

TextRank と依存関係情報の組み合わせによる Argument Component の分類

出口 衛

山口 和紀

東京大学

{deguchi-mamoru, yamaguch}@graco.c.u-tokyo.ac.jp

1 はじめに

近年, エッセイや討論などの非構造的な文章から, 自動的に論の構造を抽出することを目的とした Argumentation Mining が注目を集めている.

Argumentation Mining は 4 つのサブプロセスからなる. まず (1) 与えられた文章から論に関係する文や句 (Argument Component : AC) を抽出し, (2) 主張や理由のような AC の論における役割を付与する. その後, (3) 文書中の AC の組についてそれらの間に依存関係があるかどうかの分類を行い, 最後に (4) 得られた依存関係に対して, 対立や補強といった属性ラベルを付与する.

本研究ではこれらの 4 つのプロセスの中で (2)AC の分類, 特に与えられた文章からその文章の主要な主張 (MajorClaim : MC) とその主張を補強するための詳細な主張 (Claim : CL) を取得することを目的として研究を行った. 文章から MC, CL を抽出することは, その文章における要点を抽出する要約的な意味で有用であるとともに, 論構造全体の抽出においても重要なサブプロセスである. AC の分類に着目した先行研究としては, AC から抽出した素性により SVM を用いて分類を行った研究 [1] や ニューラルネットワークを用いて, MC 及び CL の抽出を行った研究 [2] がある.

また, AC の分類を単独で行うのではなく, AC 間の依存関係情報を用いることで精度を向上させる研究が行われている. [3] では AC の役割の分類と AC 間のリンクの有無の分類をそれぞれ独立に行い, 結果を整数計画法で最適化することにより精度を向上させた. また, [4] ではニューラルネットワークにおける PointerNet を用いて AC の分類と AC の組の関係抽出を同時に学習させることで分類精度を向上させている. これらの研究では, AC の分類において AC 同士の依存関係の情報を利用することで, AC の分類と依存関係抽出の両プロセスにおいて精度が向上することが示された. 一方で上

記の研究における手法では, 素性として AC の登場位置や手掛かり表現等の文書の構造に依存したものの利用していたり依存構造を段落内に制限しているなどの制約があるため, 文書のドメインに大きく依存している. また文書の局所的な情報しか用いることができないという問題点も存在する.

このような問題点に対して, [5] では, TextRank[8] を用いて文間の類似度をもとに順位付けを行って MC, CL を含む文の取得を行った. この研究は, 文章全体における文の関係性を文章の構造に非依存で利用できるという点において有用であるものの, TextRank におけるノード間の関係尺度として使用している AC の TFIDF コサイン類似度が, MC や CL の取得において適切な関係尺度であるかどうかについては言及していない.

本研究では TextRank を用いて文章全体の関係を MC や CL の取得に活用する手法は踏襲し, ノード間の関係尺度について再考する. 本研究では先行研究 [3][4] における AC 間の依存関係の情報が AC の分類に有効であるという知見に着目し, Argumentation Mining のサブプロセスである (3)AC 間の依存関係の抽出をニューラルネットワークを用いた分類器によって行い, 得られた出力値を TextRank のノード間の関係尺度として用いる手法を提案する. これにより, 「文書の構造」に依存することなく AC 間の依存関係利用することで MC 及び CL の抽出精度を向上させることに成功した.

2 背景

2.1 TextRank を用いた MajorClaim 及び Claim の取得

本章では本研究で用いた主要な技術である TextRank およびそれを用いた MC, CL の抽出手法について説明する.

TextRank は PageRank のアルゴリズムを自然言語処理分野に適用したランキングアルゴリズムであり、キーワード抽出や抽出型文書要約タスク等に利用されている。TextRank では、文や単語といったテキストの断片をグラフのノード、ノード間の関係尺度をノード間のエッジの重みとして重み付き有向グラフを構築する。この有向グラフからノードのスコアを再帰的に計算してノードの順位付けを行う。

ここで、文書中の文 S を TextRank のノードとして考える。文書 $D = \{S_1, \dots, S_n\}$ から得られる重み付き有向グラフを $G = (S, E)$ とする。ここで $E \subseteq S \times S$ であり、文 S_i から文 $S_j (1 \leq i, j \leq n)$ への有向リンクを E_{ij} で表す。有向リンク E_{ij} の重みを文 S_i と文 S_j 関係尺度 $w(S_i, S_j)$ としたとき、文 S_i のスコア $WS(S_i)$ を以下の式を用いて算出する。

$$W(S_i, S_j) = \frac{w(S_i, S_j)}{\sum_{S_k \in Out(S_i)} w(S_i, S_k)} \quad (1)$$

$$WS(S_i) = (1-d) + d * \sum_{S_j \in In(S_i)} W(S_j, S_i) * WS(S_j) \quad (2)$$

$In(S_i)$ は文 S_i への有向リンクを持つ文の集合であり、 $Out(S_j)$ は文 S_j からの有向リンクを持つ文の集合である。 d は 0 から 1 の間をとるパラメータである。先行研究 [5] では $w(S_i, S_j)$ として文 S_i, S_j の TFIDF コサイン類似度を用いている。

2.2 AC 間の依存関係抽出

本研究では TextRank におけるノード間の関係尺度として、AC 間の依存関係情報を用いるため、AC 間の依存関係の分類に関する先行研究について説明する。AC 間の依存関係に関する研究では、Argumentation Mining のサブプロセスである (3)AC 間の依存関係の抽出と (4) 抽出された関係の属性の分類、またそれらを同時に行うものがある。[3] では AC から人手で抽出した特徴を用いて、SVM により AC の組に依存関係があるかどうかの二値分類を行った。[6] では AC から得られる素性だけでなく、AC の周囲にある文脈情報を特徴量として用いて AC 間の依存関係の属性の分類を行った。また、[7] ではニューラルネットワークを用いた分類器によって AC 間の依存関係の属性を分類した。

本研究では、(3)AC 間の依存関係の抽出において、AC 間に依存関係があるかどうかを確率的な分類器を用いて分類したあと、得られた確率値を TextRank のノード間の関係尺度として用いる。本研究ではニューラルネットワークを用いた分類器を使用し、分類を行った。

3 提案手法

本研究ではニューラルネットワークを用いた分類器を用いて、文書中の AC の組に依存関係があるかどうかを確率的に分類し、得られた確率値を TextRank におけるノード間の関係尺度として用いることで AC の順位付けを行う。これにより MC および CL の抽出において、AC の出現位置やパラグラフ構造等の文書構造に依存することなく、文書全体の大域的な情報を考慮することが可能となる。

3.1 ニューラルネットワークを用いた AC 間の依存関係抽出

本研究では予備実験としてニューラルネットワークの構成をいくつか比較し、TextRank での順位付けの精度が最も高いもので最終的な評価実験を行った。本研究で用いたニューラルネットワークの構成を図 1 に示す。

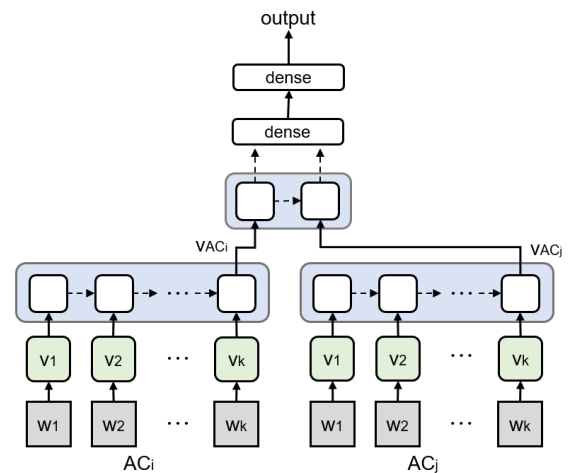


図 1: 実験で用いたニューラルネットワークの構成

文書中の AC の組 (AC_i, AC_j) を入力として、 AC_i から AC_j に対する依存関係があるかどうかの分類を確率的な分類器を用いて行う。 AC_i から AC_j に依存関係がある場合 *link*、そうでない場合 *nolink* とする。入力は AC の組 $AC_i = (w_1, w_2, \dots, w_k)$, $AC_j = (w'_1, w'_2, \dots, w'_k)$ とする。 w_l は AC に含まれる単語、 k は文書集合における AC の最大長を表す。入力された AC は Embedding 層で単語ベクトルの系列 $V_i = (v_1, v_2, \dots, v_k)$, $V_j = (v'_1, v'_2, \dots, v'_k)$ に変換され、LSTM に入力されて AC を表す文ベクトル V_{AC_i}, V_{AC_j} に変換される。 v_l は単語 w_l の単語ベクトルである。各文ベクトルは

V_{AC_i}, V_{AC_j} の順で次段の LSTM に入力され, LSTM での各 TimeStep の隠れ層の出力を結合し全結合層に入力する. 最後に softmax 関数により確率的に分類を行う. 単語ベクトルは Word2Vec の学習済みベクトルである GoogleNews-Vector¹を用いた.

3.2 TextRank を用いた Claim の抽出

3.1 の分類器により文書内の要素 AC_i と AC_j に有向リンクが存在する確率 $P(link|S_i, S_j)$ が得られる. TextRank では式 (1) において $w(S_i, S_j) = P(link|S_i, S_j)$ として計算を行う. また各要素からそれ自身への重み $w(S_i, S_i)$ は 1 とした. TextRank におけるハイパーパラメータ d はベースラインと提案手法ともに 0.85 とした.

4 実験

本実験では, ニューラルネットを用いた依存関係抽出の結果を TextRank の関係尺度として用いて, 文書から MC 及び CL を抽出することで性能が向上するかを検証する.

4.1 コーパス

実験では, Gurevych ら [3] が構築したコーパス²を用いた. Gurevych らは学生が書いたエッセイに対して, AC の役割として {MajorClaim(MC), Claim(CL), Premise(PR)} の 3 種類, 依存関係 (link) については CL から PR に対して {Support, Attack} の 2 種類, MC から CL に対して {For, Against} の 2 種類のアノテーションを行った. 本研究における依存関係抽出では, 依存関係に関する 4 種類のアノテーションを全て正例 (link), 依存関係のない AC の組を全て負例 (nolink) として実験データを作成した. 実験データの詳細を表 1 に示す.

4.2 学習データにおける負例の制限

本研究では予備実験として表 1 における学習データを用いて学習を行い順位付け結果についての分析を行った. その結果, 分類器において正例とは逆向き, つ

表 1: 実験に用いたコーパス

	essay	AC の属性			依存関係	
		MC	CL	PR	link	nolink
全データ数	402	751	1506	3832	6673	91798
学習データ	323	612	1235	3199	5569	78559
テストデータ	80	139	271	633	1104	13239

まり MC から CL または PR, 及び CL から PR に対してのリンクが張られる確率が高くなっている場合に, TextRank を用いた順位付けにおいて MC や CL の順位が低くなっていることが分かった.

この問題に対して, 本研究では表 1 の負例 (nolink) において学習に用いるものを MC から CL または PR, 及び CL から PR に対しての事例のみに制限した. これにより, 負例 (nolink) 全てを使用するよりも正例と逆向きのリンクのみを強く学習に反映させることができる. この操作によって, 学習に用いる負例の数は 26226 となった.

4.3 比較手法

比較手法としては, TextRank の要素間尺度として先行研究 [3] における各要素に含まれる単語を TFIDF で重みづけしたコサイン類似度を用いる手法 (TFIDF), 及び AC に含まれる単語埋め込み表現の平均ベクトルのコサイン類似度を用いる手法 (W2V) とした.

4.4 実験結果・考察

TextRank を用いた MC と CL の抽出結果を表 2 に示す. 提案手法において, Relation は学習データをそのまま用いたもの, Relation-Neg は学習データの負例に制限を加えたものである. 本研究におけるニューラルネットの実装には Keras³を用いた. 各層におけるユニット数は各 LSTM の隠れ層及び全結合層共に 64 とし, 出力は softmax 関数を用いた. また使用した学習済み単語ベクトルは 300 次元のベクトルを用いた. 提案手法では実験データの 20% を validation data とし, validation loss に対する early stop を採用した. 評価尺度である MC@n, CL@n はそれぞれ上位 n 番目までの AC のラベルが MC および MC または CL である場合に正解とし, 実験を 4 回行い得られた結果の平均をとってスコアを算出した.

¹<https://code.google.com/archive/p/word2vec/>

²<https://www.informatik.tu-darmstadt.de/>
(Argument Annotated Essays (version 2))

³<https://keras.io/>

表 2: TextRank を用いた MC と CL の抽出結果

評価尺度	total	TFIDF		W2V		Relation		Relation-Neg	
		correct essay	accuracy	correct essay	accuracy	correct essay	accuracy	correct essay	accuracy
MC@1	80	20	0.25	12	0.15	31.50	0.394	40.25	0.503
MC@2	80	32	0.40	23	0.2875	46.25	0.578	50.75	0.634
MC@3	80	38	0.475	33	0.4125	54.25	0.678	60.25	0.753
CL@1	80	49	0.6125	41	0.5125	57.25	0.715	64.50	0.810
CL@2	80	66	0.825	57	0.7125	74.25	0.928	76.75	0.959
CL@3	80	72	0.9	68	0.85	77.75	0.969	79.75	0.996

表 3: 各手法における AC の順位平均値

Experiment	TFIDF	W2V	Relation	Relation-Neg
AverageRank MC	6.158	6.475	4.558	3.993
AverageRank CL	6.528	6.760	5.990	5.861
AverageRank PR	8.006	7.837	8.588	8.767

提案手法では TextRank の文間尺度に対して分類器によって得られた要素間尺度を用いることで、各項目について文書間類似度を用いた手法と比較して大幅にスコアが向上した。また負例の選択を行うことで、負例の選択を行わない場合よりもスコアが向上することが分かった。これは負例を制限することで、正解と逆向きのリンクの確率値が高くなる誤りが少なくなったためであると考えられる。

各手法における MC, CL, PR の平均順位を表 3 に示す。提案手法では MC, CL の平均順位は上昇し、PR の平均順位が低下している。このことから提案手法では比較手法よりも MC, CL が平均的に上位に来るような AC 間の関係尺度を得ることができていると考えられる。

5 おわりに

本研究では文書に含まれる AC の依存関係をニューラルネットを用いて抽出し、得られた依存関係情報を TextRank における関係尺度として用いることで、文書全体の情報を考慮した文の順位付けをすることにより MC, CL を抽出した。またベースラインである TFIDF, W2V の結果よりも大幅に精度が向上し、この手法の有効性を示した。今後は依存関係抽出器の精度向上や、AC のラベルの情報を依存関係抽出に用いる手法、また AC のラベルの分類と依存関係抽出とを同時に学習する手法を検討する。

参考文献

- [1] Christian Stab and Iryna Gurevych. Identifying Argumentative Discourse Structures in Persuasive Essays. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 4656, Doha, Qatar, October 25-29, 2014.
- [2] Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. What is the Essence of a Claim? Cross-Domain Claim Identification. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2055-2066, Copenhagen, Denmark, September 7-11, 2017.
- [3] Christian Stab and Iryna Gurevych. Parsing Argumentation Structures in Persuasive Essays. arXiv:1604.07370v2 [cs.CL], Jul 22, 2016.
- [4] Peter Potash, Alexey Romanov, and Anna Rumshisky. Here's My Point: Joint Pointer Architecture for Argument Mining. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1364-1373, Copenhagen, Denmark, September 7-11, 2017.
- [5] Georgio Petasis and Vangelis Karkaletsis. Identifying Argument Components through TextRank. Proceedings of the 3rd Workshop on Argument Mining, pages 94102, Berlin, Germany, August 7-12, 2016.
- [6] Huy V. Nguyen and Diane J. Litman. Context-aware Argumentative Relation Mining. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1127-1137, Berlin, Germany, August 7-12, 2016.
- [7] Oana Cocarascu and Francesca Toni. Identifying attack and support argumentative relations using deep learning. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1374-1379, Copenhagen, Denmark, September 7-11, 2017.
- [8] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Texts. Proceedings of EMNLP 2004, pages 404-411, Barcelona, Spain. Association for Computational Linguistics. 2004.