

ディープラーニングを用いた論理構造推定手法の提案

The Estimation of Argument Structure using Deep Learning

木下涼 宇都雅輝 植野真臣
Ryo Kinoshita Uto Masaki Maomi Ueno

電気通信大学情報理工学研究科

Graduate school of Informatics and Engineering, The University of Electro-Communications.

In this article, we present a novel approach for parsing argumentation structures. We classify argument components using Long-Short Term Memory(LSTM). Then, we use the result of component classification as the feature of the next task, identifying argumentative relations. Finally, the proposed model globally optimizes argumentative relations using integer linear programming whose objective function is the probability estimated in relation identification. Comparative studies show that each of the proposal are effective and our model significantly outperforms the most advanced model.

1. まえがき

近年、文章中の論理構造を自動で推定する論証マイニングと呼ばれる技術が注目されている [Peldszus 14, Stab 17]. 論理構造の自動推定は自然言語処理分野の重要タスクの一つであり、多くの技術に応用されている [Ghosh 16].

論証マイニングでは、次のサブタスクを順に解くことで文章中の論理構造を推定する。

要素抽出 文章中から、論理構造の構成要素となる文または節を論理要素として抽出する。

要素分類 各論理要素を「主張」あるいは「前提」に分類する。「主張」は、著者の意見を表す要素であり、「前提」は、データや例を示すことで、他の論理要素の説得力を高める要素である。

関係分類 段落内の全ての論理要素間について因果関係の有無を分類する。ここで、要素 i が要素 j を支持する因果関係について、要素 i を要素 j の原因、要素 j を要素 i の結果と呼ぶ。

構造推定 各論理要素をノードとみなし、要素間の因果関係を有向エッジで表現するグラフ構造を推定する。具体的には、上記のサブタスクの結果を用いて、任意の制約を満たすような構造を推定する。一般に、循環が存在せず、各ノードから引かれるエッジは一本以下という制約が用いられる。

これまで、これらの個々のサブタスクのみに着目した研究は多数なされてきた [Lippi 16]. しかし、全てのサブタスクを統合した手法は、[Stab 17] に留まる。

[Stab 17] は、要素抽出、要素分類、関係分類を機械学習手法を用いて解き、それらの結果を線形計画法を用いて統合することで構造推定を行う手法を提案している。しかし、この手法において、因果関係がある箇所を正しく推定できる割合は50%程度であり、十分に精度が高いとは言えない。この手法の課題として、次の点が挙げられる。

- 1) 論理要素が「主張」、「前提」のどちらであるかは前後の文脈に依存すると考えられるが、この手法では要素分類に文脈情報を活用していない。
- 2) 因果関係の有無は、要素分類の結果に依存すると考えられるが、この手法では、要素分類と関係分類を独立に解いている。

これらの問題を解決するために、本研究では、文脈情報を活用し、要素分類と関係分類のサブタスクに依存関係を導入した新たな論証マイニング手法を開発する。具体的には、ディープラーニングの一種である Long-Short Term Memory(LSTM)を用いて、論理要素の前後の文脈を考慮した要素分類手法を提案する。また、要素分類によって推定される各論理要素が「主張」である確率推定値を関係分類の特徴量として用いることで、関係分類に要素分類の結果を反映する。

提案手法には、以下の利点が期待される。

- 1) LSTM を用いて文脈の情報を取り込むことで、要素分類の精度が向上する。
- 2) 要素分類の結果を関係分類に用いることで、関係分類の精度が向上する。
- 3) 要素分類、関係分類の精度が向上することにより、最終的な論理構造の推定精度が向上する。

本論文では、ベンチマークコーパスを用いて提案手法を [Stab 17] と比較した。結果として、提案手法を用いることで、要素分類の精度が約7%、関係分類の精度が約8%向上し、最終的な論理構造をより高精度に推定できることが示された。

2. 提案手法

2.1 提案手法の流れ

提案手法の流れを説明するために、 n 個の論理要素で構成される段落に対する提案手法の概要を図1に示す。

提案手法における要素分類 (Classifying Argument Components) は、各論理要素が「主張」である確率をその要素と前後の文から推定することで行う。ここで、入力 $S^{(i)} = (S_1^{(i)}, S_2^{(i)}, \dots, S_l^{(i)})$ は、段落中の i 番目の論理要素とその前後の文の単語を出現順に並べたベクトルであり、次元数はその単語数 l となる。従って、 $S_l^{(i)}$ は、論理要素 i とその前後の文章における l 番目の単語を表す。また、 H_i は、 $S^{(i)}$ を入力とした LSTM の隠れ層の値を用いて、計算される変数である。 f_i^c は i 番目の要素に対して、[Stab 17] で抽出している要素分類の特徴量ベクトルであり、次元数は86である。本研究では、要素 i が「主張」である確率推定値 q_i は、 H_i と f_i^c を所与としたシグモイド関数で表現できると仮定する。

関係分類 (Identifying Argumentative Relation) は、[Stab 17] の特徴量に加えて、要素分類の結果を用いて、論理要素間に因果関係が存在する確率を推定することで行う。 f_{ij}^r

連絡先: 木下涼, 電気通信大学情報理工学研究科, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, 182-8585, Japan, kinoshita@ai.is.uec.ac.jp

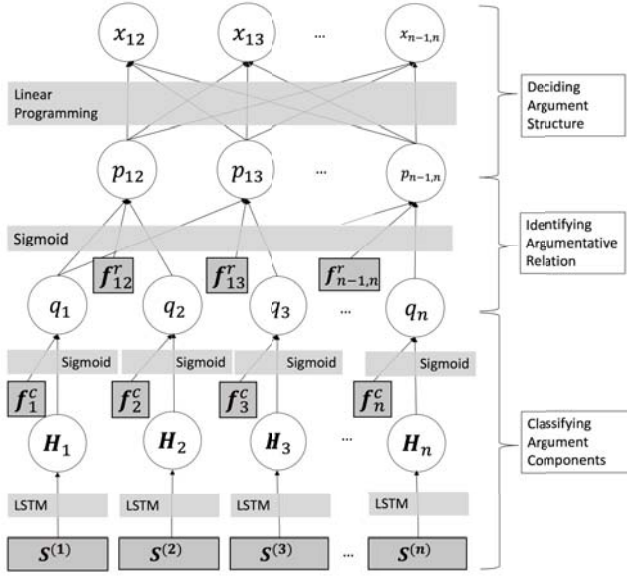


図 1: 提案手法の概要

は要素 i が要素 j の原因とした時, [Stab 17] で抽出している関係分類の特徴量ベクトルであり, 次元数は 57 である. 本研究では, 要素 i が要素 j の原因である確率推定値 p_{ij} が, 要素分類で推定された q_i , q_j と f_{ij}^r を所与としたシグモイド関数で表現できると仮定する.

構造推定 (Deciding Argument Structure) は, 循環が存在せず, 一つのノードから引かれるエッジは一本以下という制約のもと, 関係分類で推定された p_{ij} を重みとする目的関数を線形計画法を用いて解くことを行う.

以降では, 図 1 の各サブタスクについて詳細を説明する.

2.2 要素分類

提案手法では, 要素分類に論理要素の前後の文脈を利用するために, LSTM を用いる. LSTM は, 時系列データから長期の文脈を学習することのできるディープラーニング手法として, 自然言語処理分野を中心に広く活用がなされている.

LSTM の隠れ層は, 長期の依存関係を反映したものであり, 文脈の情報が保持されているとみなせる. この特徴を利用し, 提案手法では要素分類に LSTM の隠れ層に基づく特徴量を加える.

本研究では, 要素 i が「主張」である確率推定値 q_i が, $S^{(i)}$ を入力とした LSTM の隠れ層から算出される H_i と既存の特徴量 f_i^c を所与とした次のシグモイド関数で表現できると仮定する.

$$q_i = \frac{1}{1 + \exp(-w^{(1)}(f_i^c)^T - w^{(2)}H_i^T)} \quad (1)$$

$w^{(1)} = (w_1^{(1)}, w_2^{(1)}, \dots, w_{|f_i^c|}^{(1)})$, $w^{(2)} = (w_1^{(2)}, w_2^{(2)}, \dots, w_{|H_i|}^{(2)})$ は重みベクトルであり, 訓練データから学習される. 本研究では, この確率推定値を要素分類の出力とする.

LSTM の隠れ層に基づく特徴量として, 全ての隠れ層の値の平均である Average Pooling や最後の隠れ層の値が一般に用いられている [Kim 14]. そのため, 提案手法においても, Average Pooling を特徴量として用いる手法と最後の隠れ層を特徴量として用いる手法の二つを想定する.

Average Pooling を特徴量として用いる手法では, $S^{(i)}$ を LSTM の入力とした時に出力される全ての隠れ層の平均値を

特徴量 H_i^{ave} とする. ここで, H_i^{ave} の次元は, 任意の値に設定することができる. この手法では, H_i^{ave} を f_i^c と組み合わせ, 式 (1) で要素 i が「主張」である確率推定値 q_i を出力する.

一方, LSTM の最後の隠れ層を特徴量として用いる手法では, 入力 $S^{(i)}$ の最後の単語である $S_i^{(i)}$ を入力とした時の LSTM の隠れ層ベクトルを特徴量 H_i^{last} とする. ここでも, H_i^{last} の次元は, 任意の値に設定することができる. この手法では, H_i^{last} を f_i^c と組み合わせ, 式 (1) で要素 i が「主張」である確率推定値 q_i を出力する.

2.3 関係分類

論理要素間の因果関係の有無は, 個々の論理要素が「主張」であるか否かに依存すると考えられる. そこで, 提案手法では, 要素 i, j 間の関係分類を行う際に, 要素分類の出力である要素 i が「主張」である確率推定値 q_i と要素 j が「主張」である確率推定値 q_j を特徴量に加える.

本研究では, 要素 i が要素 j の原因である確率推定値 p_{ij} が, q_i , q_j , f_{ij}^r を所与とした次のシグモイド関数で表現できると仮定する.

$$p_{ij} = \frac{1}{1 + \exp(-w^{(3)}(f_{ij}^r)^T - w^{(4)}q_i - w^{(5)}q_j)} \quad (2)$$

ここで, $w^{(3)} = (w_1^{(3)}, w_2^{(3)}, \dots, w_{|f_{ij}^r|}^{(3)})$ は重みベクトル, $w^{(4)}$, $w^{(5)}$ は重みスカラーであり, 訓練データから学習される. 本研究では, この確率推定値を関係分類の出力とする.

提案手法では, 要素分類の結果を関係分類の特徴量として用いているため, ある要素が「主張」であるかを考慮した上で, 因果関係の有無を分類することができる.

2.4 構造推定

提案手法では, 要素分類の結果を関係分類で用いているため, [Stab 17] のように要素分類と関係分類の結果を統合する必要はない. 従って, 本研究では, 構造推定を次のように, 関係分類の結果のみを用いる線形計画問題として定義する.

$$\arg \max_{x \in (0,1)} \left(\sum_{i=1}^n \sum_{j=1}^n p_{ij} x_{ij} - \lambda \sum_{i=1}^n \sum_{j=1}^n x_{ij} \right) \quad (3)$$

s.t.

$$\forall i: \sum_{j=1}^n x_{ij} \leq 1 \quad (4)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} \leq n - 1 \quad (5)$$

$$\forall i: x_{ii} = 0 \quad (6)$$

$$\forall i \forall j: x_{ij} - a_{ij} \leq 1 \quad (7)$$

$$\forall i \forall j \forall k: a_{ik} - a_{ij} - a_{jk} \leq -1 \quad (8)$$

$$\forall i: a_{ii} = 0 \quad (9)$$

提案手法では式 (3) のように要素 i が要素 j の原因である確率推定値 p_{ij} を重みとする目的関数を設定し, 線形計画法を用いて, 制約を満たした上で, x_{ij} の値を決定することで最終的な論理構造を決定する.

式 (3) 中の $\lambda \sum_{i=1}^n \sum_{j=1}^n x_{ij}$ は L1 正則化項であり, λ はチューニングパラメータである. コーパス中には, 一つの段落内に複数の木構造が存在するような文章も含まれている. L1

正則化を用い、全体のエッジ数を制限することで、一つの段落内に複数の木構造が含まれる場合でも正しく推定することが可能となる。

制約である (4) から (9) 式は [Stab 17] と同様であり、循環がなく、一つのノードから引かれるエッジは一本以下である木構造を保証している。ここで、式 (7) から (9) で利用している a_{ij} は次式で値が決定され、循環構造の判定に用いられる。

$$a_{ij} = \begin{cases} 1 & (\text{要素}i\text{から要素}j\text{に可到達である}) \\ 0 & (\text{要素}i\text{から要素}j\text{に可到達でない}) \end{cases} \quad (10)$$

3. 評価実験

本章では、提案手法の有効性を評価する実験について述べる。評価実験では、小論文形式の文章に対して、論理要素の箇所と因果関係の有無がタグ付けされている "Argument Annotated Essays"^{*1} に含まれる 400 文書を用い、10 交差検証を行って各サブタスクと最終的な構造推定の精度を求めた。精度の指標には、各ラベルに対する F 値とそのマクロ平均を用い、[Stab 17] (以降では Stab2017 と表記する) と比較した。

3.1 要素分類の評価実験

本節では、要素分類に LSTM に基づく特徴量 (以降では、LSTM 特徴量と表記する) を用いる効果を評価する。

本実験では、LSTM 特徴量を求める際に、LSTM の入力を論理要素 (S) にする場合に加え、論理要素とその前文 1 文 (Pf1)、その前文 2 文 (Pf2)、論理要素以前に存在する段落内の文全て (PfAll)、論理要素とその次文 1 文 (Pb1)、その次文 2 文 (Pb2)、論理要素以降に存在する段落内の文全て (PbAll) の 6 パターンを用いた。ここで、次文を入力に含める際は、より文脈の情報を考慮できるように単語列の順序を逆転した上で入力とした。

また、それぞれの入力に対して、LSTM 特徴量を Average Pooling である H^{ave} とする手法と、最後の隠れ層である H^{last} とする手法を比較している。さらに、全ての手法に対して、隠れ層の次元数を 50・100・200 次元に設定し、比較した。ここでは、特徴量の差のみを評価するために、Stab2017 と同様に、Support Vector Machine (SVM) を用いて分類した結果を報告する。

Stab2017 と提案手法の要素分類の結果を表 1 に示す。F-Claim は「主張」の分類についての F 値、F-Premise は「前提」の分類についての F 値である。

表 1 より、論理要素のみを LSTM の入力とするだけでは、精度がほぼ変わらないことがわかる。一方で、論理要素とその前文 1 文を LSTM の入力とすることで、「主張」、「前提」のどちらの分類に対しても Stab2017 の精度を上回っていることから、LSTM を用いることで、文脈の情報を適切に活用できていることが示された。さらに、隠れ層を比較的高次元に設定した LSTM の最後の層の値を用いた際に、特に高い精度となることが明らかとなった。具体的には、論理要素と前文 1 文を入力とし、次元数 200 に設定した LSTM の最後の隠れ層を特徴量として用いた場合が最も高精度であった。これらの結果より、LSTM 特徴量を用いることで、要素分類の精度が大きく向上することが示された。

ただし、論理要素とその前後 2 文以上を入力とすると、精度は大きく低下する傾向にあることがわかった。これは、3 文

以上の入力では入力列が極端に長くなるため、LSTM を用いても勾配消失問題が生じ、学習が難しくなることが原因と考えられる。

同様に、今回の結果からは、次文の文脈を考慮することによる分類精度の改善はみられなかった。この原因として、本実験で使ったコーパスでは、論理要素と次文 1 文の単語数の平均が、論理要素と前文 1 文の単語数の平均よりも 10 単語以上多く、前後 2 文以上を入力とした時と同様に、入力列が長くなり、適切に学習できなかった可能性が考えられる。

3.2 関係分類の評価実験

次に、関係分類の特徴量として、要素分類で推定した各要素が「主張」である確率推定値 (以降では、主張特徴量と表記する) を用いる効果を評価する。

本実験では、主張特徴量を推定するのに既存の特徴量のみを用いた場合と、既存の特徴量と LSTM 特徴量を組み合わせ用いた場合の関係分類の精度を求め、関係分類を独立に行う Stab2017 の精度と比較した。また、要素分類の評価実験を踏まえ、LSTM 特徴量には、次元を 200 に設定し、論理要素と前文一文を入力とした LSTM の最後の隠れ層の値を用いている。ここでも、特徴量の差のみを評価するために、Stab2017 と同様に、SVM を用いて分類した結果を報告する。

これらの結果を表 2 に示す。F-Relation は「因果関係有」分類の F 値、F-noRelation は「因果関係無」分類の F 値、F-All は F-Relation と F-noRelation のマクロ平均である。

表 2 より、Stab2017 と LSTM 特徴量を用いない提案手法を比較すると、提案手法の「因果関係有」の分類精度が Stab2017 を大きく上回ることがわかる。同様に、「因果関係無」の分類精度も大幅に上回っており、関係分類の特徴量に主張特徴量を加えることで、関係分類の精度が大きく向上することが示された。

さらに、LSTM 特徴量を用いた提案手法が、比較手法の中で分類精度が最も高いことから、要素分類の精度向上が適切に反映されていることが示された。

3.3 構造推定精度の評価実験

本節では、提案手法の最終的な構造推定精度を評価する。ここでは、Stab2017 と提案手法のそれぞれについて、要素分類に LSTM 特徴量を用いる場合と用いない場合の構造推定精度を求める。本実験でも、要素分類の評価実験を踏まえて、LSTM 特徴量には、次元を 200 に設定し、要素と前文一文を入力とした LSTM の最後の隠れ層の値を使用する。また、式 (3) の L 1 正則化項に含まれるチューニングパラメータ λ の値は交差検証によって、刻み幅 0.01 で検証し、決定した。

これらの結果を表 3 に示す。F-Relation は「因果関係有」分類の F 値、F-noRelation は「因果関係無」分類の F 値、F-All は F-Relation と F-noRelation のマクロ平均である。

表 3 から、LSTM 特徴量を用いた手法を比較した場合、提案手法の分類精度が Stab2017 を上回ることがわかり、LSTM 特徴量を用いる場合、提案手法の構造推定手法が有効であることが示された。

一方、LSTM 特徴量を用いない場合、提案手法よりも Stab2017 の分類精度の方が高い。この原因として、Stab2017 の構造推定は要素分類の影響を受けにくい定式化であり、要素分類の精度が低い場合でも、構造推定の精度が大きく変化しないことが考えられる。

評価実験の結果から、Stab2017 と比較して、LSTM 特徴量を用いる提案手法では、論理構造推定の精度が大幅に上回るこ

*1 <https://www.ukp.tu-darmstadt.de/data/argumentation-mining/argument-annotated-essays/>

表 1: LSTM を用いた比較実験結果

要素分類手法	H	入力	次元	F-Claim	F-Premise
Stab2017	-	-	-	0.627	0.832
提案手法	H^{ave}	S	50	0.623	0.836
			100	0.623	0.838
			200	0.622	0.833
		Pfl	50	0.663	0.853
			100	0.671	0.858
			200	0.679	0.860
		Pfl	50	0.610	0.823
			100	0.611	0.824
			200	0.605	0.825
		PflAll	50	0.617	0.824
			100	0.616	0.826
			200	0.613	0.817
		Pb1	50	0.613	0.835
			100	0.612	0.836
			200	0.599	0.831
		Pb2	50	0.609	0.830
			100	0.606	0.829
			200	0.606	0.827
		PbAll	50	0.623	0.833
			100	0.615	0.825
			200	0.608	0.828
提案手法	H^{last}	S	50	0.622	0.838
			100	0.631	0.849
			200	0.648	0.855
		Pfl	50	0.608	0.832
			100	0.642	0.849
			200	0.698	0.875
		Pfl	50	0.618	0.823
			100	0.601	0.831
			200	0.601	0.834
		PflAll	50	0.614	0.831
			100	0.617	0.836
			200	0.606	0.833
		Pb1	50	0.615	0.834
			100	0.612	0.836
			200	0.598	0.834
		Pb2	50	0.613	0.835
			100	0.601	0.832
			200	0.596	0.833
		PbAll	50	0.613	0.835
			100	0.612	0.836
			200	0.599	0.831

表 2: 関係分類の評価実験結果

関係分類手法	LSTM	F-Relation	F-noRelation
Stab2017	-	0.428	0.731
提案手法	無	0.460	0.794
	有	0.507	0.828

表 3: 提案手法の評価実験結果

構造推定手法	LSTM	F-Relation	F-noRelation	F-All
Stab2017	有	0.506	0.890	0.698
	無	0.466	0.881	0.674
提案手法	有	0.527	0.894	0.718
	無	0.451	0.879	0.665

とがわかった。従って、提案手法では、Stab2017 よりも論理構造を高精度に推定できることが明らかとなった。

4. むすび

本論文では、LSTM を用いて文脈情報を活用し、サブタスク間に依存関係を導入した新たな論証マイニング手法を開発した。具体的には、以下の手順で論理構造を推定する。

- 1) 論理に関わる文や節である論理要素が「主張」であるか分類する要素分類において、入力を論理要素とその前文 1 文にした LSTM の最後の隠れ層の値を既存の特徴量と組み合わせて用いる。本研究では、各要素が「主張」である確率推定値がこれらの特徴量を所与としたシグモイド関数によって表現されると仮定し、推定された値を関係分類の特徴量として採用する。
- 2) 論理要素間の因果関係の有無を分類する関係分類では、要素分類で推定した各要素が「主張」である確率推定値を既存の特徴量と組み合わせて用いる。本研究では、因果関係の存在する確率推定値がこれらの特徴量を所与としたシグモイド関数によって表現されると仮定し、推定された値を、線形計画問題の目的関数の重みとして採用する。
- 3) 関係分類で推定された、各論理要素間に因果関係の存在する確率推定値を目的関数の重みとする線形計画問題を、木構造

の制約のもと、線形計画法を用いて解くことで、最終的な論理構造の推定を行う。

コーパスを用いた評価実験により、提案手法では、要素分類と関係分類の精度が既存手法を大きく上回っていることがわかった。また、提案手法の構造推定精度が既存研究の推定精度を大きく上回ることが明らかとなった。

ただし、それでもまだ因果関係ありについての分類精度は 53%程度と十分に高いとはいえず、応用のためにはさらなる精度の向上が必要である。そのために、さらなるディープラーニング手法の活用や特徴量選択により精度を向上させていく。

謝辞 本研究は JSPS 科研費 JP17H04726, JP17K20024 の助成を受けたものです。

参考文献

- [Ghosh 16] Ghosh, D., Khanam, A., and Muresan, S.: Course-grained Argumentation Features for Scoring Persuasive Essays, in Proceeding of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 549–554, Berlin, Germany (2016), Association for Computational Linguistics
- [Kim 14] Kim, Y.: Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014)
- [Lippi 16] Lippi, M. and Torroni, P.: Argumentation Mining: State of the Art and Emerging Trends, ACM Trans. Internet Technol., Vol. 16, No. 2, pp. 10:1–10:25 (2016)
- [Peldszus 14] Peldszus, A.: Towards segment-based recognition of argumentation structure in short texts, in Proceedings of the First Workshop on Argumentation Mining, pp. 88–97, Baltimore, Maryland (2014), Association for Computational Linguistics
- [Stab 17] Stab, C. and Gurevych, I.: Parsing Argumentation Structures in Persuasive Essays, Computational Linguistics, Vol. 43, No. 3, pp. 619–659 (2017)