

並列制約付きポインタネットワークによる議論スレッドの自動構造化

Parallel Constrained Pointer Network for Automatic Structurization of Discussion Threads

森尾 学 †

† 東京農工大学大学院 工学府

藤田 桂英 ‡

‡ 東京農工大学大学院 工学研究院

1 はじめに

近年、オンラインでの市民議論により、膨大な人数の市民が場所や時間を問わず、議論を通じて合意を導くようなことが可能になっている [8]. それに応じて、膨大な数の投稿の整理や議論の流れの理解を支援するために、Argumentation 理論に重点を当てる Argument Mining (AM) [12, 21] の適用が求められている. なぜならば、Argument Mining は Opinion Mining と異なり、主張 (Claim) や前提 (Premise) [1] といった Argument Component (AC) の構造化が行えるため、意見だけでなくその裏付けとなる前提の解釈が可能であるという利点があるためである.

AM は、議論的言説から主張や前提となる部分を抽出し、それらの関係を推論して構造化する自然言語処理の研究領域である. [29] によれば、AM は以下の 3 つのサブタスクに分割できる.

- **Component identification** は文書を非 Argumentative (NA) と Argumentative ユニットに分割し、AC の範囲を特定することに焦点を当てる.
- **Component classification** は AC を主張 (claim) や前提 (premise) のような役割に分類する.
- **Structure identification** は AC 同士のリンクを抽出することに焦点を当てる.

最後の Structure identification はマイクロレベルとマクロレベルに分類する事ができる. 一般的に、AC 同士の関係を抽出する Structure identification はマイクロレベルに分類され、マクロレベルは完全な Argument 同士の関係を特定することが多い. [28, 29] はマイクロレベルで AC 同士のアノテーションを行った. 彼らは、この 3 つのサブタスクを用いて「One-claim アプローチ」をエッセイデータに適用した. すなわち、一つのエッセイを Argument (主張をルートとする 0 個以上の前提の木構造) の集合で表現した. マクロレベルに比べてマイクロレベルのアプローチは、研究材料が多く存在することから、[29] のコーパスは多くの AM の応用研究を支えている [19, 20, 30, 3, 23, 3, 24]. このように、AM はコーパスの発展に応じて技術的な発展を遂げてきた.

しかしながら、議論スレッドに対して積極的な Argument Mining のアノテーションが行われてきたとは言いがたい. 関連研究として [4] は、Target と Callout を定義し、投稿と投稿の関係を提案した. しかし、彼らの Target や Callout は完全な argument 同士 (マクロレ

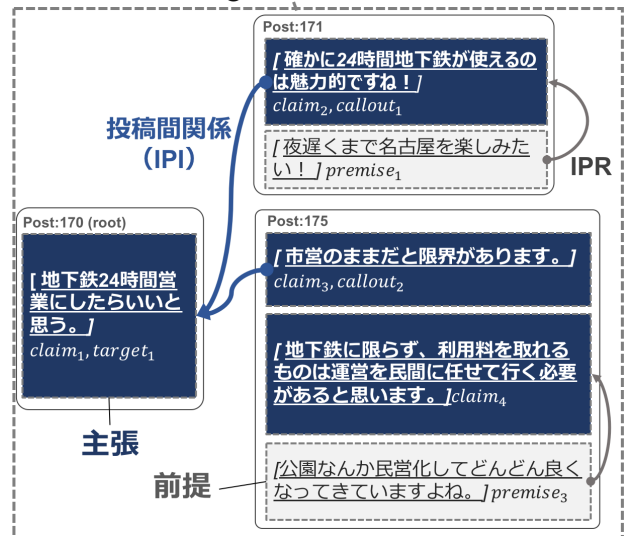
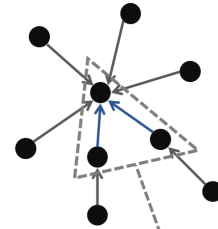


図 1: 議論スレッドのアノテーション例. 最上部のグラフはスレッドの返信構造を表している. それぞれの投稿は Argument を 0 個以上持つ. Argument とは 1 つの主張 (claim) に対して 0 個以上の前提 (premise) が付随する木構造である. IPR は主張や前提間の投稿内関係を表し、IPI は返信関係にある投稿同士の AC の関係である.

ベル) の関係のみを扱っており、投稿と投稿の間に存在するマイクロレベルな AC 同士の関係には触れていない. [6] では、投稿とそれに返信する投稿との間の AC の関係をアノテーションしたが、彼らはスレッド全体を 1 つの言説とみなしている (例えば、ある投稿の前提が別の投稿の主張に関係しているケースを許している). 我々は投稿ごとに書き手が異なる状況において、投稿単体を独立した言説とみなすことが望ましいと考えているため、彼らのスキームを適用することはできない. さらに、[6] のデータセットは 78 スレッドしかアノテーションされていないため、最先端の深層学習モデル [25, 2, 26, 16] を適用することが難しい.

AM の目標とするところはコーパスを用いて主張や前提といった AC の自動構造化を行うことである. 今まではエッセイなどのデータに対して自動構造化に向け

†morio@katfujilab.tuat.ac.jp

‡katfujilab@cc.tuat.ac.jp

た技術的な研究が行われてきたが、議論スレッドに対する研究も求められている。しかしながら、議論スレッドに対する AM の技術的な研究は多くない。近年においては AM における深層学習による End-to-End (E2E) な識別モデルが登場している [2, 25] もの、議論スレッドに対する研究は我々の知る限りで存在しない。

そこで、本研究は議論スレッドにおける AM に取り組む。本研究の貢献は次のとおりである: (1) スレッド構造を持つオンライン市民議論データに対して投稿内と投稿間の関係に着目したマイクロレベルなアノテーションを行う。(2) 「並列制約付きポインタネットワーク」を用いて、文のタグ付け、投稿内関係および投稿間インタラクションを識別する手法を提案する。この並列制約付きポインタネットワークは学習範囲を制限した 2 つのポインタネットワークを用いて効果的に学習を行うことができる新しい手法である。最先端の既存手法と比較した評価実験により、提案手法が関係の推論に対して高い精度を示すことが分かった。

我々の並列制約付きポインタネットワークは従来手法に比べて大きな技術的向上を果たした。というのも、従来の単純なポインタネットワークでは探索空間に制約がかかっていないために、スレッド構造に対して無駄な学習を行う必要があったためである。比べて、我々のモデルはスレッドの制約条件を学習時に取り入れることにより、関係の推論に対して高い精度で構造化できるという特徴がある。

2 関連研究

AM においては議論構造を自動的に識別する研究が盛んである。例えば、投稿に対する議論要素の特定 [11, 1] やリンク抽出 [24, 2, 25] の AM タスクの研究が存在する。

近年では End-to-End (E2E) な識別モデルが目下進行中の研究対象である。E2E 学習が注目されている理由は、複数の識別器を直列に用いる場合に比べて誤差が伝播しにくいためである [2]。例えば [2] は LSTM を用いて AM に対するマルチタスク学習 [26, 13] を行う手法や、LSTM-ER [16] を適用する手法を提案した。また、ポインタネットワーク (PN) を用いた関係抽出の試みが近年注目を集めている [32, 9]。[25] は Sequence-to-Sequence モデルを取り入れた PN によってエッセイの自動構造化手法を提案した。PN を用いた手法は AM において最先端の精度を達成しているため、本研究でも Potash らの手法の適用を試みた。しかしながら関係の識別に対して探索空間が広く、精度が上がらないという問題が確認された。

3 Argument Mining のための議論スレッドのアノテーション

3.1 データ

本研究でアノテーションするデータは、COLLAGREE [8, 18] と呼ばれる市民フォーラムで行われた市民議論デー

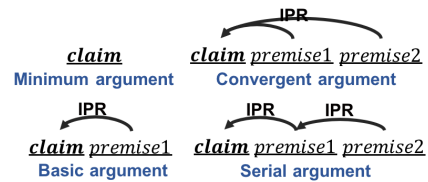


図 2: Argument の例。

タである。このデータは名古屋市と共同で 2016 年 12 月から 2017 年 1 月にかけて行われた Web 議論をベースとしている。今回収集したスレッドは 399 個で [6] の 78 スレッドと比べて量を増やすことができた。さらに、1 つの投稿あたりの文章数が平均で 4.19 個と、1 投稿あたりの情報量が多いことがわかった。

3.2 スキーム

AM に対する堅実なスキームは数多く存在しているため既存のスキームを組み合わせ、それを発展させることがスレッド構造に新たなスキームを適用させるための鍵であると言える。前提として COLLAGREE では 1 つの投稿あたりの文章数が多い。これは一つの投稿が何かしらの主張や前提を十分に述べていることを示している。そこで、[6] のように投稿を区別しないアノテーションを行うよりも、1 つの投稿そのものを単独の議論的言説とみなしてアノテーションすることが望ましいと考えられる。よって、AM における投稿内のスキームを提案し、さらにその上で投稿間のスキームを組み合わせることによって、我々独自のスキームを提案することにした。

投稿内のスキームについては、[28] のマイクロレベルアプローチを用いた。すなわち、1 つの投稿は複数（もしくは空）の Argument で構成される。1 つの Argument は "one-claim" アプローチである（ただし今回は Major claim は導入しない）。すなわち、1 つの Argument には必ず 1 つの主張と 0 個以上の前提が含まれる。以下は Argument の各要素の定義である：

- **主張 (Claim)** とは議論の的となる AC である。投稿内には関係先を持たない。
- **前提 (Premise)** は主張や他の前提に対して理由を与えたり支持や攻撃する AC のことである。
- **投稿内関係 (IPR)** は投稿の中での主張や前提の関係である。関係を (*parent* ← *child*) と表す時、*child* は *parent* に対して理由を与える。そのため、あり得るパターンは (*Claim* ← *Premise*) もしくは (*Premise* ← *Premise*) の 2 通りしか存在し得ない。

木構造を保つために 1 つの Argument は主張をルートノードとして、連結グラフかつ非 Cycle グラフでなくてはならない（例えば、前提だけでループを作ることはいできない）。図 2 は、投稿に含まれる Argument の例である。最もシンプルな例は主張だけを要素を持つ Minimum argument である。また、主張が前提を複数持

つ Convergent や、前提が前提を持つ Serial などが存在する。

なお、本論文においては、Component はルールベースで文ごとに分割することにした。すなわち、(主張, 前提, 非 Argumentative(NA)) の 3 つはそれぞれ、文章単位でタグ付けされることになる。また IPR は文章と文章の間の関係とみなせる。

続いて、投稿間のスキームを導入する。なぜならば投稿内のスキームだけでは、投稿と投稿の間のインタラクションを捉える事ができないためである。特に、オンライン市民議論において Argumentation の可視化が重要になる場面では、市民同士の主張の関係を捉えたいという要望がある。そこで、投稿間を結びつけるインタラクションに対して [4] のアプローチを組み込むことにした。彼らは投稿間のマクロレベルな関係を Target と Callout を用いてアノテーションした。しかし我々の研究では、投稿内においてミクロレベルなアノテーションを行うため、投稿間の AC 同士のミクロレベルなインタラクションを定義する。以下は我々の Target と Callout およびミクロレベルな投稿間インタラクション (IPI) の定義である:

- **Target** は投稿間の AC と AC のインタラクションにおいて、後に呼び出される対象となる要素である*。
- **Callout** は投稿間の AC と AC のインタラクションにおいて、Target に対してコメントを与える AC である。明示的な返信関係にある 2 つの投稿間のみを考慮する。本研究では議論に焦点を当てるため、Callout は主張に限定する[†]。
- **投稿間インタラクション (IPI)** は 2 つの返信関係にある投稿同士の関係である。関係を (*parent* ← *child*) と表す時、*child* は Callout、*parent* は Target である。

図 1 は以上の「投稿内」と「投稿間」のスキームを用いて、とある議論スレッドをアノテーションした例である。例えば、Post:170 は *claim*₁ をルートノードとする Minimum argument の構造を持っている。また Post:170 の *target*₁ と Post:175 の *callout*₂ が投稿間インタラクション (IPI) の関係になっていることが分かる。

3.3 アノテーション結果

大学生 11 人に COLLAGREE の 399 個のスレッドをアノテーションさせた[‡]。

表 1 はアノテーションされた最終的なコーパスである。SD は標準偏差を表す。また便宜のため、投稿レベルの統計情報も併記した。アノテーションの結果、4221 個の AC (うち 1449 個が主張、2762 個が前提) が得ら

*Target を含む投稿が別の投稿に対して返信していれば、同時に Callout にもなりうる。

[†]Callout を主張に限定することによって、主張から出る関係先が最大一つになり、問題がシンプルになるという利点がある。

[‡]実際には、時間と予算の都合でデータセットを 4 分割し、それぞれの分割において 3 人ずつ人員を割り当てた。

		数	avg. per post	SD
general	スレッド	399	-	-
	投稿	1327	-	-
	Component	5559	4.19	3.33
content	主張	1449	1.09	0.67
	前提	2762	2.08	2.36
	NA	1348	1.02	2.19
rel	IPR	2762	2.08	2.36
	IPI	745 (C ← C : 574, P ← C : 109, NA ← C : 62)	0.56	0.62

表 1: 最終的なコーパスと統計。

れた。主張より前提が多い現象は、書き手が観点のロバスト性を担保するために複数の理由を述べる傾向にあるためであり、Argumentation の分野において共通である [17]。よって、我々の COLLAGREE データにおいても、Argument Mining 特有の事象を再現することができた。さらに各投稿は平均で 1.09 個の主張を持つため、1 つの投稿は平均で 1 つ以上の Argument を保持しており、前提は平均 2.08 個であった。よって、一つの投稿を独立した議論的言説とみなすことが有効な仮定であることを確認することができた。投稿間インタラクション (IPI) において、Target が主張の場合は 574 個、前提の場合は 109 個であった。本研究では Callout の要素タイプは主張のみであることに留意すると、IPI が主張と主張を結ぶケースが全体の 77% を占めていることが分かる。すなわち、IPI は極めて議論的であると言える。

4 議論スレッドのための自動構造化手法

本節では、アノテーションしたデータに対して、どのようにして自動構造化を行うかについての方法論を述べる。

4.1 スレッド構造の系列表現

議論スレッドはそれぞれものが議論の単位となっていることが多い。そのためモデルの入力系列はスレッド単位が望ましいと言える。そこで、スレッド構造を入力系列として表現するための、モデリング方法について述べる。

前述の通り、1 つのスレッドには複数の投稿が格納されており、1 つの投稿には複数の文章が格納されている。さらに、当然ながら 1 つの文章は複数の単語で構成されている。我々は、既存研究の系列モデリング手法 [2, 25] を拡張することにした。初めに入力の単位を文章表現とする。なぜならば、各 Component のタグ (主張, 前提, NA) は文章ごとにアノテーションされているためである。続いて、スレッドでの投稿を階層の深さ順にならべ、各階層で時系列順に並べる。また各階層の区切りと各投稿の区切りにそれぞれ区切り表現を挿入し、学習させやすくする。このようにすることで、LTSM [7] への入力を可能にすることが出来るほか、階層構造や返信関係を系列で表現できる。議論スレッドを入力系列としてモデリングする例を図 3 に示

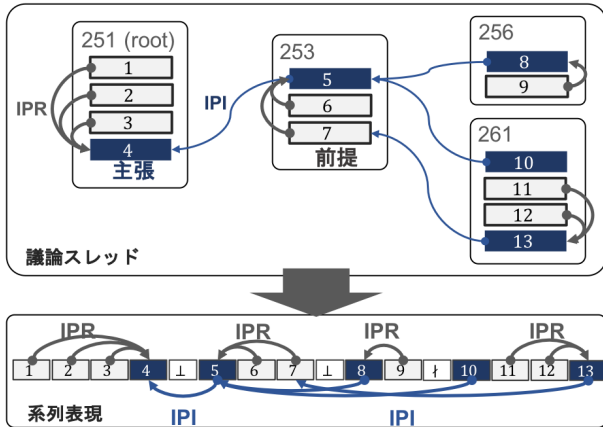


図3: スレッド構造を入力系列にモデリングする例。図の上部は議論スレッドの例。下部はその系列表現である。最優先順序は階層で、その次の優先順位は時系列となっている。⊥は階層の区切り表現、Iは投稿の区切り表現である。矢印は便宜のために表示していることに留意。

した。

4.2 並列制約付きポインタネットワーク

関連研究でも述べたように、ポインタネットワーク(PN)を用いた手法がAMにおいて最先端の精度を達成している。そこで我々はPNをベースにした手法の適用を試みた。しかしながら、単純なPNではスレッド構造の制約を満たさない部分(例えば、投稿外に向かうIPRや、返信関係のない投稿間のIPI)をも探索空間に含むため、識別精度が上がりにくいという問題がある。すなわち、スレッド構造等の制約を用いることで、大幅な精度の向上を期待できると考えられる。例えばスレッドの投稿に含まれる文章の関係先を探す場合に、スレッド全体を探索する必要は無い(その投稿内の文章のみ探索すれば良い)。そこで、我々はこのようなスレッド構造特有の制約を取り入れたニューラルモデル「並列制約付きポインタネットワーク」を提案する。

入力文章の埋め込み

初めに、並列制約付きポインタネットワークの入力について説明する。入力となる N 個のスレッド(T_1, \dots, T_N)が与えられた時に、 T_i に含まれる階層順かつその上で時系列順の投稿を($P_1^{(i)}, \dots, P_{N_i}^{(i)}$)と表す。ただし N_i は T_i に含まれる投稿数のことである。さらに、 $P_j^{(i)}$ に含まれる文章を($S_1^{(i,j)}, \dots, S_{N_{i,j}}^{(i,j)}$)と表す。ただし $N_{i,j}$ は $P_j^{(i)}$ に含まれる文章数である。また、区切り表現は含まない。まずはじめに、投稿 $P_j^{(i)}$ の k 番目の文章に出現する n 番目の単語の埋め込みベクトル w_n が与えられると、LSTMに入力される文章表現は次の式によって獲得される。

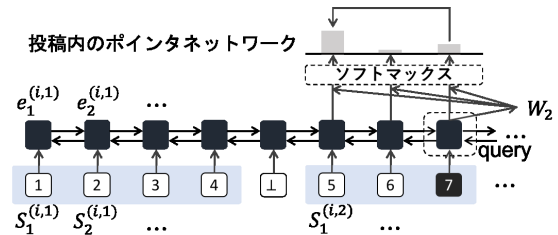


図4: 投稿内関係(IPR)の制約付きポインタネットワーク

$$A_k = \sum_n w_n \quad (1)$$

ただし、 w_n の獲得にはWord Embedding [15]やBag-of-words (BoW)などを用いる。今回は埋め込みベクトル E をモデルのパラメータとして学習させた。 A を入力として、エンコードの過程でLSTMは隠れ状態 e_i を出力する。

本研究の最終的な目的は、各文章のComponentタイプとIPRとIPIをE2Eに識別することである。ここでいうComponentタイプとは、(主張, 前提, NA)の3分類のことである。並列制約付きポインタネットワークでは、LSTMの中間ベクトルの各ステップ e_i を用いてこれらのE2Eな識別を行う。そのため識別の種類ごとにComponentタイプ、IPR、IPIの合計3つの出力層を用意する*。IPRとIPIの識別器においては別々のパラメータを用いてスレッド構造の制約に特化したポインタネットワークを構築する。このことが「並列制約付きポインタネットワーク」の所以である。

投稿内関係(IPR)の識別手法

本節では、3つの識別器のうちのIPRの識別を行う出力器を導入する。まずはじめに、双方向LSTM[5]を用いてスレッドのエンコードを行う。よって e_i は順方向の中間ベクトル \vec{e}_i と逆方向の中間ベクトル $\overleftarrow{e}_{n-i+1}$ を結合したベクトルによって表される。なお、便宜のために投稿 $P_j^{(i)}$ における双方向LSTMの隠れ状態を($e_1^{(i,j)}, \dots, e_{N_{i,j}}^{(i,j)}$)と表す。

通常のPNは、エンコードされた全ての隠れ状態 e_i を用いて関係先の探索を行うが、IPRにおいては、文章は投稿内の高々一つの文章にしか関係しない。そこで、投稿内の文章の各ステップで出力されたLSTMの中間表現のみを利用して関係先を探索するアプローチを取る。例えば、図4はスレッド T_i のエンコーディングとIPRの識別モデルを表している。ここで投稿 $j=2$ の3番目の「7」と書かれた文章の関係先を識別しましょう。この文章の関係先を求めるためには、その投稿内の文章の中間ベクトル($e_1^{(i,2)}, e_2^{(i,2)}, e_3^{(i,2)}$)のみを用いれば良い。Potashらの手法[25]では、全ての e_i を用いるため、我々の手法によって大幅に探索空間を狭める

*制約条件より、各文章から伸びる関係先は高々一つであるので、最低1つの関係識別器だけあればよい。比較対象となるSTagBLSTMとPNを用いる手法は1つの関係識別器しか持たせないようにした。

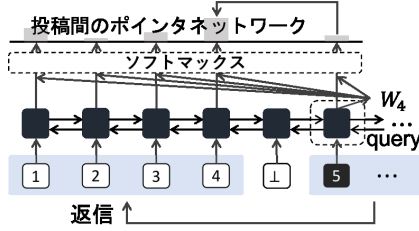


図 5: 投稿間インタラクション (IPI) の制約付きポインタネットワーク

事ができる。

IPR の出力部分をより汎的に説明すれば、PN の Attention のパラメータ W_1, W_2, v_1 を用いて、

$$u_l^{(i,j,k)} = v_1^T \tanh(W_1 e_l^{(i,j)} + W_2 e_k^{(i,j)}) \quad (2)$$

を得る。 $u_l^{(i,j,k)}$ は、 $P_j^{(i)}$ における k 番目の文章が l 番目の文章に関係する度合いを示す。また、 $e_k^{(i,j)}$ はクエリベクトルとみなせる。もし、関係先が存在しなければ Self-loop を作る。すなわちその文章自身が関係先になるように学習させる。式 2 は実数値であるが、softmax 関数を用いて確率分布に変換すれば、 $p(y_k^{ipr} | P_j^{(i)}) = \text{softmax}(u_l^{(i,j,k)})$ となる。すなわち、この確率分布は k 番目の文章が投稿内の文章 l に関係する確率が 0~1 で表現されていると言える。よって、スレッド T_i における IPR の対数尤度は、 T_i に含まれる全ての IPR の対数尤度の和であるから：

$$L_i^{ipr} = \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i,j}} \log p(y_k^{ipr} | P_j^{(i)}) \quad (3)$$

投稿間インタラクション (IPI) の識別手法

3 つの識別器のうちの IPI の識別器を導入する。本研究の Target と Callout の問題設定からも明らかであるが、IPI は、投稿と投稿の間の返信関係のみに存在する。そのため、IPR の場合と同様に、LSTM の全ての中間ベクトルを用いて関係先を探索する必要がない。

そこで、スレッド T_i 内の投稿間における返信関係のペア (*parent, child*) の集合を用いて出力層を設計することにした。具体的には、スレッド T_i における集合 $R^{(i)} = \{(j_1, j_2), \dots\}$ 、(ただし $j_1 \neq j_2 \wedge j_1 < j_2$) を導入する。 j_1 は親の投稿インデックス、 j_2 は返信投稿のインデックスを表す。なおスレッド T_i にルート投稿のみしか存在しない場合は、 $R^{(i)} = \emptyset$ である。

図 5 は、IPI における、返信投稿の文章のデコード例を示している。「5」番の文章の Target を求める際には、その返信先の投稿 ($e_1^{(i,1)}, \dots, e_4^{(i,1)}$) が探索空間になる。また、返信先の投稿のどの文章にも Target が存在しない時には、Self-loop を作れるようにゲートとなるアテンションベクトルを付加する。図 5 の最上部の確率分布「Interaction Pointer Distribution」の左 4 つの出力確率は Target となる文章の確率であり、最右の出力確率は Self-loop の確率である。

式 2 では、 $e_k^{(i,j)}$ がクエリベクトルであったことを思い出し、子投稿の文章をクエリベクトルとすることを考える。投稿間関係のポインタネットワークの Attention のパラメータ W_3, W_4, v_2 を用いて、

$$q_l^{(i,j,k)} = v_2^T \tanh(W_3 e_l^{(i,j)} + W_4 e_k^{(i,j)}) \quad (4)$$

と表される。投稿ペアが (j_1, j_2) と仮定して、子投稿である $P_{j_2}^{(i)}$ の k 番目の文章の Target を求めよう。親投稿の Attention ベクトルが $q^{(i,j_1,k)}$ と表され、かつ子投稿のデコード対象となる文章のクエリベクトルが $q_k^{(i,j_2,k)}$ と表されるので、これを結合して、 $[q^{(i,j_1,k)}; q_k^{(i,j_2,k)}]$ が得られる。このベクトルを拡張するプロセスは、[14] と同値である。最終的にスレッド T_i の対数尤度を求めると、

$$p(y_k^{ipi} | P_{j_1}^{(i)}, P_{j_2}^{(i)}) = \text{softmax}([q^{(i,j_1,k)}; q_k^{(i,j_2,k)}])$$

$$L_i^{ipi} = \sum_{(j_1, j_2) \in R^{(i)}} \sum_{k=1}^{N_{i,j_2}} \log p(y_k^{ipi} | P_{j_1}^{(i)}, P_{j_2}^{(i)}) \quad (5)$$

となる。式 5 は、スレッドに含まれる全ての返信ペア $R^{(i)}$ に対して対数尤度の和をとっていることになる。

End-to-End 学習

最後に、LSTM エンコーダのそれぞれのタイムステップにおいて、Component タイプを求める分類器について述べ、最終的な尤度を求める。Component タイプの出力層は [25] と同様に、各タイムステップにおいて文章を主張、前提、NA に分類するだけである。 $P_j^{(i)}$ における k 番目の文章の AS タイプの出力を得るとしよう。このとき、モデルのパラメータ W_{type}, b_{type} による全結合層を用いて $z_k^{(i,j)} = W_{type} e_k^{(i,j)} + b_{type}$ を計算し、softmax 関数を用いてマルチクラス分類として扱う。すなわち、次のように対数尤度を計算する事ができる。

$$p(y_k^{type} | P_j^{(i)}) = \text{softmax}(z_k^{(i,j)})$$

$$L_i^{type} = \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i,j}} \log p(y_k^{type} | P_j^{(i)}) \quad (6)$$

我々は E2E に IPR, IPI, AS タイプの識別を行う。よって、モデルの最終的な損失関数は、全てのスレッド T_i において、この 3 つの識別に必要な負の対数尤度を組み合わせるものによって表現される：

$$Loss = \frac{1}{N} \sum_i (-\alpha L_i^{ipr} - \beta L_i^{ipi} - (1 - \alpha - \beta) L_i^{type}) \quad (7)$$

ただし、 $\alpha, \beta \in [0, 1] \wedge \alpha + \beta < 1$ であり、学習を制御するためのハイパーパラメータである。

5 並列制約付きポインタネットワークの評価実験

本節では、スレッド構造に対する我々の並列制約付きポインタネットワークの有効性を検証する。

モデルタイプ	モデル	Component タイプ分類			関係抽出			
		主張 F1	前提 F1	NA F1	IPR 適合率	IPR F1	IPI 適合率	IPI F1
Joint learning	Our Model	58.5	68.7	36.0	33.8	*40.8	19.6	*24.8
	Our Model - Hyp	58.1	71.5	58.8	*45.8	*44.3	*30.4	*26.9
	S TagBLSTM	54.2	65.6	56.9	14.3	14.9	21.0	12.6
	PN with Seq2Seq	58.3	70.8	48.6	35.7	27.2	13.0	19.4
	PN without Seq2Seq	60.1	71.3	53.1	36.6	35.0	26.5	20.8
Task specific	SVM - T	53.3	64.4	52.3	13.8	22.4	6.4	11.5
	RF - T	41.0	66.8	38.3	0	0	100	1.4
	Simple - T	41.1	66.1	38.3	0	0	0	0
Joint learning w/o separator	Our Model w/o separator	43.1	66.3	29.6	30.0	36.1	9.9	13.7
	S TagBLSTM w/o separator	51.8	66.1	55.2	13.9	14.5	16.1	10.8
	PN with Seq2Seq w/o separator	40.7	67.8	52.7	30.4	23.2	10.8	14.6
	PN without Seq2Seq w/o separator	43.4	67.6	53.7	29.5	21.1	19.0	6.0

表 2: (上): 提案手法 vs. ニューラルモデルのベースライン (%). * はウィルコクソンの符号付き順位検定における有意水準 $p < 0.01$ を示す. (中): Task specific なベースライン. (下): 区切り表現を除いたときのパフォーマンス.

5.1 評価指標

正当な評価を行うために、本節では説得力のあるメトリクスを導入する。正しく分類できたかの指標として適合率、再現率、F 値を導入した。これにはポジティブケースとネガティブケースを作る必要がある。Component タイプについては単純な分類問題であるので、特に考える必要はない。IPR については、Self-loop を除く全ての関係のペアを作る [29] ことで作成できる*。IPI についても同様にペアを作る†。

5.2 ベースライン

並列制約付きポインタネットワークの有用性を示すために比較手法となるベースラインを導入する。我々の手法はポインタネットワーク (PN) を用いているため、単純な PN のモデルと比較する必要がある。そのため最先端の Potash らの手法 [25] を主なベースラインとする。彼らの Sequence-to-Sequence な PN を (PN Seq2Seq) と呼び、非 Sequence-to-Sequence なモデルを (PN No Seq2Seq) と呼ぶことにする。

PN を用いない場合の挙動を解析するために、[2] で提案された Multi Task Learning (MTL) モデル [26] を導入した (S TagBLSTM)。このモデルは AS タイプの識別器と関係識別器の 2 つの出力層が、同じ双方向 LSTM の中間層を共有するモデルである。ただし我々の場合、BIO タギングを行う識別器は無く、トークンレベルの代わりに文章レベルでの入力を行う。

E2E な学習モデルが効果的であることを示すために、Task Specific なモデルを比較対象として用意する。そこで特徴ベースの SVM [29] を導入する (SVM - T)。T はそれぞれのタスク (Claim, Premise, IPR, IPI) 別に

*例えば、ある投稿内の文章が (S_1, S_2, S_3) であったとして、ゴールドスタンダードの IPR が $S_1 \leftarrow S_2, S_1 \leftarrow S_3$ となっている場合には、ポジティブケースは明らかに $\{S_1 \leftarrow S_2, S_1 \leftarrow S_3\}$ の 2 つである。ネガティブケースは Self-loop を含まない全てのペア、すなわち $\{S_2 \leftarrow S_1, S_2 \leftarrow S_3, S_3 \leftarrow S_1, S_3 \leftarrow S_2\}$ の 4 つである (ここでの Self-loop は $\{S_1 \leftarrow S_1, S_2 \leftarrow S_2, S_3 \leftarrow S_3\}$)。

†例えば、返信元の投稿の文章が (S_1, S_2, S_3) 、返信している投稿の文章が (S_4, S_5) とする。このとき、ゴールドスタンダードのインタラクション $S_2 \leftarrow S_5$ が存在するとする。明らかに $\{S_2 \leftarrow S_5\}$ がポジティブケースとなる。ネガティブケースは Self-loop を除く全ての組み合わせなので $\{S_1 \leftarrow S_4, S_1 \leftarrow S_5, S_2 \leftarrow S_4, S_2 \leftarrow S_5, S_3 \leftarrow S_4, S_3 \leftarrow S_5\}$ の 5 つである。

なっており、それぞれ頻度が上位 500 件の単語の BoW を特徴とした‡。同様に、ランダムフォレスト (RF - T) と、ロジスティック回帰 [22] によるモデルを用意した (Simple - T)。

PN Seq2Seq や PN No Seq2Seq, S TagBLSTM については、テストにおいて制約条件を満たさないケースは全て関係無し (Self-loop) と見なして評価することにした。すなわち、False Positive とは見なさないため、偶然 Self-loop が正解だった場合でも適合率を低下させないことになる。この条件はベースラインの適合率を有利にし、提案手法の評価を不利にするので説得力のある仮定である。

PN の再現実装は Chainer [31] を用いて行った。ハイパーパラメータは [25] の数値と同じ値を用いた§。時間の都合で、50 epochs で実験を打ち切り、その時点でのモデルを用いてテストを行った。訓練:テスト=8:2 とした。また、式 7 のハイパーパラメータは $\alpha = \beta = 1/3$ とした。ただし、 L^{type} と比べて、 L^{ipr}, L^{ipi} は全ての組み合わせの損失の和なので比較的大きな値となる。そこで、比較のためにハイパーパラメータを $\alpha = \beta = 0.15$ に調整したモデルを用意した (Our Model - Hyp)。

5.3 結果

表 2 に、各ニューラルモデルのパフォーマンスを示す。スペースの都合で再現率と一部の適合率を省略した。それぞれのモデルにおいて最良スコアを記載した。関係抽出 (IPR および IPI) の精度は提案手法が極めて高い精度を示した。特に IPR の識別精度はベースラインで最も高い PN No Seq2Seq と比べても +9.3% を達成している。この結果は、スレッド構造における制約を取り入れることが関係抽出の精度向上に大きく寄与することを示している。また、S TagBLSTM は、IPR および IPI 双方において低い F 値を示しているため、PN を

‡[29] では、feature-rich な入力を用いていたが、我々のデータセットの性質とは大きく異なるため、比較のために BoW のみを用いた。学習は 2fold のクロスバリデーションを用いて最良ハイパーパラメータを選択した。

§単語埋め込み層の次元数 512、双方向 LSTMs の隠れ層の次元数 256、PN Seq2Seq のデコーダの隠れ層の次元数 512、ドロップアウト率 0.9 [27, 33]、オプティマイザは Adam [10]、ミニバッチサイズ 16。

Model	IPR - F1	IPI - F1
Our Model	*39.6	*22.6
Our Model with Param Share	36.7	11.9

表3: パラメータ共有 ($v_1 = v_2, W_1 = W_3, W_2 = W_4$) を行った時のパフォーマンスの低下 ($p < 0.01$)

用いない手法の学習が困難であることも分かった。また、Task specific の各モデルが、Joint learning に比べてどれも低い数値となったため、E2E 学習が AM において重要な役割を果たすことが確認された。

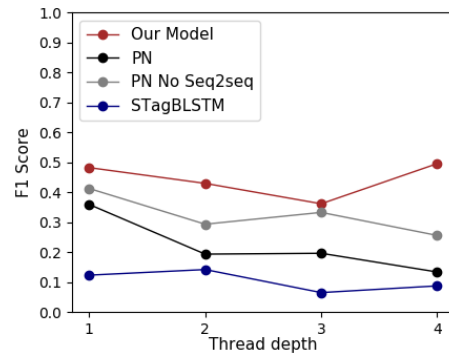
続いて、系列モデリングの節で述べた区切り表現の有効性を示すための評価を行った。表2の「w/o separator」は区切り表現を省略して学習した時のスコアを示している。PN ベースのモデルについては、区切り表現を入力に含めることで劇的な精度の改善が行われた。この結果より我々は、区切り部分の情報が LSTM にエンコードされることで、構造的な情報を保持できているのではないかと考える。逆に、興味深いことに STagBLSTM においては、区切り表現が全く精度向上に貢献しなかった。よって、構造的な入力がうまく伝達されていないことと考えられる。

続いて、我々の提案モデルが2つの並列した PN を使うことで精度が向上することを示す。すなわち、IPR の PN である式2の v_1, W_1, W_2 をそれぞれ、IPI の PN である式4の v_2, W_3, W_4 と共有したモデル (**Our Model with Param Share**) を用意した。Table 3 は F 値の平均値である (最良スコアではないことに注意)。この表は MTL 的アプローチでパラメータ共有をする必要がないことを示している。すなわち、我々の並列させたポインタネットワークを用いる手法が効果的であることが分かった。

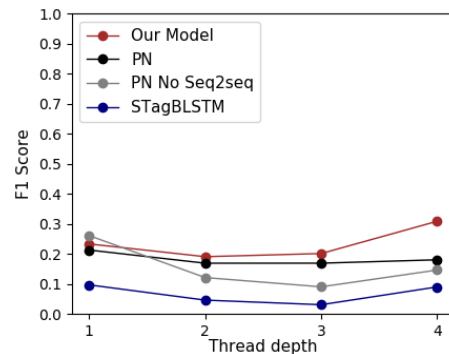
並列制約付きポインタネットワークが、どのようにスレッド構造に特化しているかを確かめるための実験を行った。具体的にはスレッドの深さと IPR と IPI の識別性能の関係を調べるため、同一のスレッドの深さを持つテストスレッドを用いた実験を行った。図6aおよび図6bに IPR と IPI それぞれの結果を示す。これらの図より、スレッドの深さが深くなるほどベースラインの IPR の識別精度が低下しているのに比べて、提案モデルはスレッドが深くなっても精度を落とさずに識別できていることが分かる。このような結果になった理由として、スレッドが深くなるほど探索空間が増大するため、Potash らの手法では冗長な識別を行わなければならない、精度が向上しないためであると考えている。逆に我々の手法はスレッドがいくら深くなっても、IPR および IPI の両方で探索空間が常に制限されるため、より深く活発なスレッドにおいて極めて有用である。

6 結論

本論文では、スレッド構造を持つオンライン市民議論データに対して、Argument Mining (AM) がどのように適用されるかを示した。初めに投稿内の提案スキーム (主張, 前提, NA, 投稿内関係) および、投稿間の



(a) 投稿内関係 (IPR) のパフォーマンス



(b) 投稿間インタラクション (IPI) のパフォーマンス

図6: スレッドの深さ (横軸) と F 値 (縦軸)。深さは少なくとも1つ返信が存在するスレッドのことである。深さが深くなっても我々の並列制約付きポインタネットワークは識別制度を保っていることが分かる。

提案スキーム (Target, Callout, 投稿間インタラクション) の概念を取り入れ、スレッド構造のための AM のマイクロレベルなスキームを提案した。続いて実際のオンライン市民議論データに対してアノテーションを行った。アノテーションの結果、極めて議論的なコーパスが得られ、自動構造化の試みを行うに至った。自動構造化の手法として我々はポインタネットワークの手法を拡張した。具体的には、スレッド構造の制約を取り入れた「並列制約付きポインタネットワーク」を提案した。最先端の手法 [25] などと比較実験を行ったところ、我々の提案モデルがスレッド構造に対して極めて有用に働くことが分かった。今後はより制約の緩い条件 (複数の関係先を持つなど) に対応して行く他、直接的な返信関係に無い投稿間のインタラクションを捉えて構造化することが課題である。

謝辞

本研究は、JST, CREST の支援を受けたものである。また、COLLAGREE による市民議論データを提供いただいた、名古屋工業大学伊藤孝行教授、秀島栄三教授、伊藤孝紀准教授、白松俊准教授に感謝する。

参考文献

- [1] Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2236–2242, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [2] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11–22, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- [3] Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 549–554, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [4] Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics, ArgMining@ACL 2014, June 26, 2014, Baltimore, Maryland, USA*, pp. 39–48, 2014.
- [5] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pp. 5–6, 2005.
- [6] Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pp. 11–21, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, Vol. 9, No. 8, pp. 1735–1780, November 1997.
- [8] Takayuki Ito, Yuma Imi, Takanori Ito, and Eizo Hideshima. Collagree: A facilitator-mediated large-scale consensus support system. In *Proceedings of the 2nd International Conference of Collective Intelligence*, 2014.
- [9] Arzoo Katiyar and Claire Cardie. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 917–928. Association for Computational Linguistics, 2017.
- [10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, Vol. abs/1412.6980, , 2014.
- [11] Marco Lippi and Paolo Torrioni. Context-independent claim detection for argument mining. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI 15*, pp. 185–191. AAAI Press, 2015.
- [12] Marco Lippi and Paolo Torrioni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, Vol. 16, No. 2, pp. 10:1–10:25, March 2016.
- [13] Héctor Martínez Alonso and Barbara Plank. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 44–53, Valencia, Spain, April 2017. Association for Computational Linguistics.
- [14] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *CoRR*, Vol. abs/1609.07843, , 2016.
- [15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.
- [16] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1105–1116. Association for Computational Linguistics, 2016.
- [17] Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artif. Intell. Law*, Vol. 19, No. 1, pp. 1–22, March 2011.
- [18] Gaku Morio and Katsuhide Fujita. Predicting argumentative influence probabilities in large-scale online civic engagement. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pp. 1427–1434, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.
- [19] Huy V. Nguyen and Diane J. Litman. Contextaware argumentative relation mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1127–1137. Association for Computational Linguistics, 2016.
- [20] Huy V. Nguyen and Diane J. Litman. Argument mining for improving the automated scoring of persuasive essays. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence, 2018.
- [21] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law, ICAIL '09*, pp. 98–107, New York, NY, USA, 2009. ACM.
- [22] Andreas Peldszus and Manfred Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 938–948, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [23] Isaac Persing and Vincent Ng. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 543–552, 2015.
- [24] Isaac Persing and Vincent Ng. End-to-end argumentation mining in student essays. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1384–1394, 2016.
- [25] Peter Potash, Alexey Romanov, and Anna Rumshisky. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1375–1384. Association for Computational Linguistics, 2017.
- [26] Anders Søgaard and Yoav Goldberg. *Deep multi-task learning with low level tasks supervised at lower layers*, Vol. 2, pp. 231–235. Association for Computational Linguistics, 2016.
- [27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, pp. 1929–1958, 2014.
- [28] Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pp. 1501–1510, 2014.
- [29] Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, Vol. 43, No. 3, pp. 619–659, 2017.
- [30] Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1882–1891. Association for Computational Linguistics, 2016.
- [31] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.
- [32] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pp. 2692–2700. Curran Associates, Inc., 2015.
- [33] Guido Zarrella and Amy Marsh. Mitre at semeval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 458–463, San Diego, California, June 2016. Association for Computational Linguistics.