしのスパンによる同一性の判定も可能であるが，本研究では関係ラベルの推定性能向上のためにラベル付きのスパンを用いた．

アルゴリズム 3: 各スパンに AST の十分条件を付与する関数 AGREEMENT

```
Function AGREEMENT(span, Count):
    S ← dict()
    Function SUBAGREEMENT(span):
        if Len(span)=1 then
            return True
        else
            S[span] ←
              Count[span]=k ∧ AGREEMENT(leftChild(span)) ∧ AGREEMENT(rightChild(span))
        return S[span]
    SUBAGREEMENT(span)
    return S
```

次に，与えられた複数の修辞構造木のうち 1 つを任意に選び，その ROOT スパンをアルゴリズム 3 に示す関数 AGREEMENT へ入力する．関数 AGREEMENT は木の各スパンに AST の十分条件が満たされているかどうかを示すフラグを付与する．あるスパンが AST である条件を満たすためには，左右の子スパンが AST の条件を満たし，かつ自身の出現頻度 Count(span) が木の数 k と一致することである．ここで，スパンの出現頻度 Count(span) は事前に数えることができるので，スパンをキーとして頻度を値に持つ辞書とすることで簡単に出現頻度を参照できる．

最後に関数 AGREEMENT によりフラグが付与された状態のスパンをアルゴリズム 4 に示す関数 FINDAST へと入力する．関数 FINDAST は付与されたフラグを元に，AST である部分木をすべて抽出する．このとき，AST 同士の間では重複が起きないよう処理される．さらに，アルゴリズム内で $l_{\min}, l_{\max}$ を用いて抽出する木の大きさを制御する．

AST がすべての木に共通して含まれるため，関数 AGREEMENT および関数 FINDAST は k 個の修辞構造木全てに対して行う必要はなく，k 個の木のうち一つだけに適用すればよい．ただし関数 MAKECOUNT によりスパンの頻度を事前に計算するときのみ，全ての木を用いて計算を行う．

4.3 実験設定

4.3.1 データセット

教師データセットは人手によって修辞構造木が付与されている標準的ベンチマークデータセットである RSTDT を用いる．RSTDT 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，Heilman and Sagae (2015) に基づき学習データのうち 40 文書を開発データとした．正解データセットの EDU の分割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構築に用いるテキストコーパスは CNN コーパス Hermann et al. (2015) を用いる．CNN コーパスは CNN Dailymail の記事を対象として収集されたデータセットである．修辞構造木を付与するための前処理として，Neural EDU Segmenter (Wang et al., 2018) による EDU の分割を適用した．

アルゴリズム 4: AST を列挙する関数 FINDAST

```
Function FindAST(span, S):
    subtrees ← list()
    Function subFindAST(span):
        if Len(span) < l_min then
            return
        else if Len(span) > l_max then
            subFindAST(leftChild(span))
            subFindAST(rightChild(span))
        else // l_min ≤ Len(span) ≤ l_max
            if S[span] = True then
                subtrees.append(span)
            else
                subFindAST(leftChild(span))
                subFindAST(rightChild(span))
    subFindAST(span)
    return subtrees
```

また，教師解析器を適用するために必要な素性の獲得に Stanford CoreNLP toolkit[*1] を用いた前処理を行なった．

4.3.2 ハイパーパラメータ

$l_{\min}$ および $l_{\max}$

$l_{\min}, l_{\max}$ は抽出する AST の最小，最大サイズを決定するパラメータである．RSTDT の文書に含まれる EDU の数は 7 から 240 であるため，$l_{\min}$ は 5 から 10 の間で開発データによって選択し，$l_{\max}$ は 240 とした．図 4.4 に示される開発データにおける $l_{\min}$ と性能の変化から $l_{\min}$ には開発データで最も良い性能であった 9 を選択した．

教師解析器の数 k

教師解析器の数 k が小さいと解析器による合意の信頼性が低下し，データセットの質が低下する．一方で，k が大きすぎると疑似正解データセットを作成するための時間が膨大となり，また合意をとるための制約が厳しくなるためデータ量も少なくなる．本論文ではデータ作成の時間を考慮して k を 4 としたが，適切な k に関してはまだ議論が必要である．

[*1] https://stanfordnlp.github.io/CoreNLP/

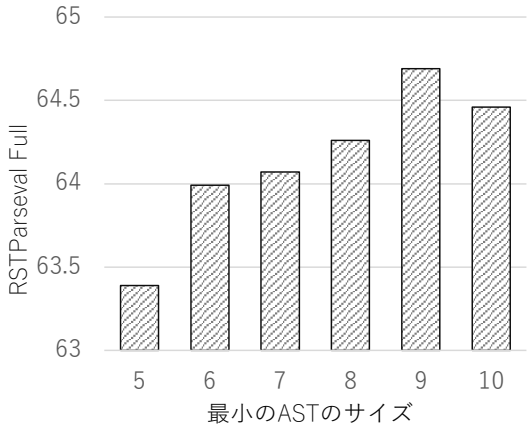

図 4.4　開発データにおける $l_{\min}$ と性能の変化

4.3.3　生徒解析器

生徒解析器は 3 章で提案したスパン分割法による下向き解析手法（Span-Based Parser: SBP）[*2]を利用した．SBP の基本的なパラメータは元の実装に従い，隠れ層の次元数 H は 500 次元とした．また，事前学習と追加学習はそれぞれ 5，10 エポックとし，追加学習時に開発データで最も良い性能を示したエポックのモデルを評価に用いた．

SBP は文書の持つ階層構造である段落と文を利用して，文書から段落，段落から文，文から EDU という各階層で別々の解析器を学習することで高い解析性能を達成した．しかし，疑似正解データセットを用いて階層ごとに事前学習を行うことは計算量から容易ではない．したがって，階層構造を利用せず単一の解析器で解析するモデルを学習し，解析時に段落と文の境界を優先的に分割する処理を取り入れることにした．具体的には，スパンの分割候補に段落の境界にあたる分割点が 1 つ以上存在する場合は，段落の境界にあたる点のみを対象に分割点を決定する．これにより，段落の境界が優先して分割される．同様に段落内では文の境界にあたる分割候補を対象にして分割点を決定し，優先的に分割する．

また，3.3 節では，異なる初期値で学習した解析器が推定した分布を平均化することでアンサンブルを行った．しかし，事前学習から追加学習までを異なる初期値で行うことは時間コストの観点から現実的でない．したがって，疑似正解データセットによる事前学習は一度だけ行い，追加学習時を複数通り行いアンサンブルに必要な解析器を用意した．

4.3.4　教師解析器

教師解析器は疑似正解データセットの作成に用いるため，生徒解析器の欠点を補えることが望ましい．この場合，修辞関係ラベルの推定で高い性能を達成している解析器が望ましい．修辞関係ラベルを高性能に推定する解析器の候補として，SVM を用いた TSP (Wang et al., 2017) とニューラルネットワークを用いた NNDisParser (Yu et al., 2018) があり，どちらも Shift-Reduce 法による上向き解析法である．

[*2] https://github.com/nttcslab-nlp/Top-Down-RST-Parser

疑似正解データセット	k	$l_{\min}$	木の数	ノードの数
DT	1	-	91,536	8,162,114
ADT	4	-	2,142	57,940
AST	4	5	534,352	4,087,989
		6	387,636	3,501,125
		7	290,532	3,015,605
		8	223,101	2,611,019
		9	175,709	2,279,275
		10	140,384	1,996,675

表 4.1　疑似正解データセットを構成する木の数およびノードの数

NNDisParser は関係ラベルの推定で TSP より高い性能を達成しているが，特徴量として文内の係り受け解析を行うニューラルネットワークの特徴ベクトルを必要とすることから，大規模なデータ作成に向いていない．したがって本研究では，TSP を教師解析器として利用する．

TSP で分類器として用いられる SVM は確率的双対座標降下法で最適化されるため，異なる初期値で複数の解析器を学習し，それらの間で合意をとることで擬似正解データセットを作成した．

4.3.5　評価指標

3.3 節と同様に RSTParseval と Original Parserval による評価を行う．

4.3.6　比較手法

部分木を活用した疑似正解データセットによる事前学習の有効性を検証するために，単一の解析器による結果を疑似正解データセットした Document Tree (DT)，複数の解析器により文書全体で合意した結果を疑似正解データセットとした Agreement Document Tree (ADT) を比較対象とした．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 4.1 に示す．

また，近年の修辞構造解析器との比較を，教師解析器として利用した Wang et al. (2017) の TSP の他に，Yu et al. (2018) の係り受け解析器の特徴ベクトルを利用した上向き解析手法，Two-Stage Parser に SpanBERT を導入しニューラルモデルに再構築した Guz and Carenini (2020) の手法，ポインターネットワークを利用した下向き解析手法に対し，敵対学習による最適化を導入した Zhang et al. (2021) の手法と比較を行う．さらに，本研究と同じく疑似正解データを利用する研究である Guz et al. (2020) の手法，EDU 分割までを end-to-end で学習する Nguyen et al. (2021) の手法とも比較を行う．

また，表 4.2 に RSTDT と MEGA-DT[*3]および提案法である AST の統計量を比較した結果を示す．MEGA-DT は配布されている train/dev/test のうち train を対象とし，AST は実験に用いた k=4, $l_{\min}$=9 のデータを対象とした．どちらの疑似正解データセットも RSTDT と比べてデータ数，ノード数共に多いが，平均 EDU 数は少ない．核性ラベルの割合を見ると，AST は RSTDT により学習された解析器によりラベル

[*3] https://nlp.cs.ubc.ca/mega-dt

データセット	データ数	ノード数	平均 EDU 数	核性ラベルの割合 (N-S/S-N/N-N)
RST-DT	347	21,404	56.6	61.1% / 16.3% / 22.6%
MEGA-DT	267,928	4,829,875	19.0	22.2% / 29.9% / 47.0%
AST (k=4, $l_{\min}$=9)	175,709	2,279,275	14.0	68.7% / 13.9% / 17.4%

表 4.2　RST-DT と疑似正解データである MEGA-DT および ADT の比較

手法	事前学習後				追学習後 (5 モデル平均/アンサンブル)							
					平均				アンサンブル			
	Span	Nuc.	Rel.	Full	Span	Nuc.	Rel.	Full	Span	Nuc.	Rel.	Full
SBP	-	-	-	-	86.3	73.1	57.6	57.3	87.1	74.6	60.0	59.6
SBP+DT	86.4	72.8	59.8	59.0	**86.9**	74.1	61.8	61.0	**87.4**	74.7	62.7	61.7
SBP+ADT	85.6	69.9	55.4	54.4	86.6	73.5	59.5	58.8	86.9	74.3	60.5	59.7
SBP+AST	86.3	72.3	59.1	58.5	86.8	**74.7**	**62.5**	**61.8**	87.1	**75.0**	**63.2**	**62.6**

表 4.3　RSTParseval による疑似正解データセット毎の比較

が付与されているため RST-DT とおおよそ同一の分布である．一方，MEGA-DT は Yelp'13 に EDU を単位とした感情極性ラベルが付与された SPOT データセット (Angelidis and Lapata, 2018) を元にして構築しているため RST-DT と分布が異なっている．

4.4　実験結果

4.4.1　疑似正解データセットの比較

表 4.3 に事前学習モデルと追加学習したモデルの RSTParseval における性能を示す．表中の SBP は疑似正解データセットを用いず，正解データセットのみで学習した実験結果であり 3 章の実験結果と同様のものである．これをベースラインとして疑似正解データセットの有効性を比較する．部分木を擬似正解データとして利用する AST はベースラインと比較してすべての指標において性能が向上しておりその有効性がわかる．疑似正解データセット間で比較すると，多くの指標で AST を用いた場合が最もよく，特に修辞関係ラベルに関連する Rel. と Full における改善が顕著である．文書単位の疑似正解データから構成される DT と ADT も同様にベースラインを上回る性能を達成したが，AST よりも性能の改善は小さい．

特に ADT はその他の疑似正解データセットと比較して改善幅が小さいが，これは疑似正解データセットのサイズが原因であると考える．表 4.1 に示したように ADT に含まれる木とスパンの数はそれぞれ 2,142 および 57,840 であり，AST の 175,709 と 2,279,275 と比較して非常に少ない．つまり，事前学習に十分な量でないと考える．一方，DT において，木の数は 91,536 と AST と比較して少ないが，それぞれの木が文書全体から構成されるため含まれるスパンの数は非常に多く，AST の 4 倍近い 8,162,114 個のスパンが含まれている．それにもかかわらず，性能においては AST のほうが良い．これは DT が単一の教師解析器から作成されているため，疑似正解データの質がその他よりも劣ることが原因であると考える．

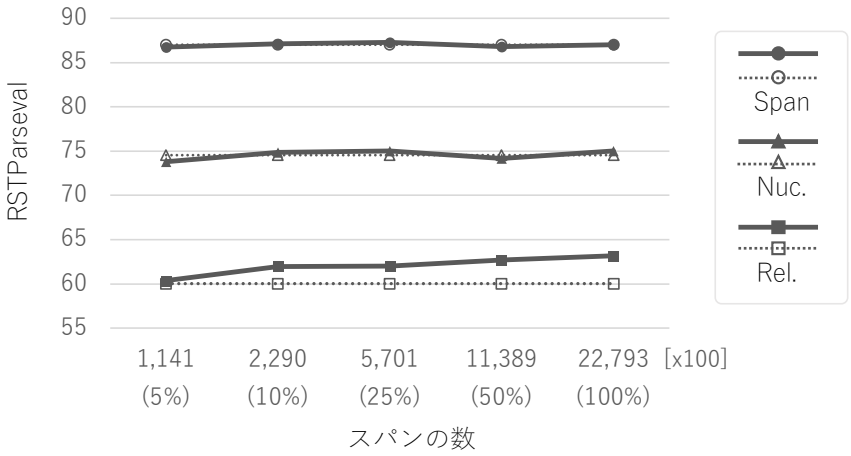

図 4.5　疑似正解データセットのサイズによる性能の変化

よって，疑似正解データセットを部分木で構築することにより，複数の教師解析器による合意を得た疑似正解データセットが大規模に獲得可能であり，文書全体を用いる疑似正解データセットよりも量と質の両面において優れていることがわかる．

また，学習にかかる計算時間の観点においても，学習時間がデータセットに含まれるスパンの数に比例しているため，AST は DT の 1/4 の時間で学習が可能である．これは部分木を用いることでデータセットを小さくできることの利点である．

4.4.2　疑似正解データセットのデータサイズによる影響

疑似正解データセットのサイズと性能の関係を調べるため，疑似正解データセットのサイズを変化させて評価実験を行った．その結果を図 4.5 に示す．実線が AST を事前学習に用いた場合 (表 4.3 における SBP+AST) の性能．点線が疑似正解データを用いない場合 (表 4.3 における SBP) の性能である．図より，Span と Nuc. はデータセットのサイズが変化してもほとんど性能に変化がない．つまり，スパン分割と核性ラベルの推定に関しては擬似正解データの効果は薄い．一方で Rel. はデータサイズの増加に応じて性能が向上しており，疑似正解データセットの有効性がわかる．これはスパンの分割および核性ラベルの推定がそれぞれ 2 クラス，3 クラスの分類問題という比較的簡単な分類問題であることに対し，修辞関係ラベルの推定は 18 クラスかつクラスの出現頻度に偏りのあるデータに対する分類問題という難しい問題であることが原因であると考える．

さらなるデータの増強方法として既存の大規模擬似データである MEGA-DT と AST を組み合わせることが考えられるが，先に述べたように Span および Nuc. はデータサイズを増やしても顕著な改善が見られないことから，核性しか付与されていない MEGA-DT との組み合わせによるデータの増強は性能向上にはつながらないと考えられる．

4.4.3　関係ラベルごとの性能比較

AST を用いることで関係ラベルの推定性能が大きく改善できることがわかった．そこで，どの関係ラベルにおいて性能が改善されたのかを確認するために，SBP+AST と他にベースラインである SBP と教師解析

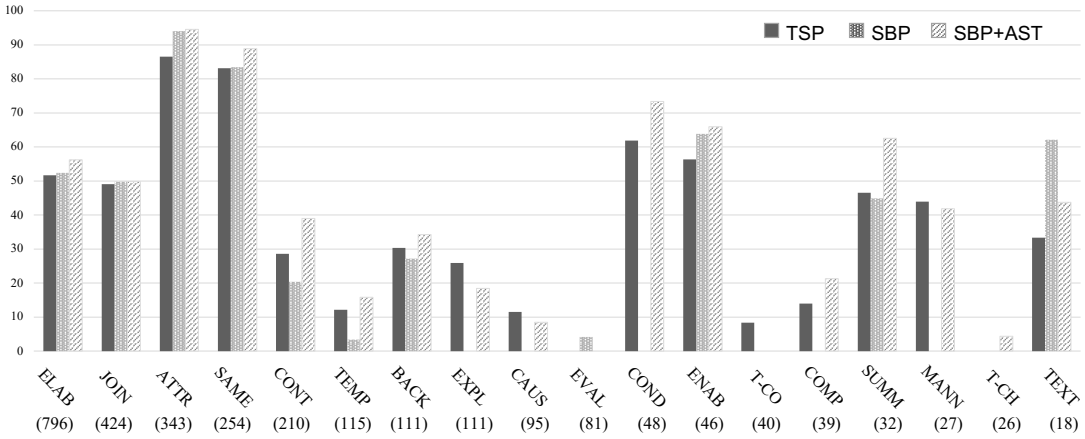

図 4.6　TSP，SBP，SBP+AST の関係ラベル (18 種類) ごとの F 値の比較

手法	単語埋め込み/学習済みモデル	外部データ	RSTParseval			OriginalParseval		
			Span	Nuc.	Rel.	Span	Nuc.	Rel.
Wang et al. (2017)	-	-	86.0	72.4	59.7	72.0	60.5	50.4
Yu et al. (2018)	係受け解析器	-	85.9	72.5	59.5	71.8	60.3	49.4
Guz et al. (2020)*	RoBERTa	-	86.2	73.0	-	72.4	61.4	-
Guz et al. (2020)*	RoBERTa	MEGA-DT	86.5	73.5	-	72.9	61.9	-
Guz and Carenini (2020)†	SpanBERT	-	87.9	75.7	63.3	75.8	64.6	53.7
Nguyen et al. (2021)*	XLNet	-	87.6	76.0	61.8	74.3	64.3	51.6
Zhang et al. (2021)*	ELMo	-	-	-	-	71.8	59.5	47.0
Zhang et al. (2021)*	XLNet	-	-	-	-	76.3	65.5	55.6
SBP	GloVe+ELMo	-	87.1	74.6	60.0	74.1	63.7	48.8
SBP+AST	GloVe+ELMo	AST	87.1	75.0	63.2	74.1	64.7	54.1

表 4.4　既存の解析器との比較

器である TSP の間で関係ラベル毎の F_1 を図 4.6 で比較する．ラベル名の下にある数値は各ラベルの出現頻度を表し，ラベルは出現頻度順に並べてある．SBP および SBP+AST は 5 モデルアンサンブルの結果を用いる．

図より，多くの関係ラベルにおいて TSP が SBP と同等かそれ以上の性能であり，SBP+AST の性能がそれらを上回っている．特に，低頻度の関係ラベルにおいて SBP+AST は大きく SBP の性能を改善している．また，TSP が推定可能だが，SBP が不得意な関係ラベルを SBP+AST によって推定可能になったことから，疑似正解データセットの作成に TSP を用いたことが性能向上に寄与していると考えらえる．

4.4.4　既存の解析器との比較

提案法である SBP+AST の 5 モデルアンサンブルと既存手法の比較を表 4.4 に示す．*で示された結果は論文で報告された値，それ以外は手元で評価した値である．†Guz and Carenini (2020) らの解析器は NoCoref の値を参照した．ここでは RSTParseval だけでなく，OriginalParseval による評価も行った．テキストスパ

モデル	正解 EDU				EDU 分割器			
	Span	Nuc.	Rel.	Full	Span	Nuc.	Rel.	Full
Nguyen et al. (2021)*	**74.3**	64.3	51.6	50.2	**68.4**	**59.1**	**47.8**	**46.6**
SBP	74.1	63.7	48.8	47.9	68.2	56.8	43.1	42.3
SBP+AST	74.1	**64.7**	**54.1**	**52.7**	68.1	57.4	47.6	45.9

表 4.5　EDU Segmenter による EDU 分割を採用した際の解析器の性能比較

ンのベクトル表現を得るために SpanBERT を採用した Guz and Carenini (2020) の手法,XLNet Yang et al. (2019) を採用した Zhang et al. (2021) の手法はその他の手法と比較して非常に高い性能を達成している．SBT+AST はこれらを除くとどちらの評価方法においても Span を除いた Nuc.,Rel. の 2 つの指標において従来法を上回っている．提案法と Guz and Carenini (2020) を比較すると，RSTParseval においては Guz and Carenini (2020) がわずかに良い性能であるが，OriginalParseval においては提案法が逆転している．正解の EDU 分割を用いた場合の RSTParseval は正解スパンと必ず一致する長さ 1 のスパンを過大評価することから OriginalParseval を利用することを文献 (Morey et al., 2017) は推奨している．よって，この結果は提案法の有効性を支持していると考える．

ポインターネットワークを用いた下向き解析法に敵対学習を導入した Zhang et al. (2021) の手法は，テキストスパンのベクトル表現を XLNet から得た場合には現状での最高性能を達成しているがそれを ELMo へ変更した場合にはスコアが大きく劣化しており，高いスコアは XLNet がもたらしたと考える．提案法は，テキストスパンのベクトル表現を GloVe+ELMo から得ているにもかかわらず Zhang et al. (2021) の方法に対し Rel. で 1 ポイント程度の差に迫るスコアを達成しており，疑似正解データの有効性は示せたと考える．

また，Guz et al. (2020) は RoBERTa (Liu et al., 2020) を用いた解析器に MEGA-DT による事前学習を適用したが，Span，Nuc. の性能改善は小さく，この点は AST を用いた提案法に関しても同様であった．これは木の構造や核性の推定がそれぞれ比較的簡単な 2 クラス，3 クラスの分類問題であるため少ない正解データのみでも十分な性能を達成できるからである．一方で，Rel. において AST は推定性能を大きく改善したが，MEGA-DT は関係ラベルが付与されていないため，Rel. の改善は不可能である．この点において AST は MEGA-DT に対して優位である．

本研究では扱わないが，比較手法に挙げられる RoBERTa や XLNet 等の大規模言語モデルを活用した手法に対して，提案法である AST を用いた事前学習を適用することも可能である．しかし，それらの大規模言語モデルは大量のテキストデータを用いた事前学習によって既に汎用的なパラメータを獲得しており，データ拡張による性能の改善は大きくないと予測される．

4.4.5　EDU 分割器による EDU 分割を用いた評価

表 4.5 に，正解の EDU，EDU 分割器により自動的に分割した EDU を用いた場合の解析器の性能を示す．* で示す結果は Nguyen et al. (2021) での報告値である．提案法は SBP+AST の 5 モデルアンサンブルを用い，Neural EDU Segmenter[*4] (Wang et al., 2018) を利用した．比較対象となる Nguyen et al. (2021) は EDU 分割と修辞構造解析を end-to-end に学習・推定する手法であり，我々の用いた EDU 分割器と EDU 分

[*4] https://github.com/PKU-TANGENT/NeuralEDUSeg

割の精度が異なる[*5]ことに注意されたい．

表 4.5 に OriginalParseval を用いた評価結果を示す．表より，どちらの手法も自動分割した EDU を用いる性能は大きく劣化する．SBP と SBP+AST を比較すると，疑似正解データは Rel.，Full に対する性能改善に強く貢献しており正解 EDU を用いた場合と同様の傾向にある．Nguyen et al. (2021) の方法は，Zhang et al. (2020) と同様にポインターネットワークを用いて再帰的にスパンを分割していき，予測された分割点が与えれたスパンの末尾である場合を分割の終了条件とすることで解析から EDU 分割までを一括で推定する．正解 EDU を用いた場合には提案法が Nguyen et al. (2021) の方法に勝っているが，自動分割した EDU の場合には劣る結果となった．提案法は，決定的に自動分割された EDU を利用しているが，Nguyen et al. (2021) の方法は end-to-end に EDU 分割までを学習することが利点となったと考える．一方，Nuc. と比較して Rel. と Full の劣化度合いは提案法のほうが小さい．これは正解 EDU の場合と同様，疑似正解データの効果であると考える．

4.5 本章のまとめ

本章では，修辞構造解析器の学習に用いるデータセットを拡張する方法について取り組んだ．修辞構造解析で用いられる RSTDT コーパスは 385 件の文書しか含まれず，その関係ラベルには偏りがあるため学習が難しい．しかし，アノテーションコストの観点から人手によるデータの拡張は容易ではない．提案法では，人手のアノテーションを介さない疑似正解データセットを大規模かつ高品質にするために部分木を活用する手法を提案した．実験では，提案法による部分木を用いた疑似正解データセットが解析性能の改善および学習時間の観点から優れていることを示した．また，大規模言語モデルを用いた手法には性能でわずかに劣る結果ではあったが，関係ラベルの推定性能はそれらに迫る性能まで改善した．

[*5] 提案法で使用した Neural EDU Segmenter の分割性能は Precision: 91.7, Recall: 97.5, F1: 94.5 である．

第 5 章

結論と今後の課題

5.1 結論

本研究では，修辞構造解析における解析性能の改善のために，スパン分割に基づく下向き修辞構造解析の提案と部分木による疑似正解データセットを活用した修辞構造解析の構築を通して貢献した．

一つ目の研究では，スパン分割に基づく下向きの解析手法を修辞構造解析へと導入し，文書の持つ階層構造のひとつである段落を用いた解析空間の階層化を提案した．修辞構造木から変換して得られる係り受け木は応用タスクに活用されており，係り受け木における長期の依存関係は修辞構造木の上部構造の解析性能が重要となるが，従来の上向き解析では解析誤りの伝播により木の上部の解析性能が低下する懸念があった．本研究では，木の上部の構造をより正確に解析するために木の上部から下向きに解析を行う解析方法としてスパン分割法による解析手法を修辞構造解析へと導入した．それに伴い，終端ノードである EDU およびその系列であるスパンの表現方法に関して新しいエンコーダを提案した．また，解析誤りの伝播を緩和するために文書の持つ階層構造のひとつである段落に着目し，段落と文を用いた三層の階層化を提案した．実験では修辞構造木および係り受け木の二つの構造において評価を実施した．修辞構造木の評価では，解析による出力と正解の修辞構造木の間で，木に含まれるスパンの一致に基づいて正しく解析できているかを評価する Standard Parseval を使用した．係り受け構造の評価では，係り先と係り元のペアの一致率に基づいて評価を行う unlabeled/labeled attachment score を用いた．どちらの評価尺度においてもベースライン手法となる従来の上向き解析と比較して提案法である下向きの解析手法は特に構造の推定において性能改善が確認された．また，段落を利用した階層化についても有効性が示された．一方で提案した解析器は修辞関係ラベルの推定性能に改善の余地があることも明らかになった．

二つ目の研究では，複数の解析器の間で一致する部分木を活用したデータ拡張手法を提案した．ニューラルネットワークの学習には大量の教師データが必要となるが，修辞構造のアノテーションコストが大きいため人手によるデータセットの拡張が容易ではない．そのため，最大のデータセットである RSTDT でも 385 文書しかなく，その修辞関係ラベルの分布には偏りがある．本研究では，既存の解析器を用いてテキストコーパスに修辞構造を付与することで疑似正解データセットの構築を行なった．複数の教師解析器によって付与した構造の間で一致するものを利用することで疑似正解データの品質を向上させることができる一方で，それらの構造が文書全体で一致するデータは限られており，十分な量の疑似正解データセットが得られない．そこで，複数の教師解析器の間で一致した部分木を疑似正解データとして活用することで，大規模かつ信頼性のある疑似正解データセットの構築を提案した．複数の木の間で一致する部分木の抽出には，木の捜査 (tree-traversal) に基づく高速なアルゴリズムを提案し，木のノード数 n に対して $O(n)$ の実行時間で部分木を抽出可能とし

た．部分木の一致により得られた疑似正解データセットは文書全体の一致によるものと比較しておよそ 40 倍の大きさとなった．実験では，解析の出力と正解の修辞構造木の間で，木に含まれるスパンの一致に基づいて正しく解析できているかを評価する Standard Parseval によって評価を行い，疑似正解データセットの種類による比較，疑似正解データセットのサイズによる性能変化，関係ラベルごとの性能，その他のベースライン手法との比較を行なった．疑似正解データセットの種類による性能の変化では，部分木を活用した提案手法が性能および学習効率においてその他の手法を上回ることを確認した．疑似正解データセットのサイズによる性能変換では，疑似正解データセットを大きくするにつれて関係ラベルの性能が改善されることを確認した．関係ラベルごとの評価では，疑似正解データセットの作成に使用した教師解析器の解析性能が高いラベルにおいて生徒解析器の解析性能も改善されることを確認した．また，ベースライン手法との比較では，事前学習済み言語モデルを活用した手法には劣るが，その他の手法と比較すると全ての評価尺度でそれらを上回った．

5.2 今後の課題

本節では，まず本研究で取り組んだ二つの研究について今後の課題を述べる．その後，二つの研究を通して得た知見から修辞構造解析における今後の方向性を議論する．

5.2.1 スパン分割に基づく下向き修辞構造解析

一つ目の研究として取り組んだスパン分割による下向きの解析手法について，今後の課題を 4 点挙げる．

一つ目の課題として，解析器の入力に利用する特徴量の改善が挙げられる．従来の手法では人手で設計された特徴量が活用されていたが，本研究では GloVe と ELMo と呼ばれる事前学習済みのベクトル表現を利用した．一方で，近年の自然言語処理では Transformer を利用した事前学習済み言語モデルを特徴量獲得に活用することで，幅広いタスクにおいて大幅な性能改善が示されている．したがって，修辞構造解析においてもそれらの事前学習済み言語モデルを適用することで性能を改善できると考えられる．実際に，いくつかの研究ではすでに事前学習済み言語モデルを修辞構造解析へと適用することで性能改善を達成しており，本研究で提案した下向きの解析器でも同様に事前学習済み言語モデルを適用することで更なる性能改善が期待できる．

二つ目の課題として，提案手法で活用した段落情報の獲得方法が挙げられる．本研究では，実験に利用した RSTDT コーパスにおいてあらかじめ付与されている段落を利用した．しかし，一般の文書においては段落が明示的に付与されていない例は珍しくない．したがって，本研究で提案した段落を活用した 3 段階の解析空間の分割には，解析前に文書の段落情報を獲得する方法が必要である．段落情報を獲得する方法のひとつとして，段落の境界を識別する分類器を学習することが考えられる．段落境界の推定を学習するだけであれば，異なる文書から段落を抽出しそれらを結合した文書を活用して簡単に分類器を学習できる．

三つ目の課題として，解析空間ごとに適したスパンの表現獲得方法の模索が挙げられる．本研究では，下向き解析器においてスパン表現を獲得するために，BiLSTM と Selective-Gate を活用したエンコーダを利用した．このエンコーダは．EDU からなるスパン，文からなるスパン，段落からなるスパンのいずれにおいても共通の構造を使用したが，修辞構造の推定に重要な要素は階層ごとに異なるかもしれない．例えば，段落のような長いスパンの関係を考えるためには，それぞれのスパンに含まれる文脈を含めた題目を獲得することが有効だと考えられるが，一方で EDU や文の間にある関係を推定するには，単語間の依存関係や接続詞などを強く考慮する必要があると考えられる．したがって，階層ごとに適したスパン表現の獲得方法を模索することは今後の課題である．また，本研究では LSTM を用いて一次元的にテキストを読み込みベクトル表現を獲得し

たが，Tree-LSTM のような構造を考慮可能なネットワークもスパン表現を得るための手段として考えられる．

四つ目の課題として，修辞構造木を変換して得られる係り受け木のさらなる性能改善が挙げられる．修辞構造木は隣接するテキストスパン間の関係性を表すため，木の位置に応じてスパンが表すテキストの長さは変化する．一方，係り受け木では，いずれも EDU 間の関係を表現しており，対象が一貫しているため扱いやすいため応用タスクで広く活用される．本研究では，係り受け木に強く影響を及ぼす修辞構造木の上部の性能を改善することで係り受け木の構造も改善した．ただ，前述したように応用タスクの多くで解析された修辞構造木を係り受け木へと変換して利用しているため，長期の依存関係のみに限らず，係り受け木全体を対象とした性能改善により，さらなる応用タスクの改善が期待できる．

5.2.2 部分木による疑似正解データセットを活用した修辞構造解析

二つ目の研究として取り組んだ，部分木を活用した疑似正解データセットの構築について，今後の課題を 3 点挙げる．

一つ目の課題として，疑似正解データセットを構築するために用いる教師解析器に，より高性能な解析器を採用することが挙げられる．本研究では，教師解析器として Two-Stage Parser (TSP) を活用した．これは TSP が分類器として軽量な SVM を利用しており大規模なデータ作成において高速に動作すること，また，疑似正解データセットの適用対象であるスパン分割による下向き解析器と異なり TSP が上向きの解析器であることが理由である．実験結果では，教師に用いた解析器が高い性能で解析できる修辞関係ラベルにおいて大幅な性能改善が見られたことから，教師解析器と生徒解析器の間に異なる解析器を利用して多様性を持たせることが有効であると考えられる．そこで，より高性能な上向きの解析器を教師解析器として採用することで疑似正解データセットのさらなる改善が期待される．

二つ目の課題として，疑似正解データセットのフィルタリングが挙げられる．本研究では，重複する部分木を対象としてデータ抽出を行い疑似正解データセットを構築した．この際，特定のデータをフィルタリングするという手順は行なっていない．例えば，すでに高い性能である修辞構造ラベルに関する疑似正解データをフィルタリングにより取り除くことや，応用タスクで重要な役割を果たす修辞関係ラベルに集中してデータを選択することで，疑似正解データセットを構築することも可能である．

三つ目の課題として，部分木を活用することによる抽出されるデータの偏りが挙げられる．提案手法では，複数の教師解析木によって一致した部分木を疑似正解データセットとして抽出したが，この特性上，木の上部にあたる疑似正解データの獲得が難しい．修辞構造木の評価では，評価対象である木に含まれるスパンの数は木の上部に比べて下部の占める割合が大きいため，木の下部における性能改善が評価尺度に強く反映される．前述した係り受け木への変換において，特に長い依存関係を正しく捉えるために木の上部の性能改善は重要であることから，木の上部を含めた疑似正解データセットの構築が課題となる．

5.2.3 修辞構造解析における今後の方向性

本研究では，修辞構造解析における解析性能の改善に向けて，下向きの解析方法であるスパン分割による解析手法の導入，段落を用いた解析空間の分割，部分木を活用した疑似正解データの構築を行い，実際に解析性能の改善を示した．一方で，さらなる修辞構造解析の性能改善および，応用タスクにおける修辞構造木やその係り受け木の利用に向けていくつか解決すべき課題が残っている．そこで本研究で取り組んだ二つの研究を通して得られた知見から修辞構造解析における今後の研究の方向性として「ベンチマークによる性能評価と応用

タスクにおける性能改善への寄与」を例に挙げて議論する．

解析によって得られる修辞構造木，および係り受け木は応用タスクに適用されて初めて価値を発揮するが，これまでの修辞構造解析における研究の多くは，RSTDT コーパスを用いて学習，評価および比較を行なってきた．これは，RSTDT を用いたベンチマークにおいて性能が改善できれば，それを利用した応用タスクの性能も改善することを前提としているためである．したがって，現在の修辞構造解析器の評価は RSTDT コーパス上での解析性能のみを対象とした，ベンチマークとしての評価が一般的であり，それが応用タスクの性能改善にどの程度寄与するかの評価は行われていない．しかし，実際には応用タスクによって重要とされる修辞関係ラベルや構造は異なり，ベンチマークでの改善が応用タスクでの改善に直結するとは限らない．

したがって，修辞構造が応用タスクの性能改善に寄与するためには，本研究で取り組んだ解析方法やデータセットの拡張といった修辞構造解析の性能改善のほかに，まず，応用タスクの性能改善への寄与を定量化する方法が必要であると考えられる．具体的には，応用タスクの素性として用いた修辞構造の違いによる性能への影響度の計測方法や修辞構造が付与された応用タスクの評価用データセットの整備などが考えられる．これにより，応用タスクの改善を見据えた具体的な修辞構造解析の改善に向けた方向性の議論も可能になる．

謝辞

本研究を行うにあたり指導教員である奥村学教授には，研究の構想から論文執筆に至るまで終始懇切丁寧なご指導およびご鞭撻を賜りました．心より感謝申し上げます．NTT コミュニケーション科学基礎研究所の平尾努氏には，本研究に取り組む転機を頂き，終始変わらず暖かいご指導をいただきました．奈良先端科学技術大学院大学の上垣外英剛准教授には，研究のご指導のみならず，計算機環境の改善のためにご尽力賜り感謝しております．本論文の査読を引き受けてくださった熊澤逸夫先生，中山実先生，篠崎隆宏先生，船越孝太郎先生には大変有益なコメントいただきました．感謝申し上げます．研究室メンバーの皆様には大変お世話になりました．また，研究室秘書の飯山信子氏にはあらゆる事務処理でお世話になりました．感謝申し上げます．

参考文献

Kenji Abe, Shinji Kawasoe, Tatsuya Asai, Hiroki Arimura, and Setsuo Arikawa. Optimized substructure discovery for semi-structured data. PKDD '02, page 1 – 14, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3540440372.

Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018. URL http://arxiv.org/abs/1803.08375.

Stefanos Angelidis and Mirella Lapata. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31, 2018. doi: 10.1162/tacl_a_00002. URL https://aclanthology.org/Q18-1002.

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*, pages 1903–1913, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1179.

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*, pages 292–304, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1028.

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. URL https://www.aclweb.org/anthology/W01-1605.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?

id=Hk95PK9le.

David duVerle and Helmut Prendinger. A novel discourse parser based on support vector machine classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 665–673, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1075.

Vanessa Wei Feng and Graeme Hirst. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1048. URL https://aclanthology.org/P14-1048.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052. IEEE, 2005.

Grigorii Guz and Giuseppe Carenini. Coreference for discourse parsing: A neural approach. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 160–167, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.codi-1.17. URL https://aclanthology.org/2020.codi-1.17.

Grigorii Guz, Patrick Huber, and Giuseppe Carenini. Unleashing the power of neural discourse parsers - a context and structure aware approach using large scale pretraining. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3794–3805, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.337. URL https://aclanthology.org/2020.coling-main.337.

Michael Heilman and Kenji Sagae. Fast rhetorical structure theory discourse parsing, 2015.

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. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf.

Hugo Hernault, Helmut Prendinger, David A du Verle, and Mitsuru Ishizuka. Hilda: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3):1–33, 2010.

Tsutomu Hirao, Masaaki Nishino, Yasuhisa Yoshida, Jun Suzuki, Norihito Yasuda, and Masaaki Nagata. Summarizing a document by trimming the discourse tree. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(11):2081–2092, 2015. doi: 10.1109/TASLP.2015.2465150.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

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)*, pages 2306–2316, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1235. URL https://www.aclweb.org/anthology/D19-1235.

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)*, pages 7442–7457, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.603. URL https://aclanthology.org/2020.emnlp-main.603.

Tatsuya Ishigaki, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. Discourse-aware hierarchical attention network for extractive single-document summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 497–506, Varna, Bulgaria, September 2019. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_059. URL https://aclanthology.org/R19-1059.

Yangfeng Ji and Jacob Eisenstein. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1002. URL https://aclanthology.org/P14-1002.

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*, pages 2603–2614, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1245.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 486–496, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1048.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. CODRA: A Novel Discriminative Framework for Rhetorical Analysis. *Computational Linguistics*, 41(3):385–435, 09 2015. ISSN 0891-2017. doi: 10.1162/COLI_a_00226. URL https://doi.org/10.1162/COLI_a_00226.

D. Keselman and A. Amir. Maximum agreement subtree in a set of evolutionary trees-metrics and efficient algorithms. SFCS '94, page 758 – 769, USA, 1994. IEEE Computer Society. ISBN 0818665807. doi: 10.1109/SFCS.1994.365717. URL https://doi.org/10.1109/SFCS.1994.365717.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. Top-down rst parsing utilizing granularity levels in documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8099–8106, Apr. 2020. doi: 10.1609/aaai.v34i05.6321. URL https:

//ojs.aaai.org/index.php/AAAI/article/view/6321.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. Top-down discourse parsing via sequence labelling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 715–726, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.60. URL https://aclanthology.org/2021.eacl-main.60.

Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, and Manabu Okumura. Considering nested tree structure in sentence extractive summarization with pre-trained transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4039–4044, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.330. URL https://aclanthology.org/2021.emnlp-main.330.

Qi Li, Tianshi Li, and Baobao Chang. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1035. URL https://aclanthology.org/D16-1035.

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*, pages 4190–4200, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1410. URL https://www.aclweb.org/anthology/P19-1410.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach. 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

William C. Mann and Sandra A. Thompson. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, Information Sciences Institute, June 1987.

Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization.* MIT Press, November 2000. ISBN 9780262133722. URL https://mitpress.mit.edu/9780262133722/the-theory-and-practice-of-discourse-parsing-and-summarization/.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

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*, pages 1319–1324, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1136. URL https://aclanthology.org/D17-1136.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. RST parsing from scratch. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computa-*

tional Linguistics: Human Language Technologies, pages 1613–1625, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.128. URL https://aclanthology.org/2021.naacl-main.128.

Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. Instance-based learning of span representations: A case study through named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6459, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.575. URL https://aclanthology.org/2020.acl-main.575.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

Mitchell Stern, Jacob Andreas, and Dan Klein. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1076. URL https://aclanthology.org/P17-1076.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf.

Wenhui Wang and Baobao Chang. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1218. URL https://aclanthology.org/P16-1218.

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)*, pages 184–188, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2029. URL https://www.aclweb.org/anthology/P17-2029.

Yizhong Wang, Sujian Li, and Jingfeng Yang. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1116. URL https://aclanthology.org/D18-1116.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text sum-

marization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.451. URL https://aclanthology.org/2020.acl-main.451.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

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*, pages 559–570, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1047.

M.J. Zaki. Efficiently mining frequent trees in a forest: algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1021–1035, 2005. doi: 10.1109/TKDE.2005.125.

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*, pages 6386–6395, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.569. URL https://www.aclweb.org/anthology/2020.acl-main.569.

Longyin Zhang, Fang Kong, and Guodong Zhou. Adversarial learning for discourse rhetorical structure parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3946–3957, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.305. URL https://aclanthology.org/2021.acl-long.305.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1101. URL https://aclanthology.org/P17-1101.

研究業績

本論文に関する業績

論文誌 (査読あり)

1. 小林尚輝，平尾努，上垣外英剛，奥村学，永田昌明，”疑似正解データを活用したニューラル修辞構造解析”，自然言語処理，29-3, pp.875 - 900 (2022)

国際会議論文 (査読あり)

1. Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura and Masaaki Nagata, ”Top-down RST Parsing Utilizing Granularity Levels in Documents”, *In proceedings of the AAAI Conference on Artificial Intelligence (AAAI2020)*
2. Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura and Masaaki Nagata, ”Improving Neural RST Parsing Model with Silver Agreement Subtrees”, *In proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL2021)*

国内会議論文 (査読なし)

1. 小林尚輝，平尾努，上垣外英剛，奥村学，永田昌明，”階層構造を考慮したトップダウン談話構造解析”，言語処理学会 第 25 回年次大会 (2019)
2. 小林尚輝，平尾努，上垣外英剛，奥村学，永田昌明，”疑似正解データを利用した修辞構造解析器の改善”，言語処理学会 第 27 回年次大会 (2021)

その他の業績

国際会議論文 (査読あり)

1. Naoki Kobayashi, Tsutomu Hirao, Kengo Nakamura, Hidetaka Kamigaito, Manabu Okumura and Masaaki Nagata, ”Split or Merge: Which is Better for Unsupervised RST Parsing?”, In *proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2019)*
2. Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, Hiroya Takamura and Manabu Okumura,

”Bridging between emojis and kaomojis by learning their representations from linguistic and visual information”, *In proceedings of the Conference on Web Intelligence (WI2019)*

3. Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, Hiroya Takamura, Manabu Okumura, ”Making Your Tweets More Fancy: Emoji Insertion to Texts”, *In proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP2021)*
4. Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito and Manabu Okumura, ”Considering Nested Tree Structure in Sentence Extractive Summarization with Pre-trained Transformer”, In *proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2021)*
5. Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura and Masaaki Nagata, ”A simple and Strong Baseline for End-to-End Neural RST-style Discourse Parsing”, In *findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing (EMNLP2022)*

国内会議論文 (査読あり)

1. 平尾努，小林尚輝，上垣外英剛，奥村学，木村昭悟，”動画談話構造解析に向けたデータセット構築”，第 25 回 画像の認識・理解シンポジウム (2022)

国内会議論文 (査読なし)

1. 小林尚輝，平尾努，中村健吾，上垣外英剛，奥村学，永田昌明，”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”，言語処理学会 第 25 回年次大会 (2019)
2. 小林尚輝，平尾努，上垣外英剛，奥村学，永田昌明，”言語モデルと解析戦略の観点からの修辞構造解析器の比較”，言語処理学会 第 28 回年次大会 (2022)
3. 小林尚輝，真鍋陽俊，小田悠介，”高速な契約書レビューのための計算量の削減”，言語処理学会 第 28 回年次大会 (2022)

受賞

- 東京工業大学工学院 情報通信系優秀学生賞（修士） (2020)
- 若手奨励賞，小林尚輝，”疑似正解データを利用した修辞構造解析器の改善”，言語処理学会 第 27 回年次大会 (2021)