

BERT を用いた Table-Filling による固有表現抽出と関係抽出

Youmi Ma 平岡達也 岡崎直観
東京工業大学

{youmi.ma,tatsuya.hiraoka}@nlp.c.titech.ac.jp
okazaki@c.titech.ac.jp

1 はじめに

固有表現およびその組である関係の抽出は、コンピュータによる情報抽出や知識獲得において重要な課題である [1, 2]. 固有表現と関係を同時に抽出する手法のひとつとして、Table-Filling を用いたものがある [3]. Table-Filling を用いた手法では、図 1 のように、表の対角要素に固有表現のラベルを、非対角要素に関係のラベルを配置することにして、二つの抽出タスクをひとつの表を埋める問題に帰着させる. しかし、Table-Filling を用いた手法で高い精度を達成するためには、固有表現や関係の分類に用いる特徴量や、表を埋める順序 (デコーディング順序) を緻密に設計する必要がある [3, 4, 5, 6].

近年、事前学習済み言語モデルを用いて固有表現と関係を同時に推定するモデルが提案されている [4, 7, 8]. 本稿では、文脈を考慮した単語埋め込みを用いることで、特徴量選択とデコーディング順序の問題を同時に解決する新たな Table-Filling に基づく固有表現抽出・関係抽出のモデルを提案する. 提案手法では特徴量選択の問題を解決するために、事前学習済みの BERT モデル [9] で入力文をエンコードし、文脈を考慮した単語埋め込みを得る. 単語埋め込みをスパン単位でプーリングした値を固有表現の特徴量とし、表の対角要素を埋める. 固有表現の特徴量で表の対角要素を埋め、これをもとに関係を表す非対角要素を一挙に埋めることで、デコーディングにおいて順序の考慮を不要とする.

CoNLL04 [10] と ACE05¹⁾ データセットを用いた実験を通して、提案手法はシンプルであるが、最高性能に匹敵することを確認した. また、提案手法においてデコーディング順序を考慮したモデル設計とした場合でも、デコーディング順序が性能に大きな影響を与えないことを確認した. さらに実験と分析を通して、固有表現抽出器を同時に学習することで埋

1) <https://catalog.ldc.upenn.edu/LDC2006T06>

		Johanson		lives		London	
		Smith		in			
		1	2	3	4	5	j
Johanson	1	1 B-PER	6 ⊥	6 ⊥	6 ⊥	6 LiveIn	→
Smith	2		2 L-PER	6 ⊥	6 ⊥	6 LiveIn	→
lives	3			3 0	6 ⊥	6 ⊥	
in	4				4 0	6 ⊥	
London	5					5 U-LOC	
		i					

図 1 Table-Filling の概要. セル内の数字は表を埋める (デコーディング) 順序を示し, ⊥ は対応する関係がないことを示す.

め込み表現の学習が効果的に行われ、関係抽出器の性能向上に繋がることを確認した²⁾.

2 提案手法

本研究の目的は、長さ n の単語列 w_1, w_2, \dots, w_n が与えられた時に、関係を $(a_0 \langle t_0 \rangle, \text{relation}, a_1 \langle t_1 \rangle)$ の形で抽出することである. ただし、 t_0, t_1 はそれぞれ 1 つ以上の w_n から成る固有表現 a_0, a_1 に対応するラベルを示す. また、 \mathcal{E} を固有表現のラベルの集合、 \mathcal{R} を関係のラベルの集合とする.

提案手法では図 1 のように、表を用いて固有表現と関係のラベル空間を表現する. 具体的には、 $n \times n$ の上三角行列 Y を定義する. 対角成分 $Y_{i,i} \in \mathcal{E}$ は i 番目の単語に対応する固有表現のラベルであり、非対角成分 $Y_{i,j} \in \mathcal{R}$ は i 番目の単語から j 番目の単語への関係のラベルである. Zhang ら [6] による研究と同様に、行列の下半分は使用

2) 本稿で紹介されたモデルは、https://github.com/YoumiMa/Enhanced_TF にて公開している.

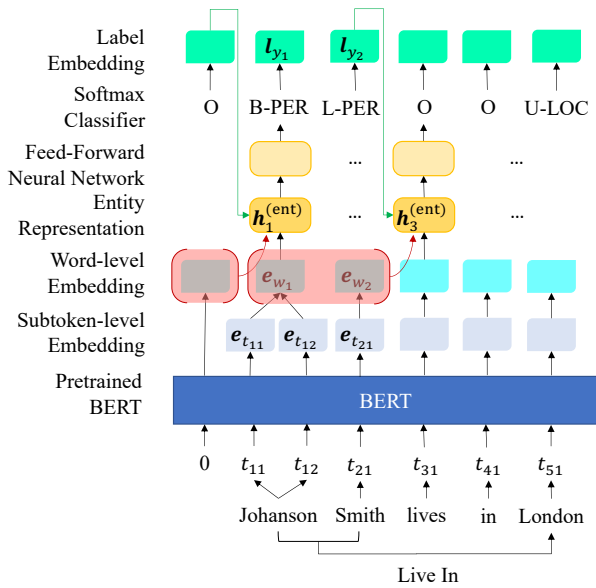


図2 固有表現抽出器の概要.

しない。また、固有表現が複数の単語にまたがる場合は、固有表現を構成する全ての単語の要素に対して関係ラベルを付与する。例えば図1では、関係 $\overrightarrow{\text{LIVEIN}}$ は「Johanson」に対応する $Y_{1,5}$ と「Smith」に対応する $Y_{2,5}$ の両方に付与される。

提案手法は固有表現抽出器と関係抽出器を事前学習済みのBERTエンコーダの出力層に積み重ねる。図2と図3のように、各単語埋め込みをサブワード埋め込みからプーリングして計算し (§2.1)、固有表現抽出 (§2.2) と関係抽出 (§2.3) を行う。

2.1 単語埋め込み

BERTはWordPieceを用いて単語をサブワード化するため、単語単位のタスクである固有表現抽出を解くためには、サブワード埋め込みを単語単位に変換する必要がある。本研究では、単語を構成する全てのサブワード埋め込みの最大プーリングによって、単語埋め込みを計算する [9]。

$$e_{w_i} = f(e_{t_{i,1}}, e_{t_{i,2}}, \dots, e_{t_{i,s}}). \quad (1)$$

単語 w_i はサブワード $t_{i,1}, t_{i,2}, \dots, t_{i,s}$ から構成され、 e_{w_i} と $e_{t_{i,k}}$ はそれぞれ単語 w_i とサブワード $t_{i,k}$ の埋め込みを示し、 $f(\cdot)$ は最大プーリング関数である。

2.2 固有表現抽出

固有表現のラベルをBILOU方式 [11] で表現することで、固有表現抽出を系列ラベリング問題として扱う。提案手法では、Zhangら [6] に倣い、固有表現

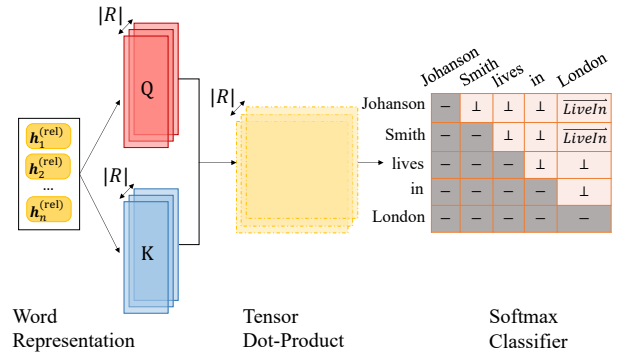


図3 関係抽出器の概要.

埋め込みをスパン単位で取り扱う。

具体的に、位置 i の単語 w_i に対し、固有表現の予測で用いる埋め込み $h_i^{(ent)}$ は以下三つのベクトルの連結 (\oplus) である。

$$h_i^{(ent)} = e_{w_i} \oplus l_{y_{i-1}} \oplus f(e_{w_{\text{first}(i-1)}}, \dots, e_{w_{i-1}}). \quad (2)$$

ただし、 $l_{y_{i-1}}$ は直前の位置 $i-1$ の単語 w_{i-1} の固有表現ラベルの埋め込み、 $f(e_{w_{\text{first}(i-1)}}, \dots, e_{w_{i-1}})$ は直前のスパンの埋め込み (該当する全ての単語埋め込みの最大プーリング) を示す。また、 $\text{first}(i)$ は単語 w_i を含むスパンの先頭位置を示す³⁾。なお、初期位置 $i=1$ については、直前の位置の固有表現ラベル y_0 にはスパンでないことを示すラベルである Outside (O) を用い、 e_{w_0} には零ベクトルを用いる。

単語 w_i の固有表現ラベルを予測するために用いる確率分布 \hat{y}_i は、全結合層とソフトマックス関数 σ を用いて計算する。

$$\hat{y}_i = \sigma(W^{(ent)} h_i^{(ent)} + b^{(ent)}). \quad (3)$$

\hat{y}_i から確率が最大となるラベルを選ぶことで、行列の対応する成分 $Y_{i,i}$ ($= y_i$) を埋める。提案手法は、行列の対角成分 $Y_{i,i}$ を $i=1$ から $i=n$ まで埋めることによって、固有表現抽出を行う。

2.3 関係抽出

関係抽出は §2.2 の予測結果に基づいて行う。関係抽出器は、行列の対角成分に埋められた固有表現のラベル情報を用いて、非対角成分を同時に予測する。各単語対の関係ラベル空間における確率分布は、内積を用いて計算する⁴⁾。

単語 w_i に対し、関係の予測で用いる埋め込み $h_i^{(rel)}$ は、固有表現のスパンの埋め込み z_i と固有表

3) 例えば、図2で示した文を処理する際に、「John Smith」は2単語から構成されるスパンが固有表現として識別されるため、 $\text{first}(1) = \text{first}(2) = 1$ である。

4) Deep Bi-affine Attention [12, 13] を用いた計算と比較した結果、内積だけで十分であることを事前の実験からわかった。

現のラベルの埋め込み l_{y_i} の連結 (\oplus) である。さらにスパンの埋め込み z_i は、単語 w_i に対応するスパン範囲の全ての単語埋め込みから計算する。

$$h_i^{(\text{rel})} = z_i \oplus l_{y_i}, \quad (4)$$

$$z_i = f(e_{w_{\text{first}(i)}}, \dots, e_{w_i}, \dots, e_{w_{\text{last}(i)}}). \quad (5)$$

ここで $\text{last}(i)$ は §2.2 で定義された $\text{first}(i)$ と同様に、単語 w_i を含む固有表現のスパンの末尾位置を示す。つまり、 z_i は $\text{first}(i)$ から $\text{last}(i)$ までの単語の埋め込みの最大プーリングである。

関係の予測を行うために、 $h_i^{(\text{rel})}$ に対して、行列 $W_r^{(q)}, W_r^{(k)} \in \mathbb{R}^{d_{\text{att}} \times d_{\text{rel}}}$ とバイアス $b_r^{(q)}, b_r^{(k)} \in \mathbb{R}^{d_{\text{att}}}$ を用いた線形変換を行う。

$$q_{i,r} = W_r^{(q)} h_i^{(\text{rel})} + b_r^{(q)}, \quad (6)$$

$$k_{i,r} = W_r^{(k)} h_i^{(\text{rel})} + b_r^{(k)}. \quad (7)$$

ただし、 d_{rel} は $h_i^{(\text{rel})}$ の次元数、 d_{att} は変換後の次元数である。全ての $r \in \mathcal{R}$ と $i \in \{1, \dots, n\}$ を集めたテンソルを $Q \in \mathbb{R}^{n \times |\mathcal{R}| \times d_{\text{att}}}$ と $K \in \mathbb{R}^{n \times |\mathcal{R}| \times d_{\text{att}}}$ とする。

$$Q_{i,r,:} = q_{i,r} = W_r^{(q)} h_i^{(\text{rel})} + b_r^{(q)}, \quad (8)$$

$$K_{i,r,:} = k_{i,r} = W_r^{(k)} h_i^{(\text{rel})} + b_r^{(k)}. \quad (9)$$

各単語対 $(w_i, w_j) (1 \leq i < j \leq n)$ に対する関係空間 \mathcal{R} での確率分布は、内積とソフトマックス関数により計算する⁵⁾。

$$\hat{y}_{i,j} = \sigma(\text{QK}^T)_{i,j,:}. \quad (10)$$

$\hat{y}_{i,j}$ から確率が最大となるラベルを選び、行列で対応する要素 $Y_{i,j}$ に埋めることによって、関係を特定する。提案手法は式 (10) を用いて、非対角成分を一斉 (独立) に埋めるため、関係の予測結果の間に依存関係がない。なお、学習時には式 (5) を以下のように置き換えることで、学習が安定することが経験的にわかっている。

$$z_i = e_{w_i}. \quad (11)$$

これにより、単語単位で誤差逆伝播を行うことができ、プーリング操作による複雑なパラメータ更新を避けることでモデルの性能向上に寄与する。

2.4 学習と予測

固有表現抽出と関係抽出の交差エントロピーの総和を損失関数とする。固有表現抽出では Zhang

5) なお、提案手法は Multi-Head Self-Attention [14] に着想を得ているが、各固有表現の埋め込みにおける重みの計算を目的としない。

表 1 固有表現抽出と関係抽出の実験結果。NER は固有表現抽出の結果、RE は固有表現ラベルを用いない関係抽出の結果、RE+ は固有表現ラベルを用いた関係抽出の結果である。† は BERT-base、‡ は BERT-large、◊ は ALBERT をエンコーダとして用いた結果である。

	モデル	NER	RE	RE+
CoNLL04	Zhang ら [6]	85.6	-	67.8
	Li ら [15]†	87.8	-	68.9
	Eberts と Ulges [7]†	88.9	-	71.5
	Wang と Lu [4]◊	90.1	73.8	73.6
	提案手法†	90.2	72.8	72.6
	提案手法‡	90.5	73.8	73.8
ACE05	Zhang ら [6]	83.6	-	57.5
	Li ら [15]	84.8	-	60.2
	Dixit と Al-Onaizan [16]	86.0	62.8	-
	Wadden ら [17]†	88.6	63.4	-
	Lin ら [8]‡	88.8	67.5	-
	Wang ら [18]†	87.2	66.7	63.2
	Wang と Lu [4]‡	88.2	67.4	-
	Wang と Lu [4]◊	89.5	67.6	64.3
提案手法†	87.6	66.2	62.6	
提案手法‡	88.4	67.5	64.6	

ら [6] に倣い、学習時は正解のスパンとラベルを使い、予測時は貪欲法で得られたスパンとラベルを用いる。関係抽出では、学習時と予測時のいずれも固有表現抽出器が予測したスパンとラベルを用いる。

3 実験と考察

3.1 実験結果

提案手法の性能を確かめるために、CoNLL04 [10] と ACE05 の 2 つのデータセットで実験し、その性能を表 1 に示した。実験結果から、提案手法は関係抽出 (RE と RE+) において、両データセットの最高性能モデルに匹敵する性能を持つことを確認した。特に、提案手法と同じく BERT をエンコーダとして用いる Eberts と Ulges の手法 [7] や Wadden らの手法 [17] の性能を上回ることから、提案手法の構造が性能向上に寄与することが確かめられた。Wang と Lu [4] の手法は ALBERT [19] をエンコーダとして使うことで高い性能を達成しているが、エンコーダを BERT に揃えると、提案手法と同等の性能であった。

ところが、ACE05 での固有表現抽出 (NER) において、提案手法は比較手法である Wadden らの手法 [17] より低い精度を示している。これは比較手法が文書単位を入力とし、[SEP] で分割された複数の文に対して予測を行うため、文単位で予測を行う提案手法と実験設定が異なるからである。提案手法の固有表現抽出器に文書単位の入力を与えると、性能

表2 文書単位で固有表現抽出を行う実験の結果 (ACE05 テストデータ).

モデル	入力	NER		
		Pre.	Rec.	F1
Wadden ら [17] [†]	文書	-	-	88.6
提案手法 [†]	文	87.2	88.1	87.6
提案手法 [†]	文書	88.9	89.2	89.0

表3 デコーディング順序に関する対照実験の結果 (CoNLL04 テストデータ). 全ての関係を一齐に予測するモデルと, 順序を考慮して予測するモデルを比較する.

順序	NER			RE		
	Pre.	Rec.	F1	Pre.	Rec.	F1
考慮なし	89.9	90.6	90.2	75.1	70.6	72.8
考慮あり	89.6	90.5	90.0	73.1	70.3	71.7

が比較手法に匹敵することを表2に示した.

3.2 デコーディング順序による影響

Table-Filling を用いた既存手法は予め定義された順序に従い, モデルの出力結果を考慮しつつ, 表をセルごと埋める [3, 5, 6]. これらの研究は前の予測結果が以後の予測に役立つと仮定しているが, 提案手法では内積で表の非対角成分をまとめて予測するため, 各セルの予測は互いに独立である.

提案手法においてデコーディング順序を考慮する必要がないことを示すために, 順序を考慮してデコーディングを行う実験設定での性能を確認する. 具体的には, 単語対 (w_i, w_j) の関係を予測する時に, 左のセル (w_i, w_{j-1}) と下のセル (w_{i+1}, w_j) の情報を用いる. これにより, 表の左下から右上へと順番にデコーディングを行う手法を用意し, その性能を表3に示した.

実験結果より, デコーディング順序を考慮することによる性能向上は得られなかった. これは, BERT が長距離の依存関係を考慮して入力文をエンコードするため, 前の予測結果を考慮して逐次的に予測を行う必要が無くなったためと考えられる.

3.3 同時学習による影響

既存研究は, 固有表現抽出と関係抽出の同時学習はタスク間の依存関係を暗黙的にモデル化し, 全体の性能に寄与すると報告している [8, 20, 21]. 提案手法において両タスクを同時にモデル化するメリットはどのくらいあるのだろうか?そこで, 固有表現の認識結果から, 固有表現のセグメント情報やカテゴリ情報を隠したとき, 関係抽出の性能がどのような影響を受けるのかを調査した. 具体的には, 固有

表4 固有表現抽出器の学習・予測設定を変更することによる関係抽出器の性能変化 (CoNLL04 テストデータ).

「Seg.」はセグメント情報, 「Cat.」はカテゴリ情報を示す. また, \pm は固有表現のラベルをランダムに選んだ結果, \star は正解の固有表現のラベルを使った結果である.

学習		予測		RE		
Seg.	Cat.	Seg.	Cat.	Pre.	Rec.	F1
✓	✓	✓	✓	75.1	70.6	72.8
✓	✓	✓	✗	78.7	66.8	72.2
✓	✗	✓	✗	74.1	70.4	72.2
✓	✗	\pm	✗	76.0	65.7	70.5
✗	✗	\star	✗	79.1	75.6	77.3
✓	✓	\star	✗	81.9	75.1	78.3

表現ラベル (例: B-PER) から, セグメント情報だけを残したり (例: B), カテゴリ情報 (例: PER) だけを残して, 関係抽出器の性能を測定した (表4).

従来, 固有表現抽出の性能が関係抽出の性能に影響を与えると考えられてきたが, 提案手法では, カテゴリを用いずにセグメントだけを利用して関係抽出モデルの学習や予測を行っても, 性能に大きな差は現れないことが分かった. これは BERT をエンコーダとして用いたため, 固有表現のカテゴリは埋め込みから推測可能な場合が多く, カテゴリ情報を明示的に与えなくても関係が予測できてしまったためと考えられる. また, 予測時に固有表現のセグメントのラベルをランダムに書き換えた場合は, 固有表現のスパンの情報が得られなくなり, 関係抽出の性能低下を引き起こす. 一方で, 正解のスパンを関係抽出器に与えると, 大幅な性能向上が見られる. さらに, 固有表現抽出器の学習を加えることで, 性能が向上した. これは関係抽出器にとって, 固有表現の正確なスパン情報が重要であること, 固有表現抽出の学習を行うことによる BERT の単語埋め込みの改善が得られることを示唆している.

4 おわりに

本研究では, Table-Filling に基づき, 固有表現と関係を同時に抽出するモデルを提案した. 文脈を考慮した単語埋め込みの活用により, 従来手法の特徴量選択とデコーディング順序に関する課題を撤廃できることを示した. 実験結果により, 提案手法は CoNLL04 [10] と ACE05 の2つのデータセットで最高性能に匹敵する性能を示すことを確認した.

謝辞 この成果は, 国立研究開発法人新エネルギー・産業技術総合開発機構 (N E D O) の委託業務 (JPNP18002) の結果得られたものです.

参考文献

- [1] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, Vol. 3, p. 1083–1106, 2003.
- [2] GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 427–434, 2005.
- [3] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1858–1869, 2014.
- [4] Jue Wang and Wei Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1706–1721, 2020.
- [5] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (COLING)*, pp. 2537–2547, 2016.
- [6] Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1730–1740, 2017.
- [7] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In *24th European Conference on Artificial Intelligence (ECAI)*, 2020.
- [8] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7999–8009, 2020.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (NAACL)*, pp. 4171–4186, 2019.
- [10] Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pp. 1–8, 2004.
- [11] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pp. 147–155, 2009.
- [12] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations (ICLR)*, 2017.
- [13] Dat Quoc Nguyen and Karin Verspoor. End-to-end neural relation extraction using deep biaffine attention. In *Proceedings of the 41st European Conference on Information Retrieval (ECIR)*, 2019.
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, pp. 5998–6008, 2017.
- [15] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1340–1350, 2019.
- [16] Kalpit Dixit and Yaser Al-Onaizan. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5308–5314, 2019.
- [17] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5784–5789, 2019.
- [18] Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1692–1705, 2020.
- [19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*, 2020.
- [20] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1105–1116, 2016.
- [21] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, Vol. 8, pp. 64–77, 2020.
- [22] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface’s transformers: State-of-the-art natural language processing. *CoRR*, Vol. abs/1910.03771, , 2019.
- [23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- [24] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 402–412, 2014.
- [25] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (NAACL)*, pp. 3036–3046, 2019.

A 実験設定

提案手法は HuggingFace Transformers [22] を用いて実装し、NVIDIA Tesla V100(16 GiB) で実験を行った。本稿で報告された全ての実験結果は、5つの異なるランダムシードを用いて学習を行い、その平均値である。

なお、モデルには事前学習済みのパラメータとランダムな初期状態にあるパラメータが両方存在するため、一様な学習率を設定することはモデルの訓練において不適切だと考えられる。よって、Waddenら [17] に倣い、両抽出器と事前学習済みの BERT モデルを異なる学習率を用いて訓練する。訓練時のパラメータ更新は、AdamW [23] を用いる。

A.1 データセット

データセットの分割などの前処理と実験設定は既存研究に従う。

CoNLL04 固有表現が4種類、関係が5種類定義されている。Eberts と Ulges [7] と Gupta ら [5] の分割に従い、まずデータセットを訓練・検証・テストデータに分け、訓練データと検証データでハイパーパラメータを選択する。次に訓練データと検証データをあわせて訓練を行い、得られたモデルの性能をテストデータを用いて評価する [3, 6, 24].

ACE05 固有表現が7種類、関係が6種類定義されている。Waddenら [17] の前処理スクリプト⁶⁾を用いることにより、既存研究と同じ分割を作成した [4, 20, 25]. なお、配布されたデータのうち、head要素として記述された箇所を固有表現のスパンとして扱う [8, 20, 24, 25].

A.2 ハイパーパラメータ

提案手法により導入されたハイパーパラメータとその値を表5に示す。ハイパーパラメータの選択は、それぞれのデータセットの検証データを用いて行う。ただし、訓練時に使われたスケジューラーは、まず学習率をゼロから表に示した値まで {ウォームアップ期間} × {エポック数} のエポックをかけて線形に上昇させ、次に余弦関数で減衰させるものである。

表5 ハイパーパラメータの概要.

	CoNLL04	ACE05
固有表現埋め込みの次元数 $ I_y $	50	50
関係埋め込みの次元数 d_{att}	20	30
バッチサイズ	8	16
学習率 (BERT-base)	5×10^{-5}	5×10^{-5}
学習率 (その他)	1×10^{-3}	1×10^{-3}
ドロップアウト	0.3	0.3
ウォームアップ期間	0.2	0.2
エポック数	30	30

表6 アブレーションスタディの結果 (CoNLL04 テストデータ). 「-ラベル」は式2と式4からラベルの埋め込みを削ったモデル, 「-スパン」は式2からスパンの埋め込みを削ったモデル, 「-両方」は両方を削ったモデルを示す.

モデル	NER			RE		
	Pre.	Rec.	F1	Pre.	Rec.	F1
提案手法	89.7	90.5	90.1	74.7	70.8	72.7
-ラベル	89.7	90.5	90.1	74.1	70.6	72.3
-スパン	89.5	90.6	90.0	74.3	69.6	71.9
-両方	89.4	90.2	89.8	73.5	69.7	71.3

B 特徴量による影響

提案手法で特徴量として用いたスパンの埋め込みとラベルの埋め込み (式2と式11) がモデルに与える影響を調べるために、各埋め込みを削った場合の実験を行い、結果を表6に示す。式2からスパンの埋め込み $f(e_{w_{\text{first}(i-1)}}, \dots, e_{w_{i-1}})$ を削る目的は、スパン単位のパラメータ更新を防止し、スパンの埋め込み表現を学習させないことである。

スパンの埋め込み表現を学習させないことにより、関係抽出器の性能低下が表6から見られた。これはスパンの埋め込みが提案手法において重要であることを示し、§3.3の結論を裏付ける結果だと考えられる。

6) <https://github.com/dwadden/dygiepp>