

意味役割付与のためのスパン選択モデル

大内 啓樹 * 2,3,a) 進藤 裕之^{1,2,b)} 松本 裕治^{1,2,c)}

概要: 本稿では、スパン (フレーズ) 選択に基づいた意味役割付与モデルを提案する。提案モデルは、入力文の可能なスパンをすべて考慮し、各意味役割ラベルに対するスコアを計算する。デコード時は、よりスコアの低いスパンを順に選択する。提案モデルの主な長所は以下の2点である: (i) 構文木などの情報を用いずにすべての可能なスパンを考慮可能, (ii) 入力系列全体を考慮したスパンレベルの特徴量を使用可能。評価実験の結果、本提案モデルは英語意味役割付与タスクにおける最高精度となる F1 値 87% を記録した。

1. はじめに

意味役割付与 (Semantic Role Labeling; SRL) は、「いつ、どこで、誰が、何を、誰に、どうした」といった述語と項の関係を同定する意味解析タスクである。例として、以下の文を考える。

入力文: She₁ kept₂ a₃ cat₄
正解の項: [A0] [A1]

入力として文とターゲットとなる述語が与えられ、その述語の項を予測する。述語 kept の項として、She を動作主 (A0), a cat を被動作主 (A1) として同定できれば正解となる。各項は1単語以上からなる単語のまとまりである。本稿ではこのまとまりを「スパン (span)」と呼ぶ。項となるスパン予測は、「BIO タグ系列ラベリング」や「ラベル付きスパン予測」に基づいて行われることが多い。

本節では、はじめに、上述した2つの主要な既存アプローチとその問題点を概観する。次に、提案モデルの概要と性能、分析結果について述べる。最後に、本稿の要約・貢献を記述する。

1.1 BIO タグ系列ラベリングに基づくアプローチ アプローチの概要

SRL は BIO タグ系列ラベリングとして解くことができる。各スパンの先頭とスパン内のそれ以外の単語はそれぞれ、「B」と「I」のタグを用いて表現される。スパン外の単語は「O」のタグで表現される。これらの BIO タグを各単語に対して予測する。最先端のニューラル SRL モデル [11], [32], [37] もこのアプローチを採用している。これらのモデルは、ニューラルネットによって計算した特徴量を利用することにより、高精度を記録している。

問題点

このアプローチに基づくモデル (BIO 型モデル) では、スパンレベルの特徴量が使用困難である。基本的なモデルは、各スパンを直接的に予測する代わりに、予測した BIO タグから各スパンを再構成することによって SRL の結果とする。つまり、各単語に対して特徴量抽出と BIO タグ予測を行なっている。モデルの構造上、単語レベルを超えたスパンレベルの特徴量を明示的に組み込むことが困難な点が問題として挙げられる。

1.2 ラベル付きスパン予測に基づくアプローチ アプローチの概要

もう一つの主流のアプローチは、ラベル付きスパン予測に基づくものである [6], [31]。このアプローチでは、意味役割ラベルとスパンのペア (ラベル付きスパン) に対して妥当性を表すスコアを割り当て、スコアの低いペアを出力する。このアプローチの長所は、スパンレベルの特徴量を使用可能な点である。

*本稿の一部は、著者が奈良先端科学技術大学院大学在籍中に書いたものである。

¹ 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

² 理化学研究所革新知能統合研究センター

Riken Center for Advanced Intelligence Project

³ 東北大学

Tohoku University

a) hiroki.ouchi@riken.jp

b) shindo@is.naist.jp

c) matsu@is.naist.jp

問題点

性能面で、スパン型モデルは BIO 型モデルに届かずにいる。その原因として、以下の2点が考えられる。

原因 (a) 構文解析の誤り伝搬

原因 (b) スパンレベルの特徴量抽出法

従来のスパン型モデルは、構文解析によって得られた構文木の構成素に基づいて候補となるスパンを同定する。これは、構文解析性能に大きく依拠するため、構文解析誤りが SRL の誤りの要因となることが多い。また、候補として同定したスパンの特徴量抽出方法として前向き伝搬型ニューラルネットが使用されている [6]。しかし、スパンの周辺文脈を局所的にしか考慮していないため、性能は限定的なものとなっている。

1.3 本稿のスパン型モデル

提案モデルの概要

上記の問題を踏まえ、本稿では、シンプルかつ効果的なスパン型モデルを提案する (2 節)。SRL をスパン選択問題として捉え、以下のような流れで解析する。

手順 (1) 入力文の可能なスパンをすべて列挙する。

手順 (2) 列挙した各スパンにスコアをつける。

手順 (3) スコアのより高いスパンを出力する。

手順 (1) では、入力文のとりうるすべてのスパンを考慮するため、従来法のように構文木を用いる必要はない。手順 (2) では、双方向型リカレントニューラルネットの利用によって、入力文全体を考慮した特徴量を各スパンに対して抽出し、スコア計算に用いる。これらの方法によって、従来のスパン型モデルが抱える問題の緩和が期待できる。

提案モデルの性能

提案モデルの性能評価のため、英語 SRL タスクで標準的に用いられるデータセット (CoNLL-2005/2012 Datasets[3], [21]) を用いて実験を行った。実験の結果、提案モデルが最先端の BIO 型モデルを上回る精度を記録した (4 節・表 2, 3)。さらなる拡張として、複数モデルにおけるスパン特徴ベクトルに基づくアンサンブル法を提案した (3.2 節)。結果として、最先端の単語埋め込み (ELMo [19]) とアンサンブル法を併用することにより、これまでの最高精度を 2 ポイント以上更新する F 値 87% を記録した (4 節・表 4)。

提案モデルの分析

より詳細な分析として、BIO 型モデルとの比較を通して、提案モデルの定量・定性的分析を行った (5 節)。分析の

結果、BIO 型モデルはスパン境界の同定性能が高い一方 (5 節・表 5)、スパン型の提案モデルは各スパンへのラベル付与性能が高いことがわかった (5 節・表 6)。また、最先端の単語埋め込み (ELMo) を用いることにより、スパン境界の同定性能が大きく補完されることもわかった (5 節・表 5)。

1.4 本稿の貢献

要約すると、本稿の主要な貢献は以下の3点である。

- スパン選択に基づく SRL モデルの提案
- スパン特徴ベクトルに基づくアンサンブル法の提案
- BIO 型 SRL モデルとの比較に基づく、提案モデルの長所・短所の定量的・定性的分析

2. モデル

本稿では、SRL をスパン選択問題として解く。この問題では、各意味役割ラベルに対して、入力文の可能なスパン候補集合から適切なスパンを選ぶ。

2.1 スパン選択問題

問題の定式化

入力として、 T 単語からなる文 $w_{1:T} = w_1, \dots, w_T$ とターゲットの述語の位置 p が与えられる。出力として、ラベル付きスパンの集合 $Y = \{(i, j, r)_k\}_{k=1}^{|Y|}$ を予測する。

入力: $X = \{w_{1:T}, p\}$

出力: $Y = \{(i, j, r)_k\}_{k=1}^{|Y|}$

各ラベル付きスパン $\langle i, j, r \rangle$ は、単語位置インデックス i と j ($1 \leq i \leq j \leq T$)、意味役割ラベル $r \in \mathcal{R}$ から構成される。

Y を予測するためのシンプルな方法として、各ラベル r に対して、可能なスパン集合 S から最大スコアのスパン (i, j) を選ぶ方法が考えられる。

$$\operatorname{argmax}_{(i,j) \in S} \operatorname{SCORE}_r(i, j), r \in \mathcal{R}. \quad (1)$$

ここで、関数 $\operatorname{SCORE}_r(i, j)$ は、各スパン $(i, j) \in S$ に対する実数値を返す (2.2 節)。入力文 $w_{1:T}$ における可能なスパンの数 $|S|$ は $\frac{T(T+1)}{2}$ であり、スパン集合 S は以下のように定義される。

$$S = \{(i, j) \mid i, j \in \{1, \dots, T\}, i \leq j\}.$$

注意点として、いくつかの意味役割は入力文に出現しない。このような出現しない意味役割を扱うため、本稿では述語のスパン (p, p) を NULL スパンとして定義する。出現しない意味役割に対しては、この NULL スパンを選ぶよう、モデルを学習する。^{*1}

^{*1} 述語自体が自身の項となることはないため、述語のスパンを NULL スパンとして定義した。

具体例

例として以下の文を考える．

$$\begin{array}{cccc} \text{She}_1 & \text{kept}_2 & \text{a}_3 & \text{cat}_4 \\ [\text{A0}] & & [\text{A1}] & \end{array}$$

$$Y = \{ \langle 1, 1, \text{A0} \rangle, \langle 3, 4, \text{A1} \rangle \}$$

入力文は $w_{1:4} = \text{She kept a cat}$ であり，述語位置は $p = 2$ である．正解のラベル付きスパン集合は $Y = \{ \langle 1, 1, \text{A0} \rangle, \langle 3, 4, \text{A1} \rangle \}$ である．

この例文の可能なスパン集合は以下の通りである．

$$\mathcal{S}_{w_{1:4}} = \{ (1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4) \} .$$

ここで，述語スパン (2, 2) は NULL スパンとして扱われる．これらの候補スパンの中から，各ラベルに対してスコアの最も高いスパンを選ぶ．意味役割ラベル A0 として，(1, 1) を選べば正解である．同様に，ラベル A1 として，(3, 4) を選べば正解である．残りすべての意味役割ラベルに対しては，NULL スパンである (2, 2) を選び，「該当スパンなし」とすれば正解である．

2.2 スコアリング関数

式 1 で用いられる関数 SCORE_r として，各ラベル r に対するスパン (i, j) の確率をモデル化する．

$$\begin{aligned} \text{SCORE}_r(i, j) &= P_\theta(i, j | r) \\ &= \frac{\exp(F_\theta(i, j, r))}{\sum_{(i', j') \in \mathcal{S}} \exp(F_\theta(i', j', r))} . \end{aligned} \quad (2)$$

ここで，パラメータ θ を持つ関数 F_θ は実数値を返し，スパン集合 $\mathcal{S}$ で正規化される．

パラメータの学習には，以下のような学習データセットを用いる．

$$\begin{aligned} \mathcal{D} &= \{ (X^{(n)}, Y^{(n)}) \}_{n=1}^{|\mathcal{D}|} , \\ X &= \{ w_{1:T}, p \} , \\ Y &= \{ \langle i, j, r \rangle_k \}_{k=1}^{|Y|} . \end{aligned}$$

本稿では，クロスエントロピーを最小化することにより，関数 F_θ のパラメータ θ を学習する．

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_{(X, Y) \in \mathcal{D}} \ell_\theta(X, Y) , \\ \ell_\theta(X, Y) &= \sum_{\langle i, j, r \rangle \in Y} \log P_\theta(i, j | r) . \end{aligned} \quad (3)$$

上記の $\ell_\theta(X, Y)$ は各事例に対する損失関数である．

2.3 関数 F_θ の構成

式 2 の F_θ は，3 つの関数に基づく：基本特徴量抽出関数 f_{base} ，スパン特徴量抽出関数 f_{span} ，ラベリング関数 f_{label} ．

$$\mathbf{h}_{1:T} = f_{base}(w_{1:T}, p) , \quad (4)$$

$$\mathbf{h}_s = f_{span}(\mathbf{h}_{1:T}, s) , \quad (5)$$

$$F_\theta(i, j, r) = f_{label}(\mathbf{h}_s, r) . \quad (6)$$

はじめに， f_{base} が入力文の各単語 $w_t \in w_{1:T}$ に対し，基本特徴量ベクトル $\mathbf{h}_t$ を計算する．次に，基本特徴量ベクトルの系列 $\mathbf{h}_{1:T}$ から， f_{span} が各スパン $s = (i, j)$ に対しスパン特徴量ベクトル $\mathbf{h}_s$ を計算する．最後に， $\mathbf{h}_s$ を用いて， f_{label} が各ラベル r に対しスコアを計算する．

これらの関数として，任意の関数を使用可能である．本稿では，ニューラルネットワークを使用した．ネットワークアーキテクチャについては 3 節で詳述する．

2.4 デコード

argmax (式 1) により，各ラベルに対して 1 つのスパンが選ばれる．これは，計算効率が良い反面，以下の問題に直面する．

- (a) スパン同士がオーバーラップする可能性がある．
- (b) 1 つのラベルに対し，複数のスパンを選択できない．

(a) に関しては，例えば， $\langle 1, 3, \text{A0} \rangle$ と $\langle 2, 4, \text{A1} \rangle$ が選ばれた場合，2 つのスパンが一部重なってしまう．(b) に関しては，以下の例を考える．

$$\begin{array}{cccc} \text{He} & \text{came} & \text{to the U.S.} & \text{yesterday at 5 p.m.} \\ [\text{A0}] & [\text{A4}] & [\text{TMP}] & [\text{TMP}] \end{array}$$

この例では，ラベル TMP が 2 つのスパン (yesterday と at 5 p.m.) に付与される．意味役割ラベルは，(i) コアラベルと (ii) 付加詞 (Adjunct) ラベルに大別される．上記の A0・A4 のようなラベルはコアラベルであり，文を成り立たせる上で必須の項とされる．一方，TMP のようなラベルは付加詞ラベルであり，文に付加的な意味を加える任意の項とされる．付加詞ラベルは，上記の例のように複数のスパンに付与されることがある．

本稿では，上記の問題を解消するための貪欲探索法を利用する．この探索では，以下の 2 つの制約を守りながら，より高いスコアのラベル付きスパンを順に選ぶ．

- 重複制約 選択済みのスパンとオーバーラップ (重複) するスパンを選ばない
- 個数制約 付加詞ラベルに対して複数のスパンを選べるが，コアラベルに対しては 2 つ以上のスパンは選ばない

本探索法の詳細な記述として，付録 A.3 に擬似コードとその説明を記す．

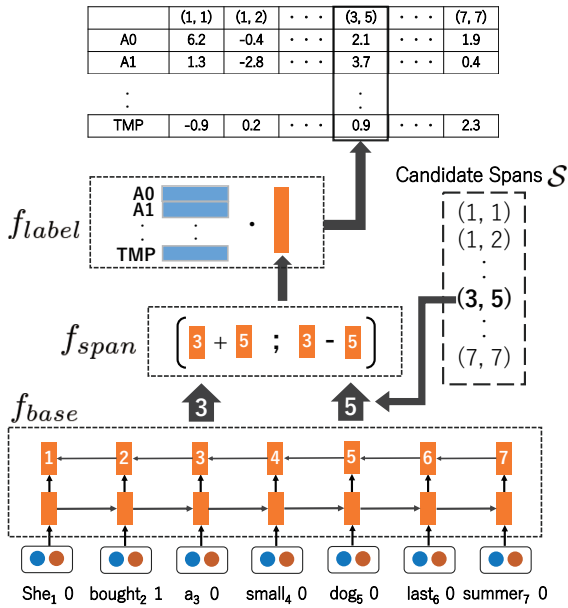


図 1 提案モデルのネットワークアーキテクチャ。

3. ネットワークアーキテクチャ

各スパンのスコア計算のため、2.3 節で 3 つの関数 (f_{base} , f_{span} , f_{label}) を導入した。本節では、これらの具体的な定義を記述する。

3.1 BiLSTM-Span モデル

図 1 は提案モデルの構成の概略である。図 1 下部は、 f_{base} として双方向型 LSTM (BiLSTM) [8], [9], [25] を用いて、入力から基本特徴量ベクトルを計算している過程を示している。図 1 中部では、 f_{span} がスパン (3, 5) に対するスパン特徴量を計算している。図 1 上部では、 f_{label} がスパン特徴量をもとにスパン (3, 5) に対する各ラベルのスコアを計算している。以降でこれらの 3 つの関数を詳述する。

基本特徴量抽出関数 f_{base}

f_{base} として、BiLSTM を用いる。

$$f_{base}(w_{1:T}, p) = \text{BiLSTM}(w_{1:T}, p) .$$

BiLSTM としていくつかの異なるものが提案されている。本稿では先行研究のニューラル SRL モデル [11], [37] に使用されたものを利用する。以降では BiLSTM の概要のみを説明し、より詳しい説明は付録 A.1 に記述する。

BiLSTM の第一層は単語ベクトル $\mathbf{x}^{word} \in \mathbb{R}^{d^{word}}$ と述語位置ベクトル $\mathbf{x}^{mark} \in \mathbb{R}^{d^{mark}}$ を入力として受け取る。単語ベクトルとして、既存の単語埋め込み (Word Embeddings) が利用可能である。述語位置ベクトルは、バイナリ特徴量をもと作られる。バイナリ特徴量として、述語位置 p にある単語には 1 が与えられ、それ以外の単語は 0 が与えられる。例えば、図 1 最下部では、述語位置 $p = 2$ であり、そ

の位置にある単語 bought にバイナリ特徴量 1 が与えられている。

これらを入力として受けとり、BiLSTM の最上部 (L 番目) の層まで状態ベクトル (hidden states) を計算する。最終的に、状態ベクトルの系列 $\mathbf{h}_{1:T}^{(L)} = \mathbf{h}_1^{(L)}, \dots, \mathbf{h}_T^{(L)}$ を得る。この系列 $\mathbf{h}_{1:T}^{(L)}$ を基本特徴量ベクトル $\mathbf{h}_{1:T}$ (式 4) とし、関数 f_{span} の入力 (式 5) に用いる。各ベクトル $\mathbf{h}_t \in \mathbf{h}_{1:T}$ は d^h 次元を持つ。

スパン特徴量抽出関数 f_{span}

f_{span} は、 f_{base} で計算された特徴量ベクトル $\mathbf{h}_{1:T}$ をもとに、以下のようなスパン特徴量ベクトルを計算する。

$$f_{span}(\mathbf{h}_{1:T}, s) = [\mathbf{h}_i + \mathbf{h}_j; \mathbf{h}_i - \mathbf{h}_j] . \quad (7)$$

ここでは、当該スパン $s = (i, j)$ に対して、スパン境界に位置する i 番目と j 番目のベクトルの加減算が行われている。加算 $\mathbf{h}_i + \mathbf{h}_j$ と減算 $\mathbf{h}_i - \mathbf{h}_j$ の結果として得られるベクトルを結合し、 $2d^h$ 次元を持つスパンベクトル $\mathbf{h}_s$ が得られる。

図 1 中部は、この計算過程の概略を示している。はじめに、当該スパン (3, 5) に対して、 f_{span} は 3 番目と 5 番目の基本特徴量ベクトル ($\mathbf{h}_3$ と $\mathbf{h}_5$) を受け取る。次に、これら 2 つのベクトルに対して加減算を行う。結果として得られるベクトルを結合し、ラベリング関数 f_{label} に入力として与える。

本稿で定義したスパン特徴量ベクトルの計算法 (式 7) は、構文解析分野で用いられた方法 [28], [33], [35] を参考にしている。スパンに基づくこれらの特徴量は、既存の BIO 型 SRL モデルで利用することは困難であったが、本稿の提案モデルでは容易に利用可能である。

ラベリング関数 f_{label}

f_{label} は、スパンベクトル $\mathbf{h}_s$ を入力として受けとり、各ラベル $r \in R$ に対するスコアを計算する。

$$f_{label}(\mathbf{h}_s, r) = \mathbf{W}[r] \cdot \mathbf{h}_s . \quad (8)$$

ここで、 $\mathbf{W} \in \mathbb{R}^{|R| \times 2d^h}$ は、ラベル r に対応する行ベクトルを持つパラメータ行列である。 $\mathbf{W}[r]$ は r 行目のベクトルを表す。 $\mathbf{W}[r]$ と $\mathbf{h}_s$ の内積計算の結果として、ラベル r に対するスパン $s = (i, j)$ のスコアが得られる。

図 1 上部はこの計算過程の概略を示している。 f_{label} は、 $\mathbf{h}_3$ と $\mathbf{h}_5$ の加減算の結果として得られるスパンベクトル $\mathbf{h}_s$ を受け取る。次に、 $\mathbf{h}_s$ と $\mathbf{W}[r]$ の内積を計算する。ラベル A0 のスコアは 2.1 であり、A1 のスコアは 3.7 である。同様に、すべてのスパン (S) と各ラベルに対するスコア計算により、スコア行列 (図 1 最上部) が得られる。

	CoNLL-2005			CoNLL-2012		
	Train	Dev	Test	Train	Dev	Test
文数	39.8	1.3	2.8	115.8	15.6	9.4
述語数	90.7	3.2	6.0	253.0	35.2	24.4

表 1 データセットの統計. 各数値は千単位で表している.

3.2 アンサンブル法

M 個のモデルが出力するスパンベクトルを用いたアンサンブル法を提案する.*2各モデル $m \in M$ は異なるランダム初期化 (ランダムシード) から構築されるため, 出力する値に分散が生じる. この分散を利用するため, 各モデルが出力するスパンベクトル $\mathbf{h}_s^{(m)}$ の重み付き和を利用する.

$$\mathbf{h}_s^{\text{moe}} = \mathbf{W}_s^{\text{moe}} \cdot \sum_{m=1}^M \alpha_m \mathbf{h}_s^{(m)},$$

$$f_{\text{label}}^{\text{moe}}(\mathbf{h}_s^{\text{moe}}, r) = \mathbf{W}^{\text{moe}}[r] \cdot \mathbf{h}_s^{\text{moe}}.$$

これは, 既存のアンサンブル法である mixture of experts (MoE)[26] の変種とみなせる. はじめに, 各モデル $m \in M$ のスパンベクトル $\mathbf{h}_s^{(m)}$ の重み付き和を計算する. $\mathbf{W}_s^{\text{moe}}$ はパラメータ行列であり, $\alpha_m \in \mathbb{R}$ はモデル数に対して正規化される ($\sum_{m=1}^M \alpha_m = 1$) 学習可能なパラメータである. 次に, $\mathbf{h}_s^{\text{moe}}$ を用いて, 式 8 と同様に, 各ラベル r に対してスコアを計算する. デコード法は単一モデルと同様である.

学習時は, アンサンブルモデルのパラメータ $\{\mathbf{W}_s^{\text{moe}}, \mathbf{W}^{\text{moe}}, \{\alpha_m\}_1^M\}$ のみを更新する. つまり, 各モデル m のパラメータは固定する. 目的関数として式 3 のクロスエントロピーを同様に用いる.

4. 実験

本節では, 提案モデルの性能評価実験について記述する. はじめに実験設定について述べ, 最後に結果を述べる.

4.1 データセット

CoNLL-2005/2012 のデータセットを用いた.*3 学習/開発/評価セットの分け方は既存研究と同様である. 表 1 はデータセットの統計を示している. CoNLL-2005 の学習セットは約 4 万文, CoNLL-2012 の学習セットは約 10 万文の規模である. 結果のスコア計算には CoNLL-2005 Shared Task で提供されている公式スクリプト*4を利用する.

*2 SRL のためのアンサンブル法として product of experts (PoE) が頻繁に利用されており, その有効性が示されている [6], [11], [32]. しかし, 事前実験において, 本稿のスパン型モデルでは PoE の効果が見られなかった. そのため, スパン型モデルに有効に働くアンサンブル法の設計に至った.

*3 既存研究に従い, 次の URL で提供されているバージョンの OntoNotes コーパスを CoNLL-2012 のデータセット構築に利用した: <http://cemantix.org/data/ontonotes.html>.

*4 <http://www.lsi.upc.edu/~srlconll/soft.html>

4.2 比較モデル

BIO 型の比較モデルとして, BiLSTM-CRF モデル [37] を用いる. このモデルで用いられる BiLSTM は, 本稿の提案モデルの基本特徴量抽出関数 f_{base} と同様のものを用いる. ラベルの確率計算には CRF を用いる. 本節では, このモデルを「CRF モデル (CRF)」と記述し, 提案モデルを「スパンモデル (SPAN)」と記述する.

4.3 モデルのセットアップ

単語埋め込み

単語埋め込み (Word Embeddings) は SRL モデルの性能に大きな影響を与える. したがって, モデルの性能検証には複数種類の単語埋め込みを用いることが望ましい. 本稿では 2 種類の単語埋め込みを用いる.

- 一般的な単語埋め込み: SENNA*5[4]
- 言語モデル型単語埋め込み: ELMo*6[19]

SENNA は, これまで多くの NLP タスクで利用されてきた単語埋め込みである. 一方, ELMo (Embeddings from Language Models) は, 最近提案され, 多くの NLP タスクでの有効性が報告されている. ELMo は, 言語モデルの中間表現を単語埋め込みとして用いる.

SENNA と ELMo は, 文脈依存性が異なる単語埋め込みであると言える. つまり, SENNA は文脈非依存の埋め込みであり, 入力文に関わらず各単語に対して常に同一のベクトルを割り当てる. 対照的に, ELMo は文脈依存の埋め込みであり, 入力文が異なれば同一単語に対しても異なるベクトルを割り当てる. このように, 性質の異なる 2 種類の単語埋め込みを実験に用いる. これらの埋め込みを, 学習時に更新せず, 固定のまま利用する.

モデルのハイパーパラメータ

述語位置埋め込みとして, ランダム初期化した 50 次元ベクトルを学習時に更新して用いる. 基本特徴量抽出関数 f_{base} として, 300 次元の隠れ状態を伴う BiLSTM を用いる. 既存研究 [11] に従い, BiLSTM は random orthonormal matrices[24] を初期値として用いる. その他のハイパーパラメータの詳細は, 付録 A.2 に記述する.

学習

目的関数として, クロスエントロピー (式 3) と L2 正則化を用いる.

$$\mathcal{L}_\theta = \sum_{(X,Y) \in \mathcal{D}} \ell_\theta(X,Y) + \frac{\lambda}{2} \|\theta\|^2.$$

*5 <http://ronan.collobert.com/senna/>

*6 <http://allennlp.org/elmo>

EMB	MODEL	Development			Test WSJ			Test Brown			Test ALL		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
SENNA	CRF	81.7	81.3	81.5	83.3	82.5	82.9	72.6	70.0	71.3	81.9	80.8	81.4
	SPAN	83.6	81.4	82.5	84.7	82.3	83.5	76.0	70.4	73.1	83.6	80.7	82.1
	SPAN (Ensemble)	85.6	82.6	84.1	86.6	83.6	85.1	78.2	71.8	74.8	85.5	82.0	83.7
ELMo	CRF	86.6	86.8	86.7	87.4	87.3	87.3	78.5	78.3	78.4	86.2	86.1	86.1
	SPAN	87.4	86.3	86.9	88.2	87.0	87.6	79.9	77.5	78.7	87.1	85.7	86.4
	SPAN (Ensemble)	88.0	86.9	87.4	89.2	87.9	88.5	81.0	78.4	79.6	88.1	86.6	87.4

表 2 実験結果：CoNLL-2005 データセットにおける適合率 (P), 再現率 (R), F1 値 (F1).

EMB	MODEL	Development			Test		
		P	R	F1	P	R	F1
SENNA	CRF	82.8	81.9	82.4	82.9	81.9	82.4
	SPAN	84.3	81.5	82.9	84.4	81.7	83.0
	SPAN (Ensemble)	86.0	83.0	84.5	86.1	83.3	84.7
ELMo	CRF	86.1	85.8	85.9	86.0	85.7	85.9
	SPAN	87.2	85.5	86.3	87.1	85.3	86.2
	SPAN (Ensemble)	88.6	85.7	87.1	88.5	85.5	87.0

表 3 実験結果：CoNLL-2012 データセットにおける適合率 (P), 再現率 (R), F1 値 (F1).

	CoNLL-05			CoNLL12
	WSJ	Brown	ALL	
SINGLE MODEL				
ELMo-SPAN	87.6	78.7	86.4	86.2
Peters+ 18	-	-	-	84.6
Tan+ 18	84.8	74.1	83.4	82.7
He+ 17	83.1	72.1	81.6	81.7
Zhou+ 15	82.8	69.4	81.1	81.3
FitzGerald+ 15	79.4	71.2	-	79.6
Täckström+ 15	79.9	71.3	-	79.4
Toutanova+ 08	79.7	67.8	-	-
Punyakanok+ 08	79.4	67.8	77.9	-
ENSEMBLE MODEL				
ELMo-SPAN	88.5	79.6	87.4	87.0
Tan+ 18	86.1	74.8	84.6	83.9
He+ 17	84.6	73.6	83.2	83.4
Zhou+ 15	82.8	69.4	81.1	81.3
FitzGerald+ 15	80.3	72.2	-	80.1
Toutanova+ 08	80.3	68.8	-	-
Punyakanok+ 08	79.4	67.8	77.9	-

表 4 既存モデルとの比較：評価セットにおける F1 値.

ここで、ハイパーパラメータ λ は正則化を調整する係数であり、0.0001 とする。

パラメータの最適化には Adam($\beta_1 = 0.9, \beta_2 = 0.999$)[12] を用いる。学習率は 0.001 からはじめ、50 エポックの学習後、25 エポックごとに半減させる。ミニバッチサイズは 32 とする。単語埋め込みと各 LSTM の入力ベクトルに対

し、ドロップアウト (Dropout)[27] をドロップ率 0.1 で適用する。

アンサンブルモデル (3.2 節) のインプットとして、5 つのベースモデルのスパン特徴量 $h_s^{(m)}$ を用いる。各ベースモデルは異なるランダム初期化によってそれぞれ独立に学習済みである。前述したように、アンサンブルモデル学習時は、ベースモデルのパラメータは学習しない。

学習エポック数は 100 とする。開発セットで最大 F1 値を記録したエポックにおけるパラメータを保存し、評価セットで性能の評価をする。学習には単一 GPU で 1-2 日を要した。

4.4 結果

表 2 と 3 は CoNLL-2005 と 2012 データセットにおける結果を示している。本稿では、独立に学習した 5 つのモデルの平均スコアを報告する。

ELMo を用いたアンサンブルスパンモデルが最も高い F1 値 (約 87%) を記録した。単一モデル同士の比較では、SENNa と ELMo のどちらの単語埋め込みを使った場合でも、スパンモデルが CRF モデルより一貫して高い F1 値を記録した。ELMo を用いた場合に精度差は小さくなっているが、これは精度の上限に近づいたため自然であると解釈できる。異なる単語埋め込みを用いたモデル同士の比較では、ELMo を用いたモデルの方が SENNA を用いたモデルより精度が顕著に高い。

表 4 は既存モデルとの F1 値の比較を示している。スパンモデルは、単一・アンサンブルどちらのモデルにおいても、両データセットにおいて最高精度を記録している。

EMB	MODEL	CoNLL-05		CoNLL-12	
		F1	diff	F1	diff
SENNA	CRF	87.0	-0.4	87.9	-0.6
	SPAN	86.6		87.3	
ELMo	CRF	91.2	-0.7	90.9	-0.6
	SPAN	90.5		90.3	

表 5 スパン境界予測の F1 値.

EMB	MODEL	CoNLL-05		CoNLL-12	
		F1	diff	F1	diff
SENNA	CRF	93.8	+1.5	93.6	+1.5
	SPAN	95.3		95.1	
ELMo	CRF	95.2	+0.9	94.4	+1.3
	SPAN	96.1		95.7	

表 6 意味役割ラベルの正解率.

5. 分析

スパンモデルをより詳細に理解するため、本節では解析結果の分析を行う。分析として、主に 2 つの疑問について調査し、それぞれの調査結果は以下のように要約できる。

疑問

- (a) スパン・CRF モデルのそれぞれの長所は何か？
- (b) ELMo によって、意味役割付与のどのような側面が改善されたか？

発見

- (a) CRF モデルがスパン境界同定精度が高いのに対し (5.1 節)、スパンモデルはラベル予測精度が高い (5.2 節)。
- (b) ELMo はスパン境界の同定精度向上に大きく寄与している (5.1 節)。

疑問 (a) と (b) に関して調査するため、次の 2 つの側面から解析結果を定量的に分析する。

- スパン境界 (i, j) の同定
- 意味役割ラベル r の予測

これらは、意味役割付与の部分問題として解釈できる。つまり、意味役割付与で予測するラベル付きスパン $\langle i, j, r \rangle$ は、上記 2 つの問題が合わさったものである。

本節では、はじめに、この 2 つの側面に個別に着目し、スパン・CRF モデルの性能に関する定量的分析を行う (5.1・5.2 節)。その後、スパンモデルの特徴に関して、より詳細に分析する。特に、「スパンモデル内でどのようなスパンベクトル・ラベルベクトルが学習されているか」に着目し、定性的分析を行う (5.3 節)。

5.1 スパン境界同定精度

4 節の実験で用いられた単一モデルに関して分析する。各単一モデルが出力した結果に関して、スパン境界の一致のみを評価した F1 値を表 5 に示す。正解判定基準として、もし、予測されたラベル付きスパン $\langle \hat{i}, \hat{j}, * \rangle$ が、正解のスパン $\langle i, j \rangle$ と一致していればラベルに関わらず正解とみなす。結果として、両データセットにおいて CRF モデルがス

パンモデルより高い F1 値を記録している。単語埋め込み間の比較をすると、SENNA を用いたモデルに比べ、ELMo を用いたモデルは 3 ポイント以上高い F1 値を記録している。この結果から、ELMo による意味役割付与精度向上の要因として、スパン境界同定精度向上が大きく寄与していると考えられる。

5.2 意味役割ラベル予測精度

単一モデルの出力結果に対して、意味役割ラベルのみを評価した正解率を表 6 に示す。評価対象として、スパン境界が正解と一致しているラベル付きスパンのみを扱う。

結果として、スパンモデルが CRF モデルより高い正解率を記録している。興味深い結果として、SENNA-ELMo 間で比較をすると、ラベル予測精度差があまり大きくない (約 1 ポイント) ことが挙げられる。5.1 節におけるスパン境界同定精度差が SENNA-ELMo 間で 3 ポイント以上であったのと比べると、ELMo による精度向上幅は小さいと言える。

この理由を、SRL タスクの持つ統語的・意味的性質から考える。スパン境界は、句構造の構成素 (Constituent) のそれと一致するため、統語的な側面と関係が深い。一方、意味役割ラベルはタスクの意味的性質をより反映している。例えば、統語的には「主語」であるスパンに異なる意味役割ラベルが付与されることも少なくない。つまり、タスクの定める意味役割に依拠する比重が大きくなる。これら両側面が意味役割付与には求められ、最終的な結果に影響を与える。そのうちの統語的側面の精度向上に ELMo は特に寄与していると考えられる。

意味役割ラベル別精度

各モデルの意味役割ラベルごとの得意・不得意を調査する。表 7 は、頻出の意味役割ラベルごとの F1 値を示している。ラベル A0 と A1 に関しては、CRF モデル-スパンモデル間で大きな差は見られない。ラベル A2 に関しては、スパンモデルが CRF モデルよりも一貫して高い F1 値を記録している。ラベル A3 に関しては、概ねスパンモデルの方が良い結果であるが、一部の設定 (CoNLL-2005 における ELMo を用いた設定) においてはほぼ同等の性能となっている。付加詞ラベル (AM-*) に関しては、一貫していない結果となった。

Label	CoNLL-2005				CoNLL-2012			
	SENNA		ELMo		SENNA		ELMo	
	CRF	SPAN	CRF	SPAN	CRF	SPAN	CRF	SPAN
A0	89.9	90.2	93.0	93.2	89.9	90.0	92.5	92.5
A1	83.2	83.8	89.1	89.2	84.7	85.1	88.7	89.0
A2	70.9	73.1	80.0	81.2	78.6	79.4	83.2	84.2
A3	64.4	71.2	78.8	78.5	61.9	62.9	69.0	70.7
AM-ADV	59.3	61.9	68.1	67.0	63.2	63.7	67.5	67.0
AM-DIR	43.2	47.3	56.6	54.5	54.1	52.0	61.1	59.7
AM-LOC	58.2	60.5	68.1	68.3	65.8	65.0	72.0	72.0
AM-MNR	61.4	61.3	66.5	67.7	64.4	65.7	70.5	71.1
AM-PNC	57.3	60.2	68.8	67.7	18.5	13.7	20.2	16.1
AM-TMP	81.8	82.7	86.1	86.0	82.2	82.3	86.1	86.2
Overall	81.5	82.5	86.7	86.9	82.4	82.9	85.9	86.3

表 7 意味役割ラベルごとの精度 : CoNLL-2005 と 2012 の開発セットにおける F1 値.

pred / gold	A0	A1	A2	A3	ADV	DIR	LOC	MNR	PNC	TMP
A0	0	48	7	13	3	0	0	3	0	0
A1	69	0	43	20	0	42	6	6	0	0
A2	10	29	0	20	6	57	43	22	40	11
A3	5	2	7	0	0	0	0	3	20	0
ADV	0	0	0	0	0	0	6	35	40	55
DIR	0	0	14	6	0	0	0	0	0	0
LOC	15	10	4	0	6	0	0	19	0	0
MNR	0	2	9	33	31	0	12	0	0	33
PNC	0	2	7	6	0	0	6	0	0	0
TMP	0	2	4	0	51	0	25	9	0	0

図 2 スパン型モデルのラベル予測混同行列. 各セルの数値は正解ラベルに対する予測ラベルの割合を示している.

意味役割ラベル混同行列

モデルのラベル予測誤り傾向分析の一つとして, ELMo を用いたスパンモデルのラベル予測混同行列を図 2 に示す.*7 先行研究 [11] に従い, 正解のスパン境界と一致しているラベル付きスパンのみを評価対象とする.

スパンモデルが最も混同しているのは, A0(動作主) と A1(被動作主) のラベルである. これら 2 つのラベルの予測が困難な理由の 1 つに「能格動詞」の影響がある. 次の例を考える.

People start their own business ...

[A0]

.. Congress has started to jump on ...

[A1]

どちらの文においても, 能格動詞「start」が述語である. 注目すべき点は, 統語的には「主語」にあたる位置のスパン (People と Congress) に A0 と A1 という異なるラベルが付与される点である. この違いは, 主語の位置にあるスパン

*7 SENNA を用いたスパンモデルでも同様の傾向を観測したため, ELMo を用いたモデルの混同行列のみを記載する.

“... toy makers to move [across the border] .”

GOLD:A2

PRED:AM-DIR

「across the border」の最近傍スパンとその正解ラベル

1	AM-DIR	across the Hudson
2	AM-DIR	outside their traditional tony circle
3	AM-DIR	across the floor
4	AM-DIR	through this congress
5	A2	off their foundations
6	AM-DIR	off its foundation
7	AM-DIR	off the center field wall
8	A3	out of bed
9	A2	through cottage rooftops
10	AM-DIR	through San Francisco

表 8 スパン選択モデルの予測誤り例. ラベル予測を誤ったスパンと, その誤りスパンの 10 最近傍スパンと正解ラベル.

の意味的性質 (有生性など) に起因するものである. このような事例への意味役割ラベル付与は困難を伴う. 「start」の他に「appear」なども能格動詞であり, そのような事例において A0 と A1 の混同が多く観測された.

他に混同が多かったラベルとして, コアラベル A2 と場所に関連した付加詞ラベル (AM-DIR や AM-LOC) が挙げられる. 先行研究 [11] でも指摘されているように, 多くの動詞における A2 は方向性や場所に関連した意味的關係を表す場合が多い. つまり, どちらも場所に関連しているため, それらを識別することが困難な場合がある. 提案モデルによって SRL 全体の精度は改善したが, このような混同はまだ観測された.

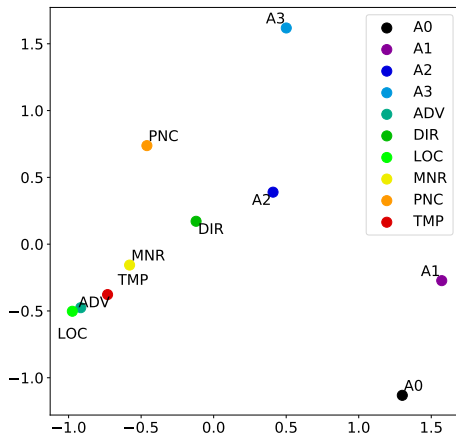


図3 ラベルベクトルの分布.

5.3 スパン選択モデルの定性的分析

スパンベクトルに関する分析

スパンモデルのスパンベクトル (2 節の式 7) とラベル予測との関係に関して定性的に分析する. 分析対象として, モデルが予測した開発データ内のスパン $\hat{s}^{(\text{dev})}$ を用いる.

表 8 上部は, 述語 move に対する項である across the border の正解ラベル (A2) と予測ラベル (AM-DIR) を示している. 表 8 下部は, across the border のスパンベクトルの最近傍にあたる 10 個のスパンを正解ラベルとともに示している. 最近傍を求めるための「近さ」として, 内積値を利用した.

$$d = \mathbf{h}_s^{(\text{dev})} \cdot \mathbf{h}_s^{(\text{train})}$$

$\mathbf{h}_s^{(\text{dev})}$ は $\hat{s}^{(\text{dev})}$ に対するベクトルである. $\mathbf{h}_s^{(\text{train})}$ は, 学習セット内の各正解スパン $s^{(\text{train})}$ に対するベクトルである. 内積値 d の大きい順に学習データ内のスパンを並べ替え, 上位 10 個のスパンを最近傍スパンとした. 表 8 が示すように, モデルが誤ったラベルを予測している場合, 異なる正解ラベルを持つスパンが最近傍を占める傾向が見られた. 対照的に, 正解ラベルが予測されたスパンの最近傍は, 同一ラベルをもつスパンで占められる傾向があった.

ラベルベクトルに関する分析

スパンモデルのラベリング関数では, 各ラベルに対応するラベルベクトル表現を学習する (2 節の式 8). 本節では, 学習されたラベルベクトルがどのように分布しているか定性的に分析する.

図 3 は学習されたラベルベクトル表現の分布を表している. これを見ると, 付加詞ラベルがかたまって位置していることがわかる. これは, 識別が困難であることを示唆している. また, コアラベル A2 も AM-DIR などの場所を表すラベルの近くに位置している. 5.2 節における A2 と AM-DIR の予測混同と一貫したものとなっている. さらなる識別能力改善のため, 互いのラベルベクトル表現を遠ざけるような学習手法 [16], [36] の効果が期待できる.

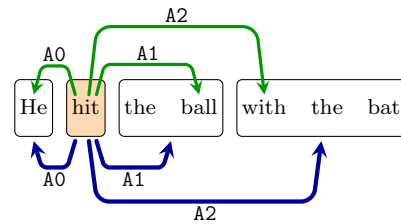


図4 依存構造型 SRL(上部) とスパン型 SRL(下部) の例.

6. 関連研究

6.1 意味役割付与とタスク

大規模コーパスを用いた意味役割付与 (SRL) は, Gildea ら [7] の先駆的研究によって注目を集めることとなった. 現在の SRL 研究は, 以下の 2 つの流派によって主に進められている.

- PropBank スタイル SRL [18]
- FrameNet スタイル SRL [1], [5]

PropBank スタイル SRL に基づいて, いくつかの Shared Task が提唱された.

- The CoNLL-2004/2005 shared tasks [2], [3] :
スパン型 (Span-based/Constituency-based)SRL
- The CoNLL-2008/2009 shared tasks [10], [29] :
依存構造型 (Dependency-based)SRL

スパンを解析単位としたスパン型 SRL と, 単語を解析単位とした依存構造型 SRL に大別される. 図 4 は, それらの例を示している. 依存構造型 (図 4 上部) では, 述語 hit の A2 の項として with を出力できれば正解となる. 一方, スパン型 (図 4 下部) では, with the bat を出力しなければならない. 本稿の研究対象はスパン型 SRL である.

6.2 BIO 型 SRL モデル

SRL は BIO 系列ラベリング問題として解くことができる [17], [20]. 最先端のニューラル SRL モデルも, この枠組みに基づいている. 先駆的なニューラル SRL モデルは Collobert ら [4] によって提案された. 彼らのモデルは, Convolutional Neural Networks (CNN) と CRF を用いて解析を行う. その後, Zhou ら [37] や He ら [37] は, CNN の代わりに多層 BiLSTM を利用し, 構文情報を用いずに当時の SRL 最高精度を記録した. 最近では, Self-Attention 機構を用いたニューラルネットを用いたモデルが, CoNLL-2005 のデータセットにおいて最高精度を達成している [32]. また, Peters ら [19] は, 言語モデルに基づいた単語埋め込み ELMo を He ら [11] のモデルに適用し, CoNLL-2012 のデータセットにおいて最高精度を達成している.

6.3 スパン型 SRL モデル

もう一つの主流なアプローチは、ラベル付きスパンのモデリングに基づくものである [14], [34]. このアプローチでは、各スパンに対し適切な意味役割ラベルを求めるモデルが提案されている。ラベルを選択する際に構造的制約を導入するため、整数線形計画法 (Integer Linear Programming) [22] や動的計画法 (Dynamic Programming) [6], [31] を用いた手法が提案されている。

既存のスパン型モデルと本稿の提案モデルとの大きな違いは、以下の点である。

既存モデル $P(r | (i, j))$ をモデル化する

提案モデル $P((i, j) | r)$ をモデル化する

既存モデルは、各スパン (i, j) に対するラベル $r \in \mathcal{R}$ の確率分布をモデル化する。一方、提案モデルは、各ラベル r に対するスパン $(i, j) \in \mathcal{S}$ の確率分布をモデル化する。本提案モデルの方が好ましいと考える理由は、「正解スパンの個数」にある。ほとんどの場合、「1 つのラベルに対し 1 つ以下のスパン」が正解となる。2.4 節で説明したように、AM-TMP のような付加詞ラベルを付与すべきスパンは複数現れうるが、あくまでもそれは少数である。既存モデルでは、NULL ラベルを設定し、ラベルを付与しないスパンに対しては、NULL ラベルの確率 (あるいはスコア) が大きくなるように学習する。しかし、スパン候補集合 $\mathcal{S}$ に含まれるほとんどのスパンに対する正解ラベルが NULL であるといった偏りが生じるため、学習する上であまり好ましくない。本稿の提案モデルでは、各ラベルに対して正解スパンが定義されるため、NULL スパン数は最大でもラベル数 $|\mathcal{R}|$ に抑えられる。それと同時に、AM-TMP のように複数の正解スパンがある場合でも、それらすべての正解スパンの確率が高くなるように学習するため、問題とならない。

その他の関連したスパン型 SRL モデルとして、FrameNet スタイル SRL における Swayamdipta ら [30] のモデルが挙げられる。彼らは、Segmental RNN [13] と semi-Markov CRF [23] を組み合わせたモデルを提案している。このモデルは、入力文の可能なラベル付きスパンすべての組み合わせに対する確率分布をモデル化している。しかし、計算量を増やした大域的モデリングにも関わらず、よりシンプルな既存の FrameNet SRL モデルに精度が届かない結果となっている。

6.4 その他の NLP タスクにおけるスパン型モデル

句構造構文解析 (Constituency Parsing) において、ニューラルネットを用いたいくつかのスパン型モデルが提案されている。Wang ら [35] は、LSTM を用いたスパンベクトル表現 (LSTM-Minus) を提案し、構文解析に利用した。Stern ら [28] も、LSTM に基づいたスパンベクトルを用い

て、句構造解析において当時の最高精度を達成した。共参照解析 (Coreference Resolution) において、Lee ら [15] は文章内すべてのスパンを考慮したモデルを提案している。本稿のモデルは、これらのモデルのスパンベクトル計算法などの点を参考にしている。

7. おわりに

本稿では、シンプルかつ効果的なスパン型モデルを提案した。SRL をスパン選択問題として扱い、各ラベルに対して適切なスパンを選択することにより解析を行う。性能評価実験を行い、強力な SRL モデルである BiLSTM-CRF を上回る結果を得た。ベースとなる提案モデルに加え、言語モデルに基づく単語埋め込みとスパンベクトルアンサンブル法を適用することにより、CoNLL-2005/2012 の両データセットにおいて最高精度を達成した。提案モデルに関する定量的・定性的分析の結果、いくつかの知見が得られた。1 つは、スパン選択モデルが、CRF 型モデルと比べてラベル予測精度が良いことがわかった。また、言語モデルに基づく単語埋め込みを用いることにより、スパン境界同定精度は向上することもわかった。

これからの研究課題として、スパン選択モデルによって学習されるスパンベクトルの定量的評価が挙げられる。スパンベクトルが、スパンのどのような特徴を反映しているか、また、他の NLP タスクでの有用性などの観点から調査を行うことは、興味深い知見につながる見込みがある。その他の方向性として、ラベル予測の混同を抑えるため、動詞フレーム知識などの外部資源の利用が考えられる。

謝辞 東北大学の乾健太郎氏、鈴木潤氏、松林優一郎氏にさまざまなご教示を頂いたことを深謝する。

参考文献

- [1] Baker, C. F., Fillmore, C. J. and Lowe, J. B.: The Berkeley FrameNet Project, *Proceedings of ACL-COLING*, pp. 86–90 (1998).
- [2] Carreras, X. and Marquez, L.: Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling, *Proceedings of CoNLL*, pp. 89–97 (2004).
- [3] Carreras, X. and Marquez, L.: Introduction to the CoNLL-2005 shared task: Semantic role labeling, *Proceedings of CoNLL*, pp. 152–164 (2005).
- [4] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.: Natural Language Processing (Almost) from Scratch, *Journal of Machine Learning Research* (2011).
- [5] Das, D., Chen, D., Martins, A. F., Schneider, N. and Smith, N. A.: Frame-semantic parsing, *Computational linguistics*, Vol. 40, No. 1, pp. 9–56 (2014).
- [6] FitzGerald, N., Täckström, O., Ganchev, K. and Das, D.: Semantic Role Labeling with Neural Network Factors, *Proceedings of EMNLP*, pp. 960–970 (2015).
- [7] Gildea, D. and Jurafsky, D.: Automatic labeling of semantic roles, *Computational linguistics*, Vol. 28, No. 3, pp. 245–288 (2002).
- [8] Graves, A., Jaitly, N. and Mohamed, A.-r.: Hybrid speech recognition with deep bidirectional LSTM, *Proceedings of Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop* (2013).
- [9] Graves, A., Fernández, S. and Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition, *Proceedings of International Conference on Artificial Neural Networks*, pp. 799–804 (2005).
- [10] Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Marquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N. and Zhang, Y.: The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages, *Proceedings of CoNLL*, pp. 1–18 (2009).
- [11] He, L., Lee, K., Lewis, M. and Zettlemoyer, L.: Deep semantic role labeling: What works and what’s next, *Proceedings of ACL*, pp. 473–483 (2017).
- [12] Kingma, D. and Ba, J.: Adam: A Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980* (2014).
- [13] Kong, L., Dyer, C. and Smith, N. A.: Segmental recurrent neural networks, *Proceedings of ICLR* (2016).
- [14] Koomen, P., Punyakanok, V., Roth, D. and Yih, W.-t.: Generalized inference with multiple semantic role labeling systems, *Proceedings of CoNLL*, pp. 181–184 (2005).
- [15] Lee, K., He, L., Lewis, M. and Zettlemoyer, L.: End-to-end Neural Coreference Resolution, *Proceedings of EMNLP*, pp. 188–197 (2017).
- [16] Luo, L., Wang, X., Hu, S., Wang, C., Tang, Y. and Chen, L.: Close yet distinctive domain adaptation, *arXiv preprint arXiv:1704.04235* (2017).
- [17] Marquez, L., Comas, P., Giménez, J. and Catala, N.: Semantic role labeling as sequential tagging, *Proceedings of CoNLL*, pp. 193–196 (2005).
- [18] Palmer, M., Gildea, D. and Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles, *Computational linguistics*, Vol. 31, No. 1, pp. 71–106 (2005).
- [19] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L.: Deep contextualized word representations, *arXiv preprint arXiv:1802.05365* (2018).
- [20] Pradhan, S., Hacioglu, K., Ward, W., Martin, J. H. and Jurafsky, D.: Semantic role chunking combining complementary syntactic views, *Proceedings of CoNLL*, pp. 217–220 (2005).
- [21] Pradhan, S., Moschitti, A., Xue, N., Uryupina, O. and Zhang, Y.: CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes, *Proceedings of EMNLP-CoNLL*, pp. 1–40 (2012).
- [22] Punyakanok, V., Roth, D. and Yih, W.-t.: The importance of syntactic parsing and inference in semantic role labeling, *Computational Linguistics*, Vol. 34, No. 2, pp. 257–287 (2008).
- [23] Sarawagi, S. and Cohen, W. W.: Semi-markov conditional random fields for information extraction, *Proceedings of NIPS*, pp. 1185–1192 (2004).
- [24] Saxe, A. M., McClelland, J. L. and Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, *arXiv preprint arXiv:1312.6120* (2013).
- [25] Schuster, M. and Paliwal, K. K.: Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing*, pp. 2673–2681 (1997).
- [26] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G. and Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, *arXiv preprint arXiv:1701.06538* (2017).
- [27] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958 (2014).
- [28] Stern, M., Andreas, J. and Klein, D.: A Minimal Span-Based Neural Constituency Parser, *Proceedings of ACL*, pp. 818–827 (2017).
- [29] Surdeanu, M., Johansson, R., Meyers, A., Marquez, L. and Nivre, J.: The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies, *Proceedings of CoNLL*, pp. 159–177 (2008).
- [30] Swayamdipta, S., Thomson, S., Dyer, C. and Smith, N. A.: Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold, *arXiv preprint arXiv:1706.09528* (2017).
- [31] Täckström, O., Ganchev, K. and Das, D.: Efficient Inference and Structured Learning for Semantic Role Labeling, *Transactions of ACL*, Vol. 3, pp. 29–41 (2015).
- [32] Tan, Z., Wang, M., Xie, J., Chen, Y. and Shi, X.: Deep Semantic Role Labeling with Self-Attention, *Proceedings of AAAI* (2018).
- [33] Teranishi, H., Shindo, H. and Matsumoto, Y.: Coordination Boundary Identification with Similarity and Replaceability, *Proceedings of IJCNLP*, pp. 264–272 (2017).
- [34] Toutanova, K., Haghighi, A. and Manning, C. D.: Joint learning improves semantic role labeling, *Proceedings of ACL*, pp. 589–596 (2005).
- [35] Wang, W. and Chang, B.: Graph-based dependency parsing with bidirectional lstm, *Proceedings of ACL*, pp. 2306–2315 (2016).
- [36] Wen, Y., Zhang, K., Li, Z. and Qiao, Y.: A discriminative feature learning approach for deep face recognition, *Proceedings of ECCV*, pp. 499–515 (2016).
- [37] Zhou, J. and Xu, W.: End-to-end learning of semantic role labeling using recurrent neural networks, *Proceedings of ACL-IJCNLP*, pp. 1127–1137 (2015).

付 録

A.1 BiLSTMの詳細

3 節で関数 f_{base} として用いた BiLSTM について説明する。

既存モデル [11], [37] に従い、多層に積み上げた LSTM を用いる。全体のネットワークは L 層からなり、奇数層では左から右に、偶数層では右から左に入力系列を処理する。

$$\mathbf{h}_t^{(\ell)} = \begin{cases} \text{LSTM}^{(\ell)}(\mathbf{h}_t^{(\ell-1)}, \mathbf{h}_{t-1}^{(\ell)}) & (\ell = \text{odd}) \\ \text{LSTM}^{(\ell)}(\mathbf{h}_t^{(\ell-1)}, \mathbf{h}_{t+1}^{(\ell)}) & (\ell = \text{even}) \end{cases}$$

ここで、奇数番目の層では、各時刻 t において、 $\ell-1$ 層目の LSTM の特徴量ベクトル $\mathbf{h}_t^{(\ell-1)}$ と時刻 $t-1$ での特徴量ベクトル $\mathbf{h}_{t-1}^{(\ell)}$ を入力とし、 $\mathbf{h}_t^{(\ell)}$ を計算する。同様に、偶数番目の層では、右から左に伝搬するため、 $\mathbf{h}_{t-1}^{(\ell)}$ の代わりに $\mathbf{h}_{t+1}^{(\ell)}$ が入力に用いられる。

第 1 層目 ($\ell = 1$) の入力 $\mathbf{h}_t^{(0)}$ として、入力文の各単語 $w_{1:T} = w_1, \dots, w_T$ に対応する特徴量ベクトル $\mathbf{h}_1^{(0)}, \dots, \mathbf{h}_T^{(0)}$ を受け取る。各特徴量ベクトル $\mathbf{h}_t^{(0)}$ は以下のように計算される。

$$\mathbf{h}_t^{(0)} = \mathbf{x}_t^{\text{word}} \oplus \mathbf{x}_t^{\text{mark}}$$

ここで、 $\mathbf{x}_t^{\text{word}} \in \mathbb{R}^{d_{\text{word}} \times |\mathcal{V}|}$ は単語 x_t の分散表現を表す。 $\mathbf{x}_t^{\text{mark}} \in \mathbb{R}^{d_{\text{mark}} \times 2}$ は、単語 x_t が述語位置 p にあるか ($t = p$) 否か ($t \neq p$) の 2 値インデックスに対応する分散表現を表している。これら 2 種類の分散表現を結合 ($\oplus$) し、ベクトル $\mathbf{h}_t^{(0)} \in \mathbb{R}^{d_{\text{word}} + d_{\text{mark}}}$ が得られる。

A.2 実験に用いたハイパーパラメータ

Name	Value
Word Embedding	50-dimensional SENNA
Mark Embedding	1024-dimensional ELMo
LSTM Layers	50-dimensional vector
LSTM Hidden Units d^h	{4, 6, 8}
Mini-batch Size	300 dimensions
Optimization	32
L2 Reg. λ	Adam
Dropout Ratio	0.0001
	0.1

表 A.1 Hyper-parameters used in the experiments.

4 章で用いたハイパーパラメータを表 A.1 に示す。なお、Adam のハイパーパラメータ β_1 と β_2 は文献 [12] で推奨されている 0.9 と 0.999 にそれぞれ設定している。

A.3 デコードに用いたアルゴリズム

2.4 節で用いた探索アルゴリズムの擬似コードを記す。

Algorithm 1 Span-Consistent Greedy Search

```

1: Input: Score Matrix  $\mathbf{M} \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{S}|}$ ,
2:     Predicate Index  $p$ 
3:     Sentence Length  $T$ 
4:     Core Label Set  $\mathcal{R}^{(\text{core})}$ 
5: spans  $\leftarrow \phi$ 
6: used_cores  $\leftarrow \phi$ 
7:  $\mathcal{U} \leftarrow \{(i, j, r, \text{score}) \in \text{flatten}(\mathbf{M})\}$ 
8:  $\mathcal{U} \leftarrow \text{filter}(\mathcal{U}, p)$ 
9: for  $(i, j, r, \text{score}) \in \text{sort}(\mathcal{U})$  do
10:   if  $r \notin \text{used\_cores}$  and
11:     is_overlap( $(i, j)$ , spans) is False then
12:     spans  $\leftarrow \text{spans} \cup \{(i, j, r)\}$ 
13:     if  $r \in \mathcal{R}^{(\text{core})}$  then
14:       used_cores  $\leftarrow \text{used\_cores} \cup \{r\}$ 
15: return spans

```

引数として、4 つの入力を受け取る (1-4 行目)。 $\mathbf{M}$ は、スパンのスコアを表す行列 (3 節の図 1 最上部) である。 p は述語位置、 T は入力文 $w_{1:T}$ の長さ (単語数)、 $\mathcal{R}^{(\text{core})}$ はコアラベル集合をそれぞれ表す。5 行目で、予測スパンを格納する変数 spans を初期化している。6 行目で、「コアラベルを 2 つ以上選ばない」制約のための変数を初期化している。7 行目で、後続の処理がしやすいように、スコア行列 $\mathbf{M}$ を一列のリストに変換している。それぞれの要素は、スパン・ラベル・スコア (i, j, r, score) からなる。

8 行目で、 $\mathcal{U}$ の中から、出力として不適切なスパンを事前に取り除く。取り除かれる要素は以下のいずれかに該当するものである：(i) 述語スパン $(p, p, *)$ 、(ii) 述語位置と重なるスパン、(iii) 述語スパンよりもスコアの低いスパン。(i) に関しては、述語スパンは正解の項になりえないため取り除く。(ii) に関しては、述語位置と重なるスパン $\{(i, j) \mid i \leq p \leq j\}$ はすべて取り除く。これも、正解の項になりえないためである。(iii) に関しては、述語スパン (p, p, r, score) よりもスコアの低いスパン $(*, *, r, \text{score})$ を取り除く。これは、「NULL スパンとして定義した述語スパン (2 節) よりもスコアが低い」ということは、「項として出力するのに不適切である」と判断できるためである。

メインの処理は 9 行目から始まる。関数 sort によって、スコア (score) の高い順に並び替えられた要素 (i, j, r, score) を処理する。10-11 行目では、出力スパンと認定するための制約を課している。10 行目の $r \notin \text{used_cores}$ は、制約「コアラベルは 2 つ以上選ばない」を表している。また、付加詞ラベルに対しては、個数制約を課さないことによって、複数スパンの選択を暗に許容している。11 行目は、制約「選択済みスパンとオーバーラップするスパンは選ばない」に関して判断を下している。関数 is_overlap は、当該

スパン (i, j) と選択済みスパン集合 `spans` を引数とし、当該スパンが選択済みスパンのいずれかとオーバーラップしているか否かを表す真偽値 (True/False) を返す。両制約を守っていれば、12 行目以降の処理に進む。

12 行目で、両制約に違反しないスパンを `spans` に登録する。13-14 行目で、スパンのラベル r がコアラベル $\mathcal{R}^{(\text{core})}$ なら、選択済みコアラベルとして `used_cores` に登録される。15 行目で、選択済みのスパン集合 `spans` を最終結果として返す。