分散表現による格フレームの格要素の汎化を利用したゼロ照応解析

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.titech.ac.jp　{hitoshi,take}@c.titech.ac.jp

1 はじめに

本稿では大規模格フレームを利用したゼロ照応解析において，分散表現を使用した素性を取り入れることで，より広範な語彙への対応を可能とした格フレーム中の格要素の汎化手法を提案する．

日本語におけるゼロ照応解析とは，与えられた文章中の述語に対して，その省略された項 (ゼロ代名詞) を検出し，埋めるべき格要素を文章中に言及されている対象，もしくは文章中で言及されていない対象から同定するタスクであり，述語項構造解析の一部として近年盛んに研究されている (松林ら, 2015; Sasano, et al., 2011; 萩行ら, 2014)．

(1) 大岡山商店街でも (ϕガ) お洒落な建物を見かける ようになった．カフェテリアが特に多くて，今月も 新しく (ϕガ)(ϕニ)オープンしてる．

例 (1) では「見かける」のガ格と「オープンしてる」のガ格，ニ格が省略されている.「オープンしてる」のガ格に埋まるべき格要素は同文中に言及されている「カフェテリア」 であり (文内ゼロ照応)，ニ格に埋まるべき格要素は前文で言及されている「大岡山商店街」である (文間ゼロ照応)．一方,「見かける」 のガ格に埋まるべき格要素は文章外の著者である (外界ゼロ照応)．

ゼロ照応解析のための有用な言語資源として格フレームがある．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，格ごとに整理した知識である．例えば例 (1) のように「オープンしてる」のガ格には「店」などの施設を示す名詞が入りやすく，ニ格には「近く」などの場所を示す名詞が入りやすいと考えられる．しかし一方で，ガ格に「サイト」や「ページ」などの名詞が入った場合には，ニ格には格要素の入らない可能性が高い (照応なし)．これらを「オープンしてる」についての別の格パターンとして分けておくことで，述語や格要素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, et al., 2011; 萩行ら, 2014)．

格フレームの構築に関しては河原らが 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 (河原, 黒橋, 2005)．また，林部らは用例をクラスタリングすることで格パターンを抽出し，述語に対してより頑健に格フレームを割り当てることに成功している (林部ら, 2015)．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．

表 1: 「オープンする:動 1」の格フレーム

格	格要素	出現回数	ベクトル
ガ	店	129	$\phi_{店}$
	カフェ	38	$\phi_{カフェ}$
	ショップ	37	$\phi_{ショップ}$
	...	...	...
	合計	231	
ニ	近く	6	$\phi_{近く}$
	跡地	2	$\phi_{跡地}$
	ところ	2	$\phi_{ところ}$
	...	...	...
	合計	57	

京大格フレームには表 1 に示すとおり，格要素として入りうる名詞句の出現形とその頻度が含まれている．表 1 を利用して例 (1) のゼロ照応解析を行う際,「オープンする」という動詞のガ格の格要素候補として「カフェテリア」が考えられる．しかし，従来手法 (河原, 黒橋, 2007) では表中の格要素の例に「カフェテリア」が含まれないことから，これは素性の一部が発火しない候補となる．この問題を改善するために解析対象の格要素と格フレームに現れる格要素群の JUMAN カテゴリ，固有表現，意味クラスを使用し汎化を行う手法が提案されている (Sasano, et al., 2011)．本稿では格パターンごとの格要素群を分散表現を用いることでベクトル化，平滑化を行い，解析対象の格要素とのコサイン類似度，及び格フレーム中の格間の差分ベクトルと解析対象の格要素間の差分ベクトルの類似度を素性として用いることで，より柔軟なゼロ照応解析を試みる．

[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林部らのクラスタリング手法で作成された未公開の新しい版である．

2 関連研究

日本語ゼロ照応解析に関しては，これまでに様々な手法が提案されてきた．ゼロ代名詞の検出も含めたゼロ照応解析を述語項構造解析の一部として処理し，「照応なし」と「外界ゼロ照応」を同一に扱う研究も多い．そのうち，ゼロ代名詞を格ごとに独立して扱っているモデル (Hayashibe et al., 2011; 林部ら, 2014) と，これらを同時に解決するモデル (Sasano, et al., 2011) がそれぞれいくつか提案されている．「照応なし」と，「外界ゼロ照応」を区別した研究としては (平, 永田, 2013; 萩行ら, 2014) などがある．

本研究は萩行ら (2014) の手法をベースに，「照応なし」と「外界ゼロ照応」を区別せずにすべての格を同時に解析するモデルを構築した．

3 ベースラインモデル

3.1 前処理

ベースラインモデルでは，Sasano, et al. (2011); 萩行ら (2014) と同様に，まず文書全体に対して形態素解析，固有表現抽出，構文解析を行う．これには JUMAN Ver.7.01[2]，KNP Ver.4.16[3] を用いた．その後，文頭から出現順に述語単位でゼロ照応解析を行う．

学習したランキングモデルの識別関数を用いてすべての述語項構造候補をスコア付けし，点数の最も高い候補を出力する．

本研究では共参照解析を行わない代わりに，コーパスに付与された共参照情報をもとに出力を評価し，正しい照応先と共参照関係にある先行詞のいずれかを対応付けることが出来ていたならば正解とする．各述語項構造は格フレーム (cf) とその格フレームの格スロットと照応先の対応付け (a) として表現する．

3.2 素性による述語項構造の表現

入力テキスト t の解析対象述語 p に格フレーム cf を割り当て，その格フレームの格スロットと先行詞の対応付けを a とした述語項構造を表現する素性ベクトルを $\phi(cf, a, p, t)$ とする． $\phi(cf, a, p, t)$ は以下のように 4 つのベクトルの結合として表現する[4]．

$$\begin{aligned}\phi(cf, a) \quad = \quad & (\phi_{overt}(cf, a_{overt}), \\ & \phi_{case}(cf, ガ \leftarrow e_{ガ}), \\ & \phi_{case}(cf, ヲ \leftarrow e_{ヲ}), \\ & \phi_{case}(cf, ニ \leftarrow e_{ニ}))\end{aligned}$$

ここで $\phi_{overt}(cf, a_{overt})$ は直接係り受けがある述語項構造を表わす素性ベクトルであり， $\phi_{case}(cf, c \leftarrow e)$ は格 c に先行詞 e が割り当てられることを表わす素性ベクトルである．格 c に先行詞 e が対応付けられない「照応なし」，「外界ゼロ照応」の場合，各格に対応する素性ベクトル $\phi_{case}(cf, c \leftarrow e)$ は cf と c のみに依存する素性以外をすべてゼロとする．$\phi_{overt}(cf, a_{overt})$ には 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を用いる． ϕ_{case} を構成する各素性ベクトルには萩行ら (2014) の素性を利用した．

[2] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[3] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP
[4] 以降，p, t については適宜省略する

4 提案手法

本研究では，上記のベースラインモデルで用いる素性に加えて，語の分散表現を用いた素性を組み込む．具体的には，格フレーム中の格要素の例から，その述語が格要素としてとりうる語の分散表現を計算し，それと入力文章中の格要素候補の分散表現との距離を新しい素性として導入する．格要素の分散表現を計算する手法として「格フレーム内平均ベクトル」，「述語内平均ベクトル」の二種類を提案する．さらに格フレーム内平均ベクトルの異なる格要素間の差 (「格間差分ベクトル」) も素性として利用する．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[5]．以下，これらの素性について説明する．

4.1 格フレーム内平均ベクトル

京大格フレームの情報を用いて，格フレームの格 c の格要素となりうる語の分散表現を以下のように計算する．表 1 の格要素となる語の例 (w) から word2vec を用いてそのベクトル表現 (ϕ_w) を計算し，これらの語ベクトル (ϕ_w) を w が格 c をともなって出現する回数 ($count(cf, c, w)$) (表 1 の「出現回数」の列) で重み付けした平均ベクトルを式 (1) によって算出する．ただし，$W(cf, c)$ は格フレーム cf と格 c に対応して京大格フレーム中に出現する格要素の全体である．

$$\overline{\phi}_{cf\_c} = \frac{\sum_{w \in W(cf,c)} count(cf, c, w) \cdot \phi_w}{\sum_{w \in W(cf,c)} count(cf, c, w)} \quad (1)$$

こうして算出された格フレーム中で述語が格 c に取りうる格要素の平均ベクトルと入力文章中の格要素候補の分散表現のコサイン類似度を計算し，素性とする．

4.2 述語内平均ベクトル

4.1 によって，格パターンごとに各格要素として取りうる語の分散表現を得ることができる．ここでは，さらに述語が各格要素に取りうる語を格パターンにわたって平均化する．まず，先に求めた格フレーム内平均ベクトルから述語ごとの平均ベクトルを計算する．格フレーム内平均ベクトルの情報 ($\overline{\phi}_{cf\_c}$) と述語に対応する出現回数 ($count(p, c, w)$) の総和の情報から述

[5] 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．

語 p と格 c ごとに対応する平均ベクトルを式 (2) で算出する．ただし，$F(p,c)$ は述語 p と格 c に対応して京大格フレーム中に出現する格フレームと格要素の組合せの全体である．

$$\overline{\phi}_{p\_c} = \frac{\sum_{cf,w \in F(p,c)} count(cf,c,w) \cdot \overline{\phi}_{cf\_c}}{\sum_{cf,w \in F(p,c)} count(cf,c,w)} \quad (2)$$

こうして算出された述語内平均ベクトルと入力文章中の格要素候補の分散表現のコサイン距離を計算し，素性とする．この素性を採用した意図は，格フレームごとの平均ベクトルよりも粗いスムージングの効果を比較，検討するためである．

4.3 格間差分ベクトル

先に求めた格フレーム内平均ベクトルから格フレームごとに格 c と格 c' の間の差分ベクトルを式 (3) で計算する．ただし，格 c，格 c' の組合せは，(ガ格，ニ格)，(ニ格，ヲ格)，(ヲ格，ガ格) の三種類である．

$$\phi_{cf\_cc'} = \overline{\phi}_{cf\_c} - \overline{\phi}_{cf\_c'} \quad (3)$$

こうして算出された格間差分ベクトルと対応する二つの格要素候補の差分ベクトルのコサイン距離を計算し，素性とする．この素性は前出の二種類の平均ベクトルとは別の性質を持っており，ここでは格フレームに対応する二つの格要素間の性質差を表現している．例えば「教える」という動詞に対して，格要素の組合せとしてはガ格に「教師」，ニ格に「生徒」が埋まる例が考えられる．この時，格要素間の性質の違いとしてガ格には「知識」を持つ名詞が埋まりやすく，一方でヲ格に埋まる格要素には「知識」が欠けているような名詞が埋まりやすい．同様にして「経験」の差分である「上司」と「部下」の組合せなどが「教える」の格要素においては同時に現れる傾向にあるだろうと考えられる．このような格要素間の関係を表現することで，複数格を同時解析するメリットの強化を意図している．

5 実験

5.1 データ

実験データとして，BCCWJ コアデータ[6]のうち動詞を含む文章から，ジャンルごとのデータ総量が同等となるよう，短い方から約 328 文章を選んで採用した．ただし，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．また，格が交代する受身，使役などの助動詞を伴って現れる述語も解析の都合上，対象としていない (飯田ら, 2010; 植田ら, 2015)．動詞の総数は 1,631 例で，5 分割交差検定により評価を行った．

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

表 2: 素性の組合せ

素性	A0	A1	A2	A3	A4	A5	A6	A7
出現形素性	o		o	o	o	o	o	o
カテゴリ素性		o	o					o
$\overline{\phi}_{cf\_c}$				o		o		
$\overline{\phi}_{p\_c}$					o		o	o
$\phi_{cf\_cc'}$						o	o	o
その他の素性	o	o	o	o	o	o	o	o

5.2 素性設計

ベースラインモデルが取り扱う素性のうち，本稿が提案する素性と互換性があるものとして，格フレーム素性がある．これらは格要素として京大格フレームに記載された出現形そのものの出現回数のみを取り扱う「出現形素性」と，格要素を JUMAN カテゴリ，固有表現によって汎化した上で取り扱う「カテゴリ素性」に分けられる．また，既存素性には格フレーム素性以外に文脈素性などの「その他の素性」が含まれる．既存素性と提案素性との組合せで表 2 のように A0–A7 の 8 種類の解析機を用意し，比較する．

5.3 学習・評価

ランキング学習には SVM^{rank}[7]を使用した．入力ごとの出力候補に対して正負の正解がラベル付けされた事例から，ランキング学習によって識別関数を学習し，推定の際には識別関数の出力が最も高くなるものを出力する．A2 がベースラインモデルである．これらに加えて KNP に anaphora オプションをつけた際のゼロ照応解析出力結果とも精度を比較する．[8]

5.4 結果と考察

実験結果を表 3 に示す．

格要素の平均化　文内照応では，カテゴリ素性の代わりとして格フレーム内平均ベクトルを使用した A3 がベースラインモデル A2 より高い精度を示している．しかし文間照応においてはこれは A2 より低い精度となった．より粗いスムージングである述語内平均ベクトルを使用した A4 は，全体的には A2 に劣るものの A3 と比較した時，文内では A3 より精度が低く，文間では逆に A3 より高くなっている．文間照応には語彙的選好に加えて構文的選好が有用であるが，各素性に対して学習された識別関数の重みを分析した結果，A3 ではより強力な語彙的選好を反映する提案素性を用いたため文脈素性をうまく学習できなかったものと考えられる．

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

[8] ただし KNP の出力はあくまで共参照解析なども含めて同時に行われたものであり，本稿における提案手法との純粋な比較はできないため，あくまで一般的に使用される解析機の性能目安としての採用である．

表 3: 実験結果の精度 (F 値)

	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
平均事例数	42.2	41.2	16.0	99.4	34.4	18.0	4.8	57.2	76.6	59.2	20.8	156.6
A0 (出現形)	.338	**.342**	**.230**	.315	**.190**	.227	.000	.194	.280	.310	**.202**	.277
A1 (カテゴリ)	.360	.326	.205	.311	.158	**.261**	.000	.187	.280	.309	.190	.274
A2 (出現形, カテゴリ)	.364	.310	.199	.307	.173	.246	.000	.187	.291	.293	.176	.271
A3 (出現形, cf_c)	.386	.336	.200	**.327**	.176	.204	.000	.182	**.302**	.298	.186	.281
A4 (出現形, p_c)	.320	.306	.202	.288	**.190**	.200	.000	.187	.269	.280	.179	.257
A5 (出現形, cf_c, cf_cc')	.381	.325	.214	.324	.166	.230	.000	.185	.296	.299	.197	.281
A6 (出現形, p_c, cf_cc')	.346	.334	.200	.308	.187	.207	.000	.195	.285	.303	.189	.274
A7 (出現形, カテゴリ, p_c, cf_cc')	.364	.338	.207	.319	.181	.241	.000	**.197**	.294	**.313**	.193	**.283**
KNP	**.437**	.212	.117	.314	.099	.118	.000	.100	.309	.187	.084	.242

格間差分ベクトル　A3 と A5 を比べると，格間差分ベクトルの素性を追加しても精度はほとんど変わらない．一方で A4 と A6 を比べると，この素性を追加することによって精度が向上している．文間照応においてはこの素性の追加によって，ガ格の精度は下がるがヲ格の精度が上がることで，文間照応全体の精度が上がっている．これは素性の特徴が上手く活用された結果であると考えられる．また，A6 にさらにカテゴリ素性を加えた A7 では文間照応，ヲ格，全体において最もよい精度を示していることから，粒度の異なる複数の汎化素性を導入することは比較的精度が低い格の精度改善に有用であると考えられる．

6　おわりに

本稿では既存の格フレームの格要素を分散表現で表わし，その格フレームを用いて日本語ゼロ照応解析に利用する手法を提案し，評価実験によってその有効性を確認した．格要素の分散表現として，格要素事例の分散表現を格フレーム内で平均化する手法，述語内で平均化する手法，格間の格要素の差分を計算する手法を提案し，それぞれの有効性について議論した．今後は，学習データを増やしたりコーパスを変えた際の本手法精度の確認と，分散表現を利用した格要素候補削減の実験を行う予定である．

謝辞

(萩行ら, 2014) に関して詳細な情報をご教示くださった萩行正嗣氏，(Sasano et al., 2008) の出現格・位置カテゴリの詳細をご教示くださった笹野遼平氏，京大格フレームと KNP の仕様をご教示くださった河原大輔氏，林部祐太氏に厚く御礼申し上げます．

参考文献

松林優一郎, 中山周, 乾健太郎. 日本語述語項構造解析タスクにおける項の省略を伴う事例の分析. 自然言語処理, Vol.22, No.5, pp.433-463, 2015.

河原大輔, 黒橋禎夫. 格フレーム辞書の漸次的自動構築. 自然言語処理, Vol.12, No.2, pp.109-131, 2005.

河原大輔, 黒橋禎夫. 自動構築した大規模格フレームに基づく構文・格解析の統合的確率モデル. 自然言語処理, Vol.14, No.4, pp.67-81, 2007.

林部祐太, 河原大輔, 黒橋禎夫. 格パターンの多様性に頑健な日本語格フレーム構築. 情報処理学会第 224 回自然言語処理研究会, NL-224-14, pp.1-8, 2015.

Yuta Hayashibe, Mamoru Komachi and Yuji Matsumoto. Japanese Predicate Argument Structure Analysis Exploiting Argument Position and Type. In Proc. of IJCNLP, pp.201-209, 2011.

林部祐太，小町守，松本裕治. 述語と項の位置関係ごとの候補比較による日本語述語項構造解析. 自然言語処理, Vol.21, No.1, pp.3-26, 2014.

Ryohei Sasano, Daisuke Kawahara and Sadao Kurohashi. A Fully-Lexicalized Probabilistic Model for Japanese Zero Anaphora Resolution. In Proc. of COLING 2008, pp.769-776.

Ryohei Sasano and Sadao Kurohashi. A discriminative approach to Japanese zero anaphora resolution with large-scale lexicalized case frames, In Proc. of IJCNLP, pp.758-766, 2011.

平博順, 永田昌明. 述語項構造解析を伴った日本語省略解析の検討. 言語処理学会第 19 回年次大会発表論文集, pp.106-109, 2013.

萩行正嗣, 河原大輔, 黒橋禎夫. 外界照応および著者・読者表現を考慮した日本語ゼロ照応解析. 自然言語処理, Vol.21, No.3, pp.563-600, 2014.

Mikolov, T., Kai, C., Corrado, G. and Dean, J. Efficient Estimation of Word Representations in Vector Space, Proceedings of Workshop at International Conference on Learning Representations, 2013.

飯田龍, 小町守, 井之上直也, 乾健太郎, 松本裕治. 述語項構造と照応関係のアノテーション： NAIST テキストコーパス構築の経験から. 自然言語処理, Vol. 17, No. 2, pp. 25-50, 2010.

植田禎子, 飯田龍, 浅原正幸, 松本裕治, 徳永健伸. 『現代日本語書き言葉均衡コーパス』に対する述語項構造・アノテーション. 第 8 回コーパス日本語学ワークショップ予稿集, pp. 205-214. 国立国語研究所, 2015.