大規模格フレームによる解候補削減を用いたニューラルネットゼロ照応解析

山城 颯太　　西川 仁　　徳永 健伸
東京工業大学 情報理工学院
{yamashiro.s.aa@m, {hitoshi, take}@c}.titech.ac.jp

1 はじめに

本研究の貢献は大きく二つに分けられる．第一に大規模均衡コーパス上で日本語ゼロ照応解析を行い評価したことと，第二にこの大規模均衡コーパス上で文内・文間ゼロ照応解析を可能にするための解候補削減手法を提案したことの二点である．

従来のゼロ照応解析研究は，新聞記事のみからなる『NAIST テキストコーパス』（NTC） (Iida et al., 2007) で評価を行うものが多かった．従って，それらの評価ではテキストドメインの違いによる影響が考慮されていない．しかしゼロ照応解析結果の応用を考えた時，新聞のみならずブログ，QA，書籍，白書，雑誌などあらゆるドメインの文書に対して頑健なゼロ照応解析手法こそより有用性が高い．我々は『 現代日本語書き言葉均衡コーパス』(BCCWJ) (Maekawa et al., 2014) を評価実験に使用した．

ひとつのモデルで文内・文間のゼロ照応解析を同時に行う際，各格に対してそれぞれ独立に解析を行うより，他の格の情報を利用して複数格を同時に解析する方がより良い精度が得られると考えられる．しかし複数格を同時に解析する際には，先行詞の広大な探索範囲の問題に対処する必要がある．特に機械学習を適用する際，正解の候補となる名詞の組合せが大幅に増加することから，BCCWJ の場合では正例と負例の比率が約 1 対 20,000 と著しく不均衡となる．このような偏った訓練データは不必要に計算量を増幅させ，かつモデルの汎化を妨げる要因となる．我々は，学習に不要な負例を削減するために，解析対象述語に対応する格フレームを用いた効率的な解候補削減手法を提案する．この提案手法により，正解を候補に残しつつ，約 1,000 分の 1 にまで解候補を削減することに成功した．

格フレーム	ガ格	回数	ヲ格	回数	ニ格	回数
オープンしてる:動 1	店	129	–	–	近く	6
	カフェ	38	–	–	跡地	2
	レストラン	14	–	–	ところ	2
	...	...	–	–	...	...
オープンしてる:動 2	ブランド	12	ショップ	59	–	–
	専門家	8	サロン	18	–	–
	オーナー	4	ブティック	13	–	–
	...	...	...	...	–	–

表 1: 「オープンしてる」の格フレーム例

2 関連研究

2.1 日本語ゼロ照応解析

Matsubayashi and Inui (2017) はフィードフォワードニューラルネットワーク (FNN) とリカレントニューラルネットワーク（RNN）を組合せて用いることで，NTC に対して直接の係り受け関係と文内のゼロ照応解析を同時に行い，直接の係り受け関係と文内ゼロ照応解析の state-of-the-art を達成している． Sasano and Kurohashi (2011) は対数線形モデルを用いて，979 文からなる Web コーパスに対して文内と文間のゼロ照応解析を同時に行い，文内・文間ゼロ照応解析の state-of-the-art を達成している．

2.2 大規模格フレーム

格フレームとは述語とその述語が取りうる項を述語の格パターンごと，格ごとに整理した共起情報である．表 1 のように格パターンに基づいて格フレームを分けることで，述語と項間の語彙的選好の知識を照応解析に利用することができる (Sasano et al., 2008; Sasano and Kurohashi, 2011; Hangyo et al., 2013)．格フレームの構築に関しては Kawahara and Kurohashi (2006) が Web テキストから格フレームを自動構築する手法を提案している．これらの大規模 Web コーパスから取得，整理された格フレーム知識は京大格フレーム[1]として公開されている．

3 提案モデル

3.1 モデル

解析対象述語 p が含まれる文を S_0 とし， 入力文書 t に含まれる S_0 から h 文前までの文をそれぞれ $S_{-1}, S_{-2}, \cdots, S_{-h}$ とする．S_0 から S_{-h} までに含まれるすべての名詞の集合を $E_p = \{e_1, e_2, \cdots, e_n\}$ とする．これらに加えて『照応なし』または『外界照応』を意味する e_{none} を E_p に追加する．述語 p に対応する京大格フレーム中の格フレーム群を $CF_p = \{cf_1^p, cf_2^p, \cdots, cf_m^p\}$ とする．1 つの格フレーム cf_l^p に

[1] http://www.gsk.or.jp/catalog/gsk2008-b/ ただし，リンク先の京大格フレームは古い版であり，本項において使用したものは未公開の新しい版である．

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

は，それぞれの格 $c \in \{$ ガ格, ヲ格, ニ格 $\}$ に対応する 3 つの格スロットがあり，E_p 中に含まれるいずれかの名詞がそれぞれの格スロットに対応する格要素である．格スロットと格要素の対応付けを $a = \langle$ ガ格 $\leftarrow e_i$, ヲ格 $\leftarrow e_j$, ニ格 $\leftarrow e_k \rangle$ とする．述語項構造候補を (cf_l^p, a) とし，これを表現する素性ベクトルを $\boldsymbol{f}(cf_l^p, a, t)$ とする．このモデルの出力は以下の式 (1) で表せる．$\boldsymbol{w}$ は訓練データから学習されるパラメータである．このモデルは， Hangyo et al. (2013) のモデルをベースとしている．

$$cf_l^{p*}, a^* = \underset{cf_l^p, a}{\operatorname{argmax}} \boldsymbol{w} \cdot \boldsymbol{f}(cf_l^p, a, t) \quad (1)$$

3.2 素性

素性ベクトル $\boldsymbol{f}(cf_l^p, a, t)$ は以下 4 種類の素性の組合せからなる．

ベースモデル素性　ベースモデル素性 ϕ_{BMF} の各要素は実数かバイナリ値である．ベースモデル素性 ϕ_{BMF} は Sasano et al. (2008) の確率的格解析モデルから得られる表層の係り受けの確率と Hangyo et al. (2013) が提案する素性群からなる． Hangyo et al. (2013) の素性は格フレーム素性，述語素性，文脈素性の 3 種類からなる．例えば，ある格要素がその格フレームの格スロットに埋まるかどうかの確率は格フレーム素性の一つである．

格要素分散表現　格要素分散表現 ϕ_e は各格 c の格要素 e_c に対応する 3 つの分散表現から構成される．語の分散表現を生成するモデルとしては word2vec (Mikolov et al., 2013) を使用した[2]．

格フレーム内平均ベクトル（MVC）　表 1 に示すように京大格フレーム内では，述語 p に対するそれぞれの格フレーム cf_l^p は各格 c に対応する単語リストから構成される．格フレーム内平均ベクトル (MVC)$\overline{\phi}_{cf_l^p(c)}$ は，格フレーム cf_l^p 中の各格 c の分散表現ベクトル ϕ_w の重み付き平均として計算される．

さらに $\overline{\phi}_{cf_l^p(ガ)}$，$\overline{\phi}_{cf_l^p(ヲ)}$，$\overline{\phi}_{cf_l^p(ニ)}$ を結合して $\overline{\phi}_{cf_l^p}$ を生成する．$\overline{\phi}_{cf_l^p}$ を使って，a と cf_l^p の関連（選択選好）を測り，尤もらしい組合せを探索する．なお我々は，MVC を照応解析，解候補削減の両方で使用する．

文脈ベクトル　文脈ベクトル $c_{cf_l^p,a,t}$ はローカルアテンション機構付き RNN の出力である．この RNN は解析対象述語を含んだ文とその前方 h 文を受け取り，対象述語に対する文脈をモデリングする．

我々のアテンション機構モデルは他の素性ベクトルの連結に基づいてアライメント重みベクトルを推論する．

[2]日本語 wikipedia (2016-09-20) の本文全文から取得した約 100 万記事に対して，次元数を 500，window を 15 として学習させることで得られたモデルを使用した．

4 格フレーム中の分散表現を利用した解候補削減

ゼロ代名詞となる格要素の先行詞候補を網羅的に探索すれば，列挙される述語項構造候補 (cf_l^p, a) の集合は爆発的な規模となり，探索範囲は非実用的なものとなる．Sasano and Kurohashi (2011) の基準を参考に，ゼロ代名詞となる格要素の先行詞候補は述語が含まれる文より 3 文前までのみを範囲として解候補削減を行っている．つまり 3.1 の h を 3 とした．BCCWJ 中の格要素の分布においては，この制限によってゼロ代名詞の 89.16%をカバーできる．

n と m をそれぞれ E_p 中の名詞句数，対象述語の格フレーム数とすると，この制限を用いてもなお，候補の数は $O(n^3m)$ となり，BCCWJ 中の各動詞に対して約 20,000 個の述語項構造候補が出現する．

4.1 述語内平均ベクトル（MVP）

我々は，格フレーム候補と項候補の組合せについて 3.2 で提案した MVC と，**述語内平均ベクトル** (MVP) $\overline{\phi}_{p(c)}$ の二種類の平均ベクトルを使用した効率的な解候補削減手法を提案する．MVP は各格 c について述語 p に対応するすべて格フレームに渡って MVC $\overline{\phi}_{cf_l^p(c)}$ の重み平均を取ったベクトルである．重みは京大格フレーム中の各格フレームの頻度に基づく．我々の解候補削減手法は Ouchi et al. (2015) の山登り法を参考に，格フレーム候補と項候補の組合せ数を削減する．この解候補削減は計算効率のみを目的とするのではなく，訓練データ中の正例・負例のデータ数の非対称性の解消も目的とする．我々のケースでは，1 つの正例に対して 20,000 の負例が生じるため，これに対処している．前述したように，我々は訓練データ中のほとんどの負例は訓練に貢献しないと考え，解候補削減を行う．

4.2 アルゴリズム

我々の提案する解候補削減手法をアルゴリズム 1 に示す．ある述語 p には，文脈に対するその語義の曖昧性を反映した複数の格フレーム CF_p が存在する．それぞれの格フレーム cf_l^p に対応する格フレーム内平均ベクトル $\overline{\phi}_{cf_l^p(c)}$ はその格フレームの選択選好を反映しているため，これと項候補ベクトル ϕ_e の距離が近いほど，その項候補 e は対象格フレーム cf_l^p の格スロット c に埋まりやすいと言える．このアルゴリズムは与えられた述語に対して，二つのベクトル間の距離が最も近くなる格フレームと項候補の組合せを探索する．しかしながら，京大格フレームは自動的手法で構築されているので，本来別々の格フレームが一つの格フレームとしてまとめられてしまっている，あるいは同じ一つの格フレームが別々に分断されてしまっている可能性がある．この問題に対処するために，我々は提案する解候補削減手法に二種類の平均ベクトルを導

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

アルゴリズム 1 解候補削減アルゴリズム

Input:
a predicate p to be analyzed,
a set of case frames CF_p corresponding to p,
a set of cases C = { ガ格, ヲ格, ニ格 },
a set of nouns E_p appearing within the h preceding sentences.
Output:
optimal cf_l^{p*}, e_c^* for the analyzed p and each case $c \in C$.

1: **for** each case $c \in C$ **do**
2: $e_c^{(0)} \Leftarrow \underset{e \in E_p}{\text{argmax}} \cos(\bar{\phi}_{p(c)}, \phi_e)$ ▷ $\bar{\phi}_{p(c)}$ is the MVP
3: **end for**
4:
5: $cf^{(0)} \Leftarrow \underset{cf_l^p \in CF_p}{\text{argmax}} \sum_{c \in C}$ Pseudo-Score$(cf_l^p, e_c^{(0)})$
6: $t \Leftarrow 0$
7: **repeat**
8: **for** each case $c \in C$ **do**
9: $e_c^{(t+1)} \Leftarrow \underset{e_c \in E_p}{\text{argmax}}$ Pseudo-Score$(cf^{(t)}, e_c)$
10: **end for**
11:
12: $cf^{(t+1)} \Leftarrow \underset{cf_l^p \in CF_p}{\text{argmax}} \sum_{c \in C}$ Pseudo-Score$(cf_l^p, e_c^{(t+1)})$
13: $t \Leftarrow t + 1$
14: **until** $e_c^{(t)} = e_c^{(t+1)}$ and $cf^{(t)} = cf^{(t+1)}$
15: **return** $cf_l^{p*} \Leftarrow cf^{(t)}, e_c^* \Leftarrow e_c^{(t)}$ for each case $c \in C$
16:
17: **function** Pseudo-Score(cf_l^p, e)
18: $score \Leftarrow 0$
19: **for** each case $c \in C$ **do**
20: $score \Leftarrow score + P(p, cf_l^p, e, c)$
21: $score \Leftarrow score + \cos(\bar{\phi}_{cf_l^p(c)}, \phi_e)$
22: $score \Leftarrow score + 0.5 \times d_{p,e}$ ▷ $\bar{\phi}_{cf_l^p(c)}$ is the MVC
23: **end for** ▷ $d_{p,e}$ is the distance between p and e
24:
25: **return** *score*
26: **end function**

入した．MVC はある述語に対する格フレームの違いを区別し，MVP は格フレームの違いを考慮せず述語のみを考慮する．

まず初期値として各格 $c \in C$ に埋まりうる項 $e_c^{(0)}$ を仮に定める (行 1-3)．MVP $\bar{\phi}_{p(c)}$ と項候補の分散表現 ϕ_e とのコサイン距離を求め，これが最小となる，すなわち対象述語に埋まる項群に最も近い項を初期項とする．この段階では，MVP を使用することで特定の格フレームではなく述語のみを考慮している．格フレーム候補と初期項の組合せを入力とした Pseudo-Score(行 17-26) の返すスコアに基づいて，これらの初期項に対して最適な格フレーム $cf^{(0)}$ を格フレームの初期値とする (行 5)．Pseudo-Score については Sasano et al. (2008) を参考に，我々は以下の 3 つの要素を考慮した．(1) 京大格フレームに基づく (述語，格フレーム，深層格，項) の組合せの出現確率，(2) 格フレーム内平均ベクトル（MVC）と項候補の間のコサイン類似度，および (3) 述語と項候補の間の文数，である．このスコアの係数は経験的に定めた．以降，格フレーム $cf^{(t)}$ を固定して項 $e_c^{(t+1)}$ を探索するフェーズ (行 8-10) と項 $e_c^{(t+1)}$ を固定して格フレーム $cf^{(t+1)}$ を探索するフェーズ (行 12) を繰り返し，格フレームと項が更新されなくなればループを抜ける (行 6-15)．このアルゴリズムでは返り値として最もスコアの高い格フレームと項の組合せを返すが，実際にはループ中の毎回の探索過程で計算した項候補のうち 3 ベストまでを候補として保存する．最終的な出力は探索の過程で保存されたすべての格フレームと項の組合せである．提案した解候補削減手法により，約 70%の正解を候補に残しつつ，約 1,000 分の 1 まで解候補を削減することができた．

5 評価実験

5.1 ゼロ照応解析手法

S0 提案する解候補削減を行い，ベースモデル素性を使用して SVM モデルを実装した．ランキング学習には SVM^{rank}[3] (Joachims, 2006) を使用した．カーネルは線形である．このモデルは正例と負例から識別関数を学習し，この識別関数が最も高い解候補を一つ出力する．

S0_each 提案する解候補削減手法の効果を評価するためには，解候補削減を用いないモデルと比較することが自然である．しかしながら，前方 3 文までに先行詞候補の探索範囲を制限しても，述語一つあたりに対して 20,000 の述語項構造候補が出現するため，訓練時の計算複雑性は現実的ではない．これは複数の格を同時推定するために，格要素候補同士の組合せを考慮していることが原因である．そこで我々は，それぞれの格を独立に解析することで，解候補削減が必要ない単一格解析手法を用意した．この手法では，3 つの格に対してそれぞれ別の SVM モデルを用意し，これらを独立に学習させて，評価の際は各格に対応するモデルのそれぞれの出力を組合せて最終的な出力とした．この各格に対して独立の SVM を用いて学習を行ったモデルを S0_each とする．

S0′ 複数格の同時推定のために我々は単純な解候補削減手法を用意した．この手法では，解析対象述語に近い方から先行する n 個の名詞のみを格要素候補として選ぶ．この時の各格に対する格要素候補数は，提案手法と同程度の格要素の組み合わせ数となるよう調整した値であり，今回は $n = 5$ とした．この単純な解候補削減手法を適用した上で SVM を用いて学習を行ったモデルを S0′ とする．

F0 ベースモデル素性，格要素分散表現，格フレーム内平均ベクトル（MVC），文脈ベクトルを使用して FNN モデルを実装した．FNN の設計に際しては Matsubayashi and Inui (2017) を参考に，誤差関数にはソフトマックスクロスエントロピーを用い，各隠れ層には batch 正則化 と ReLU 活性化関数を使用した．RNN には GRU を使用した．

[3] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

	文内				文間				All			
格	ガ格	ヲ格	ニ格	All	ガ格	ヲ格	ニ格	All	ガ格	ヲ格	ニ格	All
モデル＼例数	3,137	1,458	873	5,468	2,359	495	243	3,097	5,496	1,953	1,116	8,565
S0_each	.570	.730	.757	.643	.085	.016	.144	.080	.397	.602	.660	.480
S0′	.490	.712	.725	.589	.032	.016	.140	.038	.331	.584	.632	.435
S0	**.575**	**.758**	**.777**	**.661**	.044	.016	.145	.048	.390	**.628**	**.679**	**.491**
F0	.562	.751	.775	.647	**.139**	**.038**	**.152**	**.126**	**.402**	.620	.678	.488

表 2: BCCWJ における結果 (F 値)

5.2 データセット

実験データとして,『現代日本語書き言葉均衡コーパス』(BCCWJ) (Maekawa et al., 2014) のコアデータ[4]を使用した．BCCWJ のコアデータ約 2,000 文書に対しては，人手による述語項構造と照応関係が付与されており，これは新聞，雑誌，書籍，白書，Yahoo!知恵袋，Yahoo!ブログの 6 ドメインにまたがっている．全体のドメイン分布を反映するようにデータを分割し，全体の約 4/5 を訓練用データ，約 1/20 を開発用データとし，残りを評価用データとして使用した．本研究で対象とした述語は動詞のみで，形容詞，事態性名詞は扱っていない．

6 結果と考察

複数格同時推定の効果　表 2 は BCCWJ におけるゼロ照応解析の実験結果である．S0_each と S0 を比較すると，多くの列において，S0 が S0_each より高い精度を示していることがわかる．ただし文間ガ格，全体ガ格の列においては，S0_each が S0 より高い精度を示している．これは単格の推定では，他の格における誤りから影響を受けないため，複数格同時推定の時より値が良くなっているのだと考えられる．一方で，ニ格，ヲ格については，S0_each は他の格の情報が使えないため比較的精度が低く，全体としての精度も，複数格同時推定を行っている S0 に劣っている．

解候補削減の効果　表 2 で，S0′ と S0 を比較すると，すべての列において，S0 が S0′ より精度が高い．我々はこの結果に対して有意水準 0.1%で McNemar 検定を行い，統計的有意差を確認した．このことから我々の提案した解候補削減手法がうまく機能しているといえる．

文脈ベクトルの効果　格要素分散表現，格フレーム情報，ローカルアテンション付き RNN モデルを使用し文脈情報を導入することで，F0 は全体の精度においては S0 に劣っているが，文間照応においては S0 より高い精度を示している．

7 おわりに

本論文では分散表現で平均化した格フレームによる解候補削減アルゴリズムによって大規模な多ドメインコーパスによる訓練を可能とする日本語文内・文間ゼロ照応モデルを提案した．また，ローカルアテンション機構付き RNN と FNN を組合せて使用し様々な素性を取り入れることで，文間ゼロ照応解析においてより高い精度が出ることを確認した．

[4] http://pj.ninjal.ac.jp/corpus_center/bccwj/

謝辞

(Hangyo et al., 2013) に関して詳細な情報をご教示くださった萩行正嗣氏，(Ouchi et al., 2017) の全体像についてご教示くださった大内啓樹氏に厚く御礼申し上げます．

参考文献

Hangyo, M., Kawahara, D., and Kurohashi, S. Japanese Zero Reference Resolution Considering Exophora and Author/Reader Mentions. in *EMNLP*, pp. 924-934, 2013.

Iida, R., Komachi, M., Inui, K., and Matsumoto, Y. Annotating a Japanese Text Corpus with Predicate-Argument and Coreference Relations. in *Proceedings of the Linguistic Annotation Workshop*, pp. 132-139, 2007.

Joachims, T. Training Linear SVMs in Linear Time. in *Proceedings of the 12th ACM SIGKDD*, pp. 217-226, 2006.

Kawahara, D. and Kurohashi, S. A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. in *ACL*, pp. 176-183, 2006.

Maekawa, K., Yamazaki, M., Ogiso, T., Maruyama, T., Ogura, H., Kashino, W., Koiso, H., Yamaguchi, M., Tanaka, M., and Den, Y. Balanced corpus of contemporary written Japanese. Language Resources and Evaluation, Vol. 48, No. 2 , pp. 345-371, 2014.

Matsubayashi, Y. and Inui, K. Revisiting the Design Issues of Local Models for Japanese Predicate-Argument Structure Analysis. in *IJCNLP*, pp. 128-133, 2017.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 2013.

Ouchi, H., Shindo, H., Duh, K., and Matsumoto, Y. Joint Case Argument Identification for Japanese Predicate Argument Structure Analysis. in *ACL-IJCNLP*, pp. 961-970, 2015.

Ouchi, H., Shindo, H., and Matsumoto, Y. Neural Modeling of Multi-Predicate Interactions for Japanese Predicate Argument Structure Analysis. in *ACL*, pp. 1591-1600, 2017.

Sasano, R. and Kurohashi, S. A Discriminative Approach to Japanese Zero Anaphora Resolution with Large-scale Lexicalized Case Frames. in *IJCNLP*, pp. 758-766, 2011.

Sasano, R., Kawahara, D., and Kurohashi, S. A Fully-Lexicalized Probabilistic Model for Japanese Zero Anaphora Resolution. in *COLING*, pp. 769-776, 2008.

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.