Wikipediaの内部リンクを用いた弱教師あり共参照解析

福田 展和 * 佐藤 翔悦 * 吉永 直樹 † 喜連川 優 †,⋆

* 東京大学大学院 情報理工学系研究科 † 東京大学 生産技術研究所 ⋆ 国立情報学研究所

{fukuda, shoetsu, ynaga, kitsure}@tkl.iis.u-tokyo.ac.jp

1 はじめに

機械翻訳や構文解析を始めとする多くの自然言語処理タスクにおいて，深層学習による大幅な性能向上が報告されているが，文をまたぐ文脈を入力とする共参照解析や照応解析などの文脈解析ではその性能向上は限定的である．文脈解析は入力空間や探索空間が広大であり，解析の対象となる実世界の事物 (entity) が多様であるため，汎化性能の高いモデルを得るためには大規模な訓練データが必要となる．しかし，注釈コストの問題から大規模な訓練データの作成は困難である．

そこで，本稿では Wikipedia の内部リンクを用いて共参照解析の擬似的な訓練データを自動生成し，得られた訓練データを用いて共参照解析のモデルを事前訓練する手法を提案する．Wikipedia のリンクは特定の記事 (entity) を参照しており，共参照解析モデルの学習に転用可能であるが，実際に記事中で entity を指す言及 (mention) に対してリンクが網羅的でないという問題がある．そこで本研究では，任意の記事中の独立したリンクを組み合わせて訓練データを生成することにより，この問題を解決する．結果として，大規模な Wikipedia データを効果的に事前学習に利用することができる．

実験では共参照解析データセットである OntoNotes と WikiCoref を用いて提案手法の評価を行った．その結果，少ない訓練データからモデルを学習した場合や，異なるドメインに対して学習したモデルを適用する場合に，事前学習により CoNLL スコアが向上した．

2 関連研究

深層学習に基づく共参照解析モデルとして，mention の検出と mention のクラスタリングを同時に行う end-to-end モデル [1] が提案されている．この end-to-end モデルは人手で設計された言語的特徴量が用いられるパイプラインモデル [2] に比べて，基礎解析の誤差伝播が無く，最小限の特徴量設計でより高い精度を達成できる．本研究でも共参照解析モデルのベースラインとして，この end-to-end モデル [1] を用いる．

しかし，これらの共参照解析モデルは語彙的特徴量に依存するため異なるドメインにおいて精度が悪化することが指摘されており [3]，対策として，数多くの言語的特徴量から最適な特徴量の組み合わせを追加して用いる手法 [4] が提案されている．本研究は共参照解析モデルの自体の改善ではなく，Wikipedia から擬似的な訓練データを生成する手法を提案するものであり，これらの手法と相補的に組み合わせて利用できる．

本研究と同様に，深層学習モデルのパラメタを事前訓練する手法として，生コーパスから双方向の言語モデルを通して文脈を踏まえた単語表現を事前訓練する ELMo [5] が提案されており，共参照解析モデルの入力に用いることで精度が大きな性能向上が得られることが報告されている [5, 6]．一方，本稿の提案する事前訓練は，共参照タスクに特化したものであり，単語表現だけでなく共参照関係を捉えるモデル部分も訓練するため，ELMo と組み合わせて利用可能である．

3 Wikipediaの内部リンクに基づく弱教師あり共参照解析

本節では本研究で用いる共参照解析モデルと提案手法について述べる．

3.1 共参照解析モデル

本研究では共参照解析モデルとして，span に基づく end-to-end モデル [1] を用いる．このモデルでは入力文書中の単語系列 (span) に対して，span の mention らしさを表現する mention スコア $s_{\mathrm{m}}(i)$ を計算する．この mention スコアの高い span のみを mention に限定することで計算量を削減することができる．次に，

各 mention を照応詞，それ以前の mention を先行詞としたときの共参照スコア $s(i,j)$ を計算する．その後，各照応詞について共参照スコアの最も高い先行詞を 1 つずつ選択する操作を繰り返すことで，mention のクラスタリングを行いモデルの出力とする．

共参照スコア $s(i,j)$ は以下に述べる手順で計算する．まず入力文書を Long short-term memory (LSTM) [7] に通して span のベクトル表現 g_i を計算する．次にこの span 表現 g_i から mention スコア $s_{\mathrm{m}}(i)$ を，また，span 表現のペアから先行詞スコア $s_{\mathrm{a}}(i,j)$ を，それぞれ以下のように計算する．

$$s_{\mathrm{m}}(i) = w_{\mathrm{m}}^{\mathrm{T}}\mathrm{FFNN}_{\mathrm{m}}(g_i)$$
$$s_{\mathrm{a}}(i,j) = w_{\mathrm{a}}^{\mathrm{T}}\mathrm{FFNN}_{\mathrm{a}}(g_i, g_j, g_i \circ g_j, \phi(i,j))$$

ここで $\circ$ は要素積を表し，FFNN は順伝播型ニューラルネットワークを表している．$\phi(i,j)$ は span 間の特徴量を表し，本研究では span 間の距離を利用した．共参照スコアは mention スコアと先行詞スコアの和として計算し，mention 検出と共参照関係の推定を同時に訓練する．

$$s(i,j) = s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i,j)$$

この共参照スコアの最も高い mention を先行詞として選択する．なお，先行詞候補との共参照スコアが全て負であった場合には先行詞が無いものとする．損失関数は以下のように計算する．

$$\mathcal{L} = \sum_i \log \frac{\sum_{j' \in \mathrm{GOLD}(i)} \exp s(i,j')}{\sum_{j \in Y(i)} \exp s(i,j)}$$

ここで，$Y(i)$ は mention i の先行詞候補の集合，$\mathrm{GOLD}(i)$ は mention i の正しい先行詞を表す．

3.2 Wikipedia の内部リンクに基づく訓練データの生成

本節では Wikipedia の内部リンクから共参照解析の擬似的な訓練データを生成する手法を提案する．Wikipedia の記事には他の記事へのリンクを持つアンカーテキストが含まれている．ここで，記事は entity に，アンカーテキストは mention に対応すると考えることができる．従って，同じ記事へのリンクを持つアンカーテキストが文章 (記事) 中に複数存在すれば，共参照解析の訓練データとして用いることができる．しかし，1 つの記事において，他の同一記事へのリンクは基本的に最初に出現する mention にのみ限定される．[1] 特に，照応詞や指示詞などにはほとんどリンクが付与されないため，共参照解析の訓練データとしては不完全である．そのため，Wikipedia の記事とその記事に含まれる内部リンクをそのまま共参照解析の訓練データとして利用することは難しい．

図 1: 擬似的な共参照データの生成に用いる Wikipedia の内部リンク．

本研究ではこの問題を解決するため，図 1 のように，Wikipedia の記事から内部リンクを含む文のみを抽出し，リンクを含む記事に限定せず同一記事へのリンクを含む文を組み合わせて訓練データを生成する．具体的には，正例として同一記事へのリンクのペアを，負例として異なる記事へのリンクのペアを用いる．

3.3 自動生成した訓練データに基づく弱教師あり共参照解析

前節で生成した訓練データを用いて，共参照解析モデルを事前訓練する．具体的には，アンカーテキスト箇所の span 表現 g_i と，span 表現間の先行詞スコア $s_{\mathrm{a}}(i,j)$ を計算し，共参照関係を推定する．アンカーテキスト箇所を mention として指定するために，共参照解析モデルの mention スコアを計算する部分は訓練しない．アンカーテキストの語彙的情報によってモデルが周囲の文脈を軽視することを防ぐために，アンカーテキストは未知語トークンとしてマスクし訓練する．

損失関数は共参照スコアに対する 2 値クロスエントロピー損失とした．ここで y はリンクのペアに対して与えられる 0,1 のラベルである．

$$\mathcal{L} = y \log \sigma(s_{\mathrm{a}}(i,j)) + (1-y)\log(1-\sigma(s_{\mathrm{a}}(i,j)))$$

このようにして Wikipedia から生成した訓練データでモデルを事前訓練 (初期化) した後，既存の共参照解析の訓練データを用いてモデルを再訓練する．

[1] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

訓練ドメイン	**bc**		**bn**		**mz**		**nw**		**pt**		**tc**		**wb**	
評価ドメイン	**bc**	$\overline{\textbf{bc}}$	**bn**	$\overline{\textbf{bn}}$	**mz**	$\overline{\textbf{mz}}$	**nw**	$\overline{\textbf{nw}}$	**pt**	$\overline{\textbf{pt}}$	**tc**	$\overline{\textbf{tc}}$	**wb**	$\overline{\textbf{wb}}$
ベースライン	52.03	46.40	61.29	49.07	62.94	43.18	56.19	53.18	76.21	37.45	58.89	34.47	55.05	45.11
+事前訓練	53.44	48.67	62.26	50.39	64.15	46.64	56.18	53.44	76.10	39.79	60.84	37.22	52.75	48.22
Δ	+1.41	+2.27	+0.97	+1.32	+1.21	+3.46	−0.01	+0.26	−0.11	+2.34	+1.95	+2.75	−2.30	+3.11

表 1: OntoNotes の各ジャンルで訓練した共参照モデルの実験結果．ジャンルについて **bc** は討論放送，**bn** はニュース放送，**mz** は雑誌，**nw** はニュース記事，**pt** は聖書，**tc** は電話会話，**wb** はブログを指す．評価ドメインの $\overline{\textbf{x}}$ は **x** 以外のジャンルの評価データを用いてモデルを評価した結果である．

4 実験

本節では，提案手法を既存の共参照解析データセットに適用し，手法の有効性を確認する．特に，共参照解析で想定される対象ドメインにおける訓練データが少ない状況に注目して手法の評価を行う．

4.1 設定

提案手法の評価用データセットとして，英語のコーパスである OntoNotes に共参照解析の注釈を付与した CoNLL 2012 shared task [8] を用いた．OntoNotes にはニュースや会話などの 7 つのジャンルがあり，2802 文書の訓練データ，343 文書の開発データ，348 文書の評価データから構成される．さらに OntoNotes と異なるドメインのデータとして，Wikipedia の記事に共参照解析の注釈を付与した WikiCoref [9] を用いた．WikiCoref は 30 記事とサイズが小さいため，本稿ではこれらを分割せず，全ての記事を訓練もしくは評価データとして用いた．事前訓練に用いた Wikipedia データと WikiCoref との間に記事の重複があると考えられるが，WikiCoref に含まれる照応関係には基本的にリンクが存在しないため，事前訓練のデータと WikiCoref の mention の重複は小さいと考えられる．評価尺度には共参照解析の指標として一般的に用いられる CoNLL スコア [10] (以降，精度) を用いた．

事前訓練に用いた Wikipedia のデータは，10 記事以上からリンクされた約 48 万記事を対象にした．記事当たりの被リンク数は平均 78，総リンク数は約 3820 万であった．事前訓練時は，これらの entity からランダムにサンプルして正例と負例を生成しモデルのミニバッチ学習に用いた．共参照解析の訓練において単語埋め込みの語彙は共参照解析の訓練に用いる訓練データの語彙のみとし，開発データ・評価データにのみ含まれる語彙は全て未知語として扱った．

訓練ドメイン	**OntoNotes**		**WikiCoref**
評価ドメイン	**OntoNotes**	**WikiCoref**	**OntoNotes**
ベースライン	64.66	43.38	36.51
+事前訓練	64.76	43.60	40.70
Δ	+0.10	+0.22	+4.19

表 2: OntoNotes および WikiCoref で訓練した共参照モデルの実験結果．

共参照解析モデルの実装には PyTorch (ver. 1.0)[2] を用いた．単語埋め込みにはウェブデータから事前訓練された 300 次元の GloVe[3] を固定して用い，文字埋め込みの次元は 8 とした．文字畳み込みのカーネルサイズは 3,4,5 とし，フィルタサイズはそれぞれ 50 とした．LSTM は双方向で隠れ層は 200 次元とし，FFNN の中間層の次元は 150 次元とした．最適化手法には Adam を用い，初期学習率は 0.001 とし，100 ステップ毎に 0.1%ずつ減少させた．

4.2 結果

表 1 に OntoNotes の 7 つのジャンル (以降，ドメイン) において，各ジャンルで訓練し訓練データと同じドメインの評価データで評価した結果 (ドメイン内評価) と，異なるドメインでの評価データで評価した結果 (ドメイン外評価) を示す．訓練データと異なるデータで評価した場合には，提案手法によって大きく精度が向上した．訓練データと同じドメインの評価データで評価した場合も，基本的には精度が向上しているが，一部ドメインにおいて精度の低下が見られた．

表 2 に OntoNotes で訓練して OntoNotes と WikiCoref で評価した結果，および WikiCoref で訓練して OntoNotes で評価した結果を示す．表 2 では，大規模な学習データ (OntoNotes) を用いた場合，評価ドメインの内外によらず精度の向上は僅かであった．一方で，

[2] https://pytorch.org/
[3] https://nlp.stanford.edu/projects/glove

訓練ドメイン	**WikiCoref**	
評価ドメイン	**OntoNotes**	
ベースライン	(36.51)	Δ
+ELMo	43.32	+6.89
+事前訓練	45.60	+2.28 (+9.09)

表 3: ELMo を追加し WikiCoref を用いて訓練した共参照解析モデルの実験結果.

小規模な学習データ (WikiCoref) で学習した場合は，事前訓練によって共参照解析の精度が大きく改善している．

最後に ELMo を追加して，表 2 と同様に WikiCoref で訓練して OntoNotes で評価した結果を表 3 示す．ELMo により精度が向上したベースラインモデルに対しても事前訓練は有効であり，提案する事前訓練と ELMo には相補的な効果があることが確認できた．

4.3 考察

提案手法により，訓練データが少ない場合，特にドメイン外評価における共参照解析の精度が提案手法によって向上した (表 1 のドメイン外評価，表 2 で WikiCoref で訓練して OntoNotes で評価した場合の評価結果)．表 1 の nw，pt，wb ドメインでのドメイン内評価においては提案手法による効果が見られなかった．特に，表 1 の wb のドメイン内評価において精度が悪化した理由としては，事前訓練に用いた Wikipedia データが，このドメインにおける共参照関係に効果的でなかったためと考えられる．

表 2 において OntoNotes を訓練データに用いて学習したモデルにおいて提案手法による効果があまり見られなかった．これは，OntoNotes が比較的大規模なデータセットであるためと考えられるが，さらに詳細な分析を行った．固有名詞が含まれる mention の割合は，OntoNotes の開発データでは 30%，WikiCoref では 55%であるが，提案手法が生成する訓練データの mention は主にリンクの存在する固有名詞であり，代名詞や名詞句を扱うことができていない．また，Wikipedia において定冠詞などを含まない短い名詞句にリンクが付与されることが多いが，共参照の注釈付けでは，定冠詞などを含む長い名詞句が mention となり，mention の範囲が異なる．さらに，ニュース記事や会話などで構成される OntoNotes と，Wikipedia の記事のドメインが異なることが考えられる．

5 おわりに

本稿では Wikipedia の内部リンクを利用して，共参照解析モデルを事前訓練する手法を提案した．提案手法は大規模な学習データがある場合には共参照解析の精度に寄与しなかったが，訓練データが少ない場合，特に異なるドメインに適用する場合には大きな性能向上が確認できた．

今後の課題として，アンカーテキストだけでなく，記事の本文から類義語を mention として抽出して用いることを検討している．また，アンカーテキストの存在する記事のカテゴリが同じであるなど，より適当な組のみを擬似的な共参照関係とすることも検討している．さらに，ベースラインとして語彙的情報のみを用いるモデルを用いたが，異なるドメインに対して頑健であるモデル [4] にも適用して比較検証したい．

謝辞

本研究の一部は，情報通信研究機構の委託研究の成果です．

参考文献

[1] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. End-to-end neural coreference resolution. In *EMNLP*, pp. 188–197, 2017.

[2] K. Clark and C. D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *ACL*, pp. 643–653, 2016.

[3] N. S. Moosavi and M. Strube. Lexical features in coreference resolution: To be used with caution. In *ACL*, pp. 14–19, 2017.

[4] N. S. Moosavi and M. Strube. Using linguistic features to improve the generalization capability of neural coreference resolvers. In *Proc. EMNLP*, pp. 193–203, 2018.

[5] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *NAACL*, pp. 2227–2237, 2018.

[6] K. Lee, L. He, and L. Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL*, pp. 687–692, 2018.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[8] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *CoNLL*, pp. 1–40, 2012.

[9] A. Ghaddar and P. Langlais. Wikicoref: An english coreference-annotated corpus of wikipedia articles. In *LREC*, 2016.

[10] S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube. Scoring coreference partitions of predicted mentions: A reference implementation. In *ACL*, pp. 30–35, 2014.