述語項構造解析における単語埋め込みの影響

珊瑚 彩主紀　　西川 仁　　徳永 健伸
東京工業大学 情報理工学院
{sango.m.ab@m, hitoshi@c, take@c}.titech.ac.jp

1 導入

日本語述語項構造解析は，対象とする述語の各格の項を予測するタスクであり，様々な自然言語処理アプリケーションの土台となる技術である．

例 (1) メールを 書いて $_{v_1}$ 送ったよ $_{v_2}$． 読んでね $_{v_3}$．

述語 \ 格	ガ格	ヲ格	ニ格
v_1: 書いて	[書き手]	メール	[読み手]
v_2: 送った	[書き手]	(メール)	[読み手]
v_3: 読んで	[読み手]	((メール))	none

表 1: 例 (1) の述語項構造解析結果

例 (1) は，3 つの述語 (v_1，v_2，v_3) と 1 つの明示的な項候補 (メール) を含んだテキストである．例 (1) を述語項構造解析した結果は表 1 のようになる．ここで，角括弧で囲まれた要素は外界照応，丸括弧は文内ゼロ照応，二重丸括弧は文間ゼロ照応である．v_1 のヲ格の項である「メール」は，格標識「を」によって明示的に示されており，v_1 との係り受け関係を持っている．このような名詞は，括弧をつけないで示している．

述語項構造解析の先行研究では，形態素及び，構文解析から得られた様々な特徴を利用している [1, 2, 3, 4]．近年は，中間解析を必要としない end-to-end の手法による解析もある [5, 6]．だが，いずれも，素性の変更やモデルの改良に注力しており，単語埋め込み表現に焦点を当てた研究はない．述語項構造解析では，同じ表層表現の単語でも，文中で使われている語義が異なる場合，異なる項構造を取ることがある．そのため，適切な単語埋め込み表現を使用することは重要である．

本研究では，単語埋め込み表現の述語項構造解析への影響を調べる．本研究でのポイントは大きく二つある．第一に，述語項構造解析には分野依存性があるとわかっている [7]．そのため，単語埋め込み表現を作成する際の分野 (メディア) は，述語項構造解析の分野依存性に影響を与えるのかを調べる．今回は，同一の手法で，異なるメディアから作成された単語埋め込み表現を作成し，解析精度の比較を行った．第二に，述語項構造解析に使う際に最適な単語埋め込み表現の学習手法は何かを調べる．そのため，同一のメディアから，複数の異なる手法により単語埋め込み表現を作成し，解析精度の比較を行った．

2 問題設定

本研究では，文内述語項構造解析を扱い，文間ゼロ照応については解析対象としない．ただし，外界照応については解析対象とするため，外界三人称と文間ゼロ照応については，何らかの項は取るがその項は文中で明示されておらず特定できないことを意味する単一の unknown ラベルとして扱う．まとめると，本研究では，文内直接係り受け (intra(dep))，文内ゼロ照応 (intra(zero))，外界一人称 (exo1)，外界二人称 (exo2)，解析対象の述語は項を取らない (none)，そして外界三人称と文間ゼロ照応をまとめた unknown を扱う．

3 深層リカレントモデル

我々は，以下の三層からなるリカレントニューラルネットワーク (RNN) モデルを用いて，日本語述語項構造解析を実現する．

入力層 単語を特徴ベクトルに変換する．
隠れ層 bi-directional RNN 層と全結合層．
出力層 ソフトマックス関数により，2 値分類を行う．

我々のモデルは，各単語毎に項であるか否かを示す確率値を出力する．その単語がターゲットの述語に対する解析対象の格の項であるか否かを示すため，それぞれの格に対して別々にモデルを用意する必要がある．図 1 に，モデルの概要を示す．これは，次のように形式的に表せる．

$$\overline{\boldsymbol{x}} = \boldsymbol{w}_a \oplus \boldsymbol{w}_f \oplus \boldsymbol{b}_f \quad (1)$$

$$\boldsymbol{h}^1 = \mathrm{BiLSTM}(\overline{\boldsymbol{x}}) \quad (2)$$

$$\boldsymbol{h}^2 = \mathrm{linear}(\boldsymbol{h}^1) \quad (3)$$

$$p = \mathrm{softmax}(\boldsymbol{h}^2) \quad (4)$$

我々のモデルは，1 文ずつ入力文を受け取る．入力文中の単語 $\{w_t\}_0^T$ は，対応する単語の特徴ベクトル $\{\overline{\boldsymbol{x}}_t\}_0^T$ に変換される．単語の特徴ベクトル $\overline{\boldsymbol{x}}$ は，単語埋め込みベクトル $\boldsymbol{w}_a$，品詞埋め込みベクトル $\boldsymbol{w}_f$，及

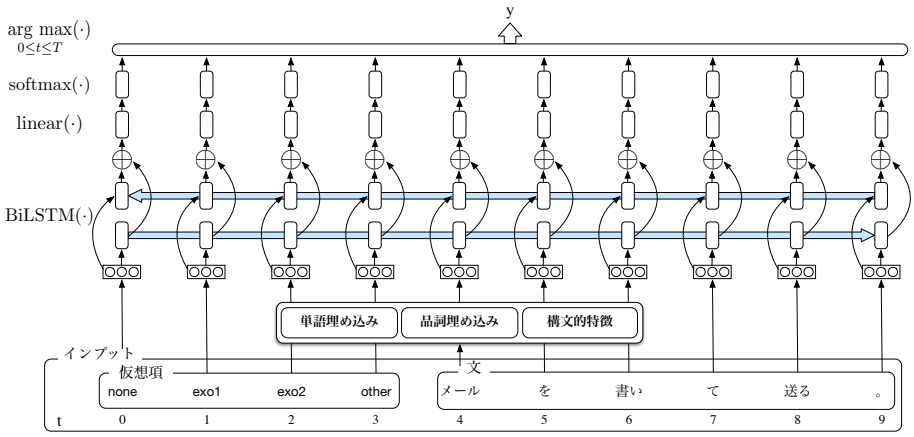

図 1: 日本語述語項構造解析のための深層リカレントモデル

び，構文的特徴ベクトル $\boldsymbol{b}_f$ を連結したベクトルとして表現される．特徴ベクトル $\overline{\boldsymbol{x}}$ は，双方向型の Long short-term memory recurrent neural network（BiLSTM）に入力される．そして，BiLSTM(·) は，各単語に対して，ベクトル $\boldsymbol{h}^1$ を計算し出力する．linear(·) 関数は，$\boldsymbol{h}^1$ を受け取り，$\boldsymbol{h}^2 = (h_0^2, h_1^2)$ を出力する．h_0^2 は単語が述語の項となる確率であり，h_1^2 は単語が述語の項でない確率である．最後に softmax(·) 関数は，$\boldsymbol{h}^2$ を受け取り確率 p を出力する．

3.1 入力層

単語埋め込み，品詞埋め込み，および構文的特徴の 3 つの特徴を定義する．

3.1.1 単語埋め込み

今回は，単語埋め込みなし，ランダム，そして，Word2Vec，FastText，ELMo それぞれの手法で作成された単語埋め込みベクトルを用意し，比較実験を行った [8, 9, 10]．詳細については， 4 節にて説明する．

3.1.2 品詞埋め込み

各単語には，CaboCha[1]による解析結果と同様の人手で修正されたタグがつけられている．本研究では，品詞埋め込みとして，単語の品詞タグ，及び，3 種類の品詞細分類タグ，活用型，活用形の 6 種類のタグそれぞれに対して，5 次元のランダムベクトルを割り当てた．したがって品詞埋め込みは，6 層のベクトルを連結することによって作る 30 次元のベクトルによって表す．欠落している層は，ゼロベクトルで埋める．

3.1.3 構文的特徴

構文的特徴ベクトルには，以下の 4 種類の特徴が含まれている．(1) 単語が各文節において主辞か否かを示す二値ベクトル，(2) 単語が各文節において機能語か否かを示す二値ベクトル，(3) コーパスに注釈付けられている文節に基づく，文頭からの文節距離を示す整数値の特徴ベクトル（入力文の最初の文節の単語は，この値がゼロとなる），(4) 係り先の文節番号（係り先がない場合は，この値が -1 となる），(5) 解析されるターゲットの述語からの形態素距離を示す整数値の特徴ベクトル，(6) その単語が解析対象の述語であるか否かを示す二値ベクトルの 6 種類である．

[1]CaboCha/南瓜: Yet Another Japanese Dependency Structure Analyzer http://taku910.github.io/cabocha/

3.1.4 仮想項

none，exo1，exo2，unknown という 4 つのラベルを出力するために，文の先頭の単語の前にそれらを示す仮想項を追加した．これらの仮想項を以下のように割り当てる．なお，先行研究では，仮想項に対して，使用する単語埋め込み表現の一人称代名詞，二人称代名詞間の類似度から単語を選出していたが，本研究では，使用する単語埋め込みが複数あるため，すべての単語埋め込みに対して以下のように統一して定めた [7].

none none に対してはゼロベクトルを割り当てた．
exo1 「私」の単語ベクトルを割り当てる．
exo2 「あなた」の単語ベクトルを割り当てる．
unknown 「これ」の単語ベクトルを割り当てる．

3.2 隠れ層

隠れ層では，各時刻 t に，特徴ベクトル $\overline{\boldsymbol{x}}_t$ と $\boldsymbol{h}_{t-1}^f$ を前向き LSTM（LSTM^f）に入力し $\boldsymbol{h}_t^f$ を計算する．逆に，各時刻 t に，特徴ベクトル $\overline{\boldsymbol{x}}_t$ と $\boldsymbol{h}_{t+1}^b$ を後ろ向き LSTM（LSTM^b）に入力し $\boldsymbol{h}_t^b$ を計算する．BiLSTM は，各時刻 t で $\boldsymbol{h}_t^f$ と $\boldsymbol{h}_t^b$ を連結し，$\boldsymbol{h}_t^1$ を出力する．

$$\begin{aligned} \boldsymbol{h}_t^1 &= \mathrm{BiLSTM}(\overline{\boldsymbol{x}}_t) \\ &= \mathrm{LSTM}^f(\overline{\boldsymbol{x}}_t, \boldsymbol{h}_{t-1}^f) \oplus \mathrm{LSTM}^b(\overline{\boldsymbol{x}}_t, \boldsymbol{h}_{t+1}^b) \end{aligned} \quad (5)$$

次に，2 次元ベクトル $\boldsymbol{h}_t^2$ を得るために，linear$(\cdot)$ 関数に $\boldsymbol{h}_t^1$ を入力する．

$$\boldsymbol{h}_t^2 = \text{linear}(\boldsymbol{h}_t^1) \tag{6}$$

3.3 出力層

出力層では，単語が解析対象とする述語の項であるか否かを判断する．softmax$(\cdot)$ 関数は，2 次元ベクトル $\boldsymbol{h}_t^2$ を単語がどの程度対象とする述語の項としてふさわしいかを示す確率値に変換する．

$$p_t = \text{softmax}(\boldsymbol{h}_t^2) \tag{7}$$

p_t は時刻 t の単語が項である確率を示す．モデルは，最も高い確率 p_y を持つ単語を項として選択する．

$$y = \underset{0 \leq t \leq T}{\arg\max}(p_t) \tag{8}$$

4 単語埋め込み

本研究では二種類の比較実験を行っている．それぞれの比較実験を行うために，異なる手法により作成した複数の単語埋め込みを用意した．第一に，単語埋め込みによる述語項構造解析の分野依存性を調べるために，単語埋め込みの学習手法は Word2Vec で統一し，学習データを変更したものを用意した．使用したデータはいずれも現代日本語書き言葉均衡コーパス（BCCWJ）[2]である [11]．述語項構造の情報がアノテーションされている 6 種の各メディア（Yahoo!知恵部，Yahoo!ブログ，白書，書籍，雑誌，新聞）のテキストのみから学習した単語埋め込みに加え，BCCWJ の全データを使って作成した単語埋め込みを用意した．また，出所の異なる大規模なデータから作成された単語埋め込みとして，日本語 Wikipedia から作成された単語埋め込み[3]を用意した [12].

第二に，単語埋め込みの学習手法の違いが述語項構造解析に与える影響を調べるために，学習データを BCCWJ 全データで統一し，以下の 6 種の異なる手法により作成した単語埋め込みを比較した．(1) 単語埋め込みなし (None)，(2) ランダムに初期化 (Random)，(3) FastText により学習，(4) Word2Vec により学習，(5) ELMo により学習 (200 次元)，(6) ELMo により学習 (1024 次元)

いずれの単語埋め込み表現も次元数は 200 とし，ELMo のみ，他の手法と比較するための 200 次元と，元論文により推奨されている 1024 次元の二種類を用意した．Word2Vec 及び，FastText は，実行時のパラメータとして，window 10，min_count 5 としている．その結果，得られた語彙サイズ，及び，BCCWJ-PAS データセットに対する未知語の割合を表 2 に示す．

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

[3] Japanese Wikipedia Entity Vector http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

5 実験

5.1 実験設定

現代日本語書き言葉均衡コーパス（BCCWJ）の述語項構造がアノテーションされているデータを使い評価を行った．各メディア毎にデータを訓練用に 70%，開発用に 10%，テスト用に 20% に分け使用した．各モデルを最大 30 エポック訓練し，開発用データにおいて最も F1 値が高いモデルを使用した．なお，単語埋め込みの比較のため，実験結果はその与える影響が大きいと考えられる文内直接係り受け及び，文内ゼロ照応の F1 値をまとめた結果を示している．

5.1.1 ハイパーパラメータ

BiLSTM のドロップアウト率は 0.2，バッチサイズは 16 とした．weight decay は 0，Adam を使い α が 0.001，β が 0.9 とし，最適化した．

5.2 実験結果

単語埋め込みの学習手法は Word2Vec で統一し，その学習データを変更した比較結果を表 3 に，単語埋め込みの学習データを統一し，学習手法を変えた比較結果を表 4 に示した．行見出しが述語項構造解析の解析対象のメディアを示している．

表 3 の各メディアのみから学習された単語埋め込みを使った結果に対して比較すると，表 2 からわかるように，語彙数，未知語の割合が大きく異なるため，メディアによって使われている語義が異なることが予想されるが，結果からは特にメディア依存性は見られない．知恵袋，白書，書籍に対して，BCCWJ 全体を使って学習された単語埋め込みが最も高いスコアとなった．これは，未知語が少なかったためではないかと考えられる．

表 3 の結果から，同じ 200 次元数では，いずれの手法も優位な差はない．これは，品詞埋め込み，構文的特徴が効いているためであると考えられるが，単語埋め込みをなくした None ではスコアが大きく下がっていることから，Random であっても単語埋め込みが必要なことがわかる．また，単語埋め込みの次元数が異なるため，純粋な比較にはならないが，述語項構造解析の解析対象が雑誌，新聞の時と BCCWJ 全体を解析対象とした時は，1024 次元の ELMo の精度が最も高かった．これは，ELMo では，その性質上，未知語が発生せず，また 200 次元では区別できなかった語義曖昧性が 1024 次元になったことで解決できたためではないかと考えられる．だが，反面，ELMo を使うと学

	学習メディア							
	知恵袋	ブログ	白書	書籍	雑誌	新聞	BCCWJ 全体	Wikipedia
語彙サイズ (形態素数)	37,848	53,139	14,301	83,614	35,211	12,956	148,048	1,015,474
未知語の割合 (%)	4.33	3.29	10.36	2.32	4.10	8.01	1.64	4.69

表 2: 未知語の割合

	Word2Vec							
	知恵袋	ブログ	白書	書籍	雑誌	新聞	BCCWJ 全体	Wikipedia
知恵袋	74.45	74.67	74.60	75.02	75.46	74.85	**76.02**	75.06
ブログ	70.12	**71.20**	69.31	69.05	69.61	71.00	69.60	69.85
白書	72.84	71.99	72.39	72.32	72.91	72.76	**73.48**	72.64
書籍	76.43	75.82	76.40	76.37	75.71	76.79	**77.02**	76.34
雑誌	75.96	75.49	76.01	**76.59**	76.29	74.37	74.82	74.74
新聞	72.32	73.02	**73.22**	72.02	72.81	72.99	72.24	72.41
BCCWJ 全体	**74.22**	74.10	74.10	73.50	74.09	74.06	73.88	73.79

表 3: 単語埋め込みのメディア依存性（F1 値）

	None	Random	FastText	Word2Vec	ELMo(200)	ELMo(1024)
	-			BCCWJ 全体		
知恵袋	48.40	74.53	74.72	**76.02**	75.10	74.94
ブログ	44.94	68.68	69.01	69.60	**71.19**	69.44
白書	46.30	**73.15**	71.35	73.48	71.09	72.41
書籍	49.51	75.31	75.43	**77.02**	76.48	76.66
雑誌	49.89	75.53	75.14	74.82	76.16	**76.60**
新聞	47.40	72.10	71.19	72.24	72.87	**73.71**
BCCWJ 全体	47.94	73.60	72.57	73.88	73.76	**74.51**

表 4: 単語埋め込みの学習法比較（F1 値）

習の収束がとても遅く，他の手法に比べ 2 倍以上の時間がかった．

6 結論

本論文では，日本語述語項構造解析における単語埋め込みの影響を調べた．結果から，BCCWJ 全データを使い，学習手法として ELMo を採用し，次元数を 1024 次元とした単語埋め込みを使うべきであることがわかった．今後はそれを単語埋め込みとして使用し，述語項構造解析の手法を改良する．

参考文献

[1] Yuichiroh Matsubayashi and Kentaro Inui. Revisiting the Design Issues of Local Models for Japanese Predicate-Argument Structure Analysis. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 128–133. Asian Federation of Natural Language Processing, 2017.

[2] Kenji Imamura, Ryuichiro Higashinaka, and Tomoko Izumi. Predicate-Argument Structure Analysis with Zero-Anaphora Resolution for Dialogue Systems. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 806–815, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[3] Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. Neural Network-Based Model for Japanese Predicate Argument Structure Analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1235–1244. Association for Computational Linguistics, 2016.

[4] Hiroki Ouchi, Hiroyuki Shindo, Kevin Duh, and Yuji Matsumoto. Joint Case Argument Identification for Japanese Predicate Argument Structure Analysis. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 961–970. Association for Computational Linguistics, 2015.

[5] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. Neural Modeling of Multi-Predicate Interactions for Japanese Predicate Argument Structure Analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1591–1600. Association for Computational Linguistics, 2017.

[6] Yuichiroh Matsubayashi and Kentaro Inui. Distance-Free Modeling of Multi-Predicate Interactions in End-to-End Japanese Predicate-Argument Structure Analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 94–106. Association for Computational Linguistics, 2018.

[7] Hitoshi Nishikawa Mizuki Sango and Takenobu Tokunaga. Effectiveness of Domain Adaptation in Japanese Predicate-Argument Structure Analysis. In *Pacific Asia Conference on Language, Information and Computation*, 2018.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, Vol. abs/1301.3781, , 2013.

[9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

[10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[11] Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. Balanced Corpus of Contemporary Written Japanese. *Lang. Resour. Eval.*, Vol. 48, No. 2, pp. 345–371, June 2014.

[12] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. Neural Joint Learning for Classifying Wikipedia Articles into Fine-grained Named Entity Types. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Posters*, pp. 535–544, 2016.