事例ベース依存構造解析のための依存関係表現学習

大内 啓樹 [1] 鈴木 潤 [2,1] 小林 颯介 [2,3] 横井 祥 [2,1] 栗林 樹生 [2,4] 吉川 将司 [2,1] 乾 健太郎 [2,1]
[1] 理化学研究所 [2] 東北大学 [3] 株式会社 Preferred Networks [4]Langsmith 株式会社
hiroki.ouchi@riken.jp, yoshikawa@tohoku.ac.jp,
{jun.suzuki,sosk,yokoi,kuribayashi,inui}@ecei.tohoku.ac.jp

1 はじめに

過去に観測した類似事例にもとづき，新たな事例に関する推論を行うことを**事例ベース推論**と呼ぶ．近年の深層学習の隆盛にともなって，「モデルの予測過程の説明性」の観点から注目を集めている．ニューラルネットワークを用いた一般的なモデルでは，予測根拠の解釈が困難であることが指摘されている [1]．一方で，k 近傍法などの事例ベース推論手法では，学習事例を直接的に予測に使用し，それらを予測根拠として予測への貢献度とともに取り出すことが容易である．予測に大きく貢献した学習事例をユーザに提示することは，「**なぜモデルはそのような予測をしたのか？**」という疑問への一種の説明の役割を果たす．この種の説明は**事例ベース説明**と呼ばれる．機械学習の専門知識を持たないユーザにとっても直感的である場合が多く，モデルの予測に対するユーザの理解を促進し，より自信を持った意思決定につながるとの報告がある [2, 3, 4, 5]．

このような利点がある一方，予測性能の高い事例ベースモデルの構築は容易ではない [6, 7]．高い予測性能を保つ鍵は**特徴空間の効果的な学習**にある．いかにして「同じクラスの事例同士を特徴空間上で近づけ，異なるクラスの事例同士は遠ざけるか」を考えなければならない．この指針のもと，Wiseman ら [6] や Ouchi ら [7] は言語の系列データのための特徴表現学習手法を提案し，一般的なモデルと同等の予測性能を保ちつつ，説明性の優れる事例ベースモデルが構築可能となったことを報告している．

本稿では，系列データからの自然な展開として，言語的な**関係**に着目する．具体的には，依存構造解析で定義される依存関係（エッジ）に焦点を絞り，事例ベース依存構造解析のための表現学習手法を提案する．実験を通して，一般的な解析システムと同等の性能を保ちつつ，説明性に優れるシステムが構築可能であることを示す．さらに，説明性を大きく損なう可能性のある，**ハブ**と呼ばれる事例の出現に関する分析結果も報告する．

2 手法

一般的な解析手法は推論時に学習データを参照しない．本稿の事例ベース解析手法は，推論時にも学習データを参照しながら解析を行う点が特徴で，予測根拠を学習事例に求められるという利点がある．

2.1 問題設定

本稿ではラベルなし依存構造解析[1)]に取り組む．具体的には**主辞選択 (head selection)** 問題 [8, 9, 10] として解く．これは簡潔な問題形式であり，GPU を用いた計算にもやさしいという理由から採用する．

トークナイズした入力文を $X = (x_0, x_1, \ldots, x_T)$ と表す．x_0 は特殊トークンである ROOT を表し，$x_1, \ldots, x_T$ は原文の T トークンを表す．各トークン x_i の主辞が x_j である確率を以下のように定義する．

$$P(x_j|x_i) = \frac{\exp(\mathrm{score}(x_j, x_i))}{\sum_{k=0}^{T} \exp(\mathrm{score}(x_k, x_i))}. \quad (1)$$

スコア関数 $\mathrm{score}(x_j, x_i)$ は実数を返す任意の関数として定義可能である（2.2 節参照）．

推論時は，入力文の各トークン x_i に対して最大確率のトークンを主辞として選択する．

$$\hat{y}_i = \underset{x_k : 0 \le k \le T}{\arg\max}\, P(x_k|x_i).$$

学習時は負の対数尤度を最小化することによって，正解主辞の確率が高くなるように学習する．

$$L = -\sum_{n=1}^{|D|} \sum_{i=1}^{T^{(n)}} \log P(y_i^{(n)}|x_i^{(n)}).$$

$D = \{X^{(n)}, Y^{(n)}\}_{n=1}^{|D|}$ は学習データ，$y_i^{(n)} \in Y^{(n)}$ は入力文の各トークン $x_i^{(n)} \in X^{(n)}$ の正解主辞である．

1) ラベルあり依存構造解析も行ったが，紙面（分量）の都合上，本稿ではラベルなし依存構造解析に絞って議論する．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

2.2 事例間の類似度に基づくスコア

式 1 中のスコア関数として，学習事例との類似度に基づくスコア（**事例ベーススコア**）を提案する．

$$\mathrm{score}(x_j, x_i) = \sum_{\langle x_\ell, x_k\rangle \in A_D} \mathrm{sim}(\boldsymbol{h}_{\langle \ell,k\rangle}, \boldsymbol{h}_{\langle j,i\rangle}) \tag{2}$$

ここで，$\boldsymbol{h}_{\langle j,i\rangle} \in \mathbb{R}^d$ はエッジ $\langle$ 主辞 x_j, 従属部 $x_i\rangle$ の d 次元特徴ベクトル，sim は類似度関数を表す．本研究ではベクトル間の内積 $\mathrm{sim}_{\mathrm{dot}}(\boldsymbol{a},\boldsymbol{b}) = \boldsymbol{a}^\top\boldsymbol{b}$ とコサイン類似度 $\mathrm{sim}_{\cos}(\boldsymbol{a},\boldsymbol{b}) = \frac{\boldsymbol{a}^\top\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$ を用いる．A_D は学習データ D 中の正解エッジからなる集合である．

$$A_D = \{\langle y_i, x_i\rangle \mid x_i \in X, y_i \in Y, (X,Y) \in D\}.$$

$y_i \in Y$ はトークン $x_i \in X$ に対する正解主辞トークンを表す．つまり，式 2 では学習データ中のエッジとの類似度の総和をスコアとしている．各類似度を予測スコアに対する貢献（寄与）度と見ることができ，「**各学習事例が予測にどの程度貢献しているか**」を人間が理解しやすい形式で示すことができる．

■**効率的な計算方法．** T トークンからなる入力文の各トークン x_i（$i = 1,\ldots,T$）に対して，$T+1$ 個のトークン x_j（$j = 0,\ldots,T$）から主辞を選ぶ場合，d 次元の特徴ベクトルを用いるスコア計算には $O(T^2d)$ の計算量[2]がかかる．一般的な事例ベース手法では，スコア計算が学習データ D の特徴ベクトルにも依存するため，$O(T^2d|D|d) = O(T^2d^2|D|)$ の計算量がかかる．一方，我々の事例ベース手法では学習時の計算量を小さく抑えられる．類似度関数 sim として内積を使う場合[3]，式 2 は以下のように書ける．

$$\mathrm{score}(x_j, x_i) = \sum_{\langle x_\ell, x_k\rangle \in A_D} \boldsymbol{h}_{\langle \ell,k\rangle}^\top \boldsymbol{h}_{\langle j,i\rangle} = \boldsymbol{h}_{\langle j,i\rangle}^\top \sum_{\langle x_\ell, x_k\rangle \in A_D} \boldsymbol{h}_{\langle \ell,k\rangle}.$$

つまり，学習データ中のすべてのベクトルを足し込んでから，当該エッジベクトル $\boldsymbol{h}_{\langle j,i\rangle}$ との内積を計算すればよい．学習データ中のベクトルの足し算は $O(|D|d)$ であり，一度計算してしまえば当該エッジベクトル $\boldsymbol{h}_{\langle j,i\rangle}$ の本数に乗法的に依存しない．よって，通常の手法なら計算量 $O(T^2d^2|D|)$ であるところを，$O(T^2d + |D|d)$ に抑えられる．さらに，先行研究 [7] に倣い，学習データからランダムに M 文をサンプリングしたサブセット $D' = \{(X'_m, Y'_m)\}_{m=1}^M$ を用いることにより，学習はシングル GPU[4] で 24 時間以内に完了できるようになった．

2） d 次元ベクトル間の内積計算・加減算は $O(d)$ とする．
3） コサイン類似度を使用する際も同様に書き換えられる．
4） NVIDIA DGX-1 with Tesla V100.

■**最も関連する既存手法．** Neighbourhood Components Analysis (NCA) [11, 6, 7] では，事例 a の近傍に学習事例 $b \in D$ がくる確率を $P(b|a) = \frac{\exp(\mathrm{sim}(b,a))}{\sum_{b'\in D}\exp(\mathrm{sim}(b',a))}$ と定義し，同一クラスの事例同士の確率が高くなるように学習する．各類似度に対して exp をとるため，類似度が高い一部の事例に確率値が大きく左右される．我々の手法は類似度を単純に足し合わせるため，そのような敏感さを伴わない．両手法の性能比較は今後の課題だが[5]，学習時の計算量の面では，NCA が $O(T^2d^2|D|)$ であるのに対し，我々の手法は $O(T^2d + |D|d)$ であるため有利であると言える．

■**一般的な手法との関連．** 重みパラメータベクトル $\boldsymbol{w} \in \mathbb{R}^d$ を用いた以下のスコア関数は，通常のニューラルモデルの出力層のスコア関数に該当する．

$$\mathrm{score}(x_j, x_i) = \mathrm{sim}(\boldsymbol{w}, \boldsymbol{h}_{\langle j,i\rangle}). \tag{3}$$

この**重みベーススコア**に基づいて特徴空間を学習し，推論時はその学習済み特徴空間を事例ベーススコア計算のために使用するといったように併用することもできる．近年の画像認識分野においても，重みパラメータに基づく特徴表現学習と事例に基づく推論（k 近傍法）を組み合わせた手法が提案され，その有用性が報告されている [12, 13, 14, 15]．依存構造解析においてそのような試みは未だないが，有望な手法のひとつであると考えられるため，本研究でもこれらの組み合わせに関する検証を行う（3 節）．

■**エッジ特徴ベクトル．** 各エッジ $\langle x_j, x_i\rangle$ の特徴ベクトルを以下のように定義する．

$$\boldsymbol{h}_{\langle j,i\rangle} = g(\boldsymbol{h}_i^{\mathrm{dep}}, \boldsymbol{h}_j^{\mathrm{head}}). \tag{4}$$

$\boldsymbol{h}^{\mathrm{dep}}, \boldsymbol{h}^{\mathrm{head}} \in \mathbb{R}^d$ はそれぞれ従属部（dependent）と主辞（head）の d 次元ベクトルであり，ニューラルエンコーダ[6]を通して得られる．関数 g として，2 本のベクトルの相互作用を効果的にとらえられる演算が望ましい．本稿では知識ベースにおける関係の表現学習の知見 [16, 17, 18] から，主要な手法のひとつである乗法的（multiplicative）合成を採用する．[7]

$$\boldsymbol{g}_{\mathrm{mul}}(\boldsymbol{h}_i^{\mathrm{dep}}, \boldsymbol{h}_j^{\mathrm{head}}) = \boldsymbol{W}(\boldsymbol{h}_i^{\mathrm{dep}} \odot \boldsymbol{h}_j^{\mathrm{head}}).$$

ここで，ベクトルの要素積（$\odot$）によって相互作用を考慮し，パラメータ行列 $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ を用いた線形変換によってベクトルを合成する．

5） シングル GPU を用いた事前実験では，NCA で長い文を学習する際にメモリオーバーになり，実験が完了できなかった．
6） 付録 A.2 を参照．
7） もうひとつの主要な手法である加法的（additive）合成も試したが，事前実験において予測性能が乗法的合成と比べて大幅に劣る結果となった．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

ID	学習	推論	類似度	PTB	UD (avg.)
(α)	重み	重み	dot	96.4	89.1
(β)	重み	重み	cos	96.4	89.0
(a)	重み	事例	dot	96.4	89.1
(b)	重み	事例	cos	85.6	81.2
(c)	事例	事例	dot	96.4	89.3
(d)	事例	事例	cos	96.2	89.1

表 1: 各テストデータにおけるラベルなし正解率.

3 実験と考察

■データセット. 標準的なベンチマークデータである英語の PennTreebank (PTB) [19] と Universal Dependencies (UD) [20] に含まれる多言語データセットを使用する．特に UD では，先行研究 [21] に従い，語族やデータ量などの多様性を考慮した 13 言語のデータセットを使用する．[8)]

■エンコーダ. BERT [22] を用いて計算した各トークンの特徴表現を双方向 LSTM [23] に入力し，その隠れ層から式 4 中の $\boldsymbol{h}^{\text{dep}}$ と $\boldsymbol{h}^{\text{head}}$ を計算する．これは，実験に用いる全システムに共通である．[9)]

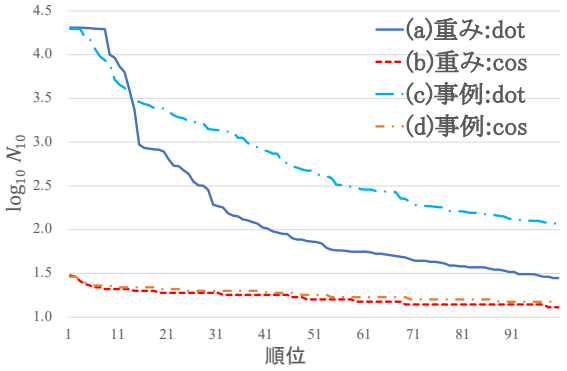
■正解率の比較. 表 1 は 6 つのシステムの正解率（unlabeled attachment score）[10)] を表している．システム (α) と (β) は学習・推論時ともに重みベーススコア（式 3）を用いた一般的な解析システムである．これらのシステム間で類似度関数の違いによる正解率の差は見られなかった．システム (a) と (b) は，学習時は重みベーススコア（式 3）を用い，推論時は事例ベーススコア（式 2）を用いている．内積（dot）を類似度関数として用いた (a) は，通常のシステム (α) と (β) と同等の正解率を保つことに成功している．対照的に，コサイン類似度（cos）を用いた (b) では正解率が大きく劣化している．システム (c) と (d) は，学習・推論時ともに事例ベーススコア（式 2）を用いている．これらのシステムも，一般的な解析システム (α) と (β) と同等の正解率を保っている．また，事例ベーススコアを学習・推論時に一貫して使用する場合は，類似度関数の違いが大きな正解率の差を生むわけではないこともわかった．これらの予測性能を見ると，事例ベース推論を採用したシステムとして (a) と (c) と (d) が良いように思われる．以降で，各システムの挙動をより詳細に分析する．

8） 各言語のデータセットの詳細は付録 A.1 の表 5 を参照.
9） モデルやハイパーパラメータ，実験設定の詳細については付録 A.2 を参照.
10） UD (avg.) は，全テストデータの正解率のマクロ平均である．各テストデータの正解率は付録 A.3 の表 7 を参照.

ID	学習	推論	類似度	N_{10}	学習事例
(a)	重み	事例	dot	20,332	⟨ROOT, find⟩
				20,302	⟨ROOT, find⟩
				20,299	⟨ROOT, find⟩
(b)	重み	事例	cos	30	⟨ROOT, people⟩
				29	⟨ROOT, build⟩
				26	⟨ROOT, parents⟩
(c)	事例	事例	dot	19,711	⟨instead, of⟩
				19,632	⟨due, to⟩
				19,611	⟨because, of⟩
(d)	事例	事例	cos	29	⟨Europe, camps⟩
				29	⟨her, everyone⟩
				27	⟨services, it⟩

表 2: 各システムにおける N_{10} 上位 3 件の実例.

図 1: N_{10} の上位 100 事例のプロット.



■近傍に出現する学習事例の偏り. 事例ベース推論を取り入れたシステム (a)〜(d) において，実際に予測に使われた近傍事例を分析する．具体的には，各学習事例 $\langle x_b, x_a \rangle$ が開発データの各事例 $\langle w_j, w_i \rangle$ の類似度上位 k 位以内に選出された回数（**近傍選出回数**）$N_k(\langle x_b, x_a \rangle)$ [24, 25] を分析する．この**近傍選出回数**が極端に高い学習事例を**ハブ**と呼ぶ [24]．本稿では，どのような学習事例がハブとなり，それらの事例は説明性（特に人間の納得度 *plausibility* [26]）の観点から見た時に有用であるかを調べる．

分析には UD-English データを用いた．表 2 は N_{10} 上位 3 件の学習事例を示している．システム (a) では，ROOT を主辞とする学習事例ばかりが開発データの近傍事例として選出され，システム (c) では，instead of や due to，because of のような定型表現が頻繁に選出される傾向が見られた．このような一部の学習事例が，開発データ内の大多数の事例の説明として人間にとって納得のいくように機能するとは言い難い．これは，**事例ベース推論がいつでも有益な説明を与えるというわけではない**ことを示唆する．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

ID	学習	推論	類似度	PTB	UD-English
(a)	重み	事例	dot	2.0	8.0
(b)	重み	事例	cos	52.9	43.9
(c)	事例	事例	dot	2.9	3.2
(d)	事例	事例	cos	73.1	46.9

表 3: Zero-shot ラベルあり依存構造解析の正解率．

対照的に，コサイン類似度を採用したシステム (b) と (d) では，一定の事例に極端な偏りは見られない．図 1 は，N_{10} の高い学習事例上位 100 事例を $\log_{10}$ スケールで降順にプロットしたものである．類似度関数として内積を用いたシステム (a) と (c) の場合，近傍選出回数に大きな偏りがあり，ハブが出現していることがわかる．

偏りの原因のひとつとして，学習事例の特徴ベクトルの長さ（L2 ノルム）が挙げられる．たとえば，表 2 中の ⟨ROOT, find⟩ のノルムは 180.4 であるのに対し，N_{10} 順位 100 位の ⟨horse, a⟩ は 75.5 であった．このように，N_k 順位上位の一部の事例は，その他の事例よりもノルムが大きくなるという傾向が見られた．コサイン類似度を用いる場合，ノルムを正規化しているため，N_k の偏りは観測されなかった．

■クエリ事例と近傍事例の類似性． 類似性を議論する上で重要なのは，「どのような観点で類似しているか」である．多様な観点があり得るが，本研究では，依存構造の関係ラベルの観点でクエリ事例と近傍事例が類似しているかを調査する．データセット中の各エッジ $\langle w_j, w_i \rangle$ には，人間が定義した関係ラベルセットのうちのひとつ $r \in R$ が付与されている．したがって，同一の関係ラベルを付与されているエッジ同士は，異なるラベルを付与されているエッジ同士よりも，各関係ラベルの定義やラベルセットの設計思想の観点から人間が見れば類似している（より多くの特徴を共有している）と言える．そこで，クエリ事例と近傍事例の持つ関係ラベルの一致率を測定し，それらの事例が人間にとっても類似していると言えるかを近似的に評価する．[11)]

まず，構築した各システムを用いて開発データ内の文を解析する．次に，得られたエッジ（クエリ事例）と最も類似度の高い学習事例（最近傍事例）を選出する．最後に，クエリ事例と最近傍事例に付与されている関係ラベルが同じなら正解[12)]とみなす．

11） 実際に人間の感じる類似性とどの程度相関するかを調査することは今後の課題とする．

12） そもそも同定したエッジが不正解であった場合は，そのまま不正解とみなす．

クエリ事例（開発データ）	最近傍事例（学習データ）
sconj If they continue to add ...	sconj If you want to start ...
nmod:poss ... your appeal is ...	det ... the staff was ...

表 4: システム (d) によるエッジ検索の実例．

この設定は，関係ラベルを学習に用いないため，「Zero-shot ラベルあり依存構造解析」とみなせる．

表 3 に関係ラベルの正解率（labeled attachment score）を記す．関係ラベルを学習に用いていないにも関わらず，コサイン類似度を採用したシステム (b) と (d) は，多くのクエリ事例に対して同一の関係ラベルを持つ学習事例を最近傍事例として選出している．特にシステム (d) は PTB において 70%を超える正解率を達成している．これらの結果から，コサイン類似度に基づいて選出した事例は，関係ラベルセットの観点から見た時，クエリ事例と類似していることが示唆された．

最後に，これまでの分析を踏まえると最良であるシステム (d) の最近傍事例を表 4 に示す．一行目の例のように，クエリ事例の主辞または従属部が機能語（この例では If）である場合に，同一の関係ラベルを持つ事例を正しく検索できている傾向が見られた．また，二行目の例のように，クエリ事例の主辞が名詞（この例では appeal）である場合，関係ラベルに関わらず，名詞を主辞（この例では staff）とするような事例が近傍にくる傾向が見られた．

■評価・分析のまとめ． 次の知見を得た: (i) 提案した事例ベースシステムは一般的なシステムと同等の性能を保つことができる，(ii) 類似度関数に内積を使うと，説明性（人間の納得度）の観点で望ましくないハブが出現する，(iii) コサイン類似度に基づいて学習・検索した場合，関係ラベルの観点で類似している事例が近傍に出現する傾向がある．

■今後の展開． 本研究のエッジ特徴ベクトルを用いることによって，単語レベルの類似度に基づく検索を超えた，**依存関係レベルの類似度に基づく検索**が可能になる．同様に，2 文間の依存構造のソフトなマッチングが可能であるため，同義文判定や文書検索などの下流タスクへの応用も考えられる．

謝辞． 本研究は JSPS 科研費 19K20351, 19H04162 の助成を受けたものです．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

参考文献

[1] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of EMNLP*, pp. 107–117, 2016.

[2] Janet L Kolodneer. Improving human decision making through case-based decision aiding. *AI magazine*, Vol. 12, No. 2, pp. 52–52, 1991.

[3] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of KDD*, pp. 1135–1144, 2016.

[4] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of NIPS*, pp. 4765–4774, 2017.

[5] Christoph Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[6] Sam Wiseman and Karl Stratos. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of ACL*, pp. 5363–5369, July 2019.

[7] Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. Instance-based learning of span representations: A case study through named entity recognition. In *Proceedings of ACL*, pp. 6452–6459, Online, July 2020. Association for Computational Linguistics.

[8] Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. Dependency parsing as head selection. In *Proceedings of EACL*, pp. 665–676, April 2017.

[9] Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*, 2017.

[10] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of EMNLP*, pp. 1923–1933. Association for Computational Linguistics, September 2017.

[11] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan R Salakhutdinov. Neighbourhood components analysis. In *Proceedings of NIPS*, pp. 513–520, 2005.

[12] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

[14] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.

[15] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.

[16] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, Vol. 26, pp. 2787–2795, 2013.

[17] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. 2015.

[18] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, 2016.

[19] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[20] Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal Dependency annotation for multilingual parsing. In *Proceedings of ACL*, pp. 92–97, 2013.

[21] Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In *Proceedings of EMNLP-IJCNLP*, pp. 2755–2768, November 2019.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[23] Alan Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *Proceedings of Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop*, 2013.

[24] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, Vol. 11, No. sept, pp. 2487–2531, 2010.

[25] Yutaro Shigeto, Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. Ridge regression, hubness, and zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 135–151. Springer, 2015.

[26] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of ACL*, pp. 4198–4205, July 2020.

[27] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[28] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

A 付録

A.1 データセットの詳細

言語	データセット名	語族	語順	学習データ
Arabic	PADT	non-IE	VSO	6.1k
Basque	BDT	non-IE	SOV	5.4k
Chinese	GSD	non-IE	SVO	4.0k
English	EWT	IE	SVO	12.5k
Finnish	TDT	non-IE	SVO	12.2k
Hebrew	HTB	non-IE	SVO	5.2k
Hindi	HDTB	IE	SOV	13.3k
Italian	ISDT	IE	SVO	13.1k
Japanese	GSD	non-IE	SOV	7.1k
Korean	GSD	non-IE	SOV	4.4k
Russian	SynTagRus	IE	SVO	48.8k
Swedish	Talbanken	IE	SVO	4.3k
Turkish	IMST	non-IE	SOV	3.7k

表 5: 実験に使用した各言語の UD データセット．「語族」の列の IE は Indo-European を表す．「語順」の列は各言語における一般的な語順を示す．「学習データ」は文数を千単位（k）で表す．

Universal Dependencies (UD) の最新バージョン 2.7[13] を用いた．表 5 に各データの統計情報を記す．

A.2 モデルと実験設定の詳細

単語埋め込み	300 次元 fastText [27]
BERT	BERT-Base, Multilingual Cased [22]
CNN ウィンドウ	3
CNN フィルタ	30
LSTM 隠れ層 (d)	300 次元
最適化	Adam [28]
学習率	0.001
ミニバッチサイズ	{8, 16, 32}
ドロップアウト	{0.1, 0.2, 0.3}
D' のサイズ M	10

表 6: 実験に使用したハイパーパラメータ．

式 4 を再掲する: $\boldsymbol{h}_{\langle j,i\rangle} = g(\boldsymbol{h}_i^{\text{dep}}, \boldsymbol{h}_j^{\text{head}})$．文を入力したところから，この式中の $\boldsymbol{h}^{\text{dep}}$ と $\boldsymbol{h}^{\text{head}}$ を作るまでの流れを記述する．まず，入力文の系列 $X = (x_0, x_1, \ldots, x_T)$[14] を，単語埋め込み，文字畳み込みニューラルネットワーク（CNN），BERT [22] を用いてベクトル表現の系列に変換する．各トークンに単語埋め込み表現[15] を割り当てた系列を $\boldsymbol{w}_{0:T} = (\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_T)$ と表す．各トークンを文字 CNN で変換した系列を $\boldsymbol{c}_{0:T} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_T)$ と表す．各トークンに BERT の表現を割り当てた系列を $\boldsymbol{b}_{0:T} = (\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_T)$ と表す[16]．これらの表現を各トークン x_t ごとに結合して，トークン表現 $\boldsymbol{x}_t = [\boldsymbol{w}_t; \boldsymbol{c}_t; \boldsymbol{b}_t]$ を得る．

次に，トークン表現系列 $\boldsymbol{x}_{0:T} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ を双方向 LSTM に入力し，その隠れ層のベクトル表現の系列 $\overrightarrow{\boldsymbol{h}}_{0:T} = (\overrightarrow{\boldsymbol{h}}_0, \overrightarrow{\boldsymbol{h}}_1, \ldots, \overrightarrow{\boldsymbol{h}}_T)$ と $\overleftarrow{\boldsymbol{h}}_{0:T} = (\overleftarrow{\boldsymbol{h}}_0, \overleftarrow{\boldsymbol{h}}_1, \ldots, \overleftarrow{\boldsymbol{h}}_T)$ を得る．ここで，$\overrightarrow{\boldsymbol{h}}_t \in \mathbb{R}^d$ と $\overleftarrow{\boldsymbol{h}}_t \in \mathbb{R}^d$ はそれぞれ d 次元の前向きと後向きの LSTM の隠れ層を表す．これらの表現を結合して，各トークンの表現 $\boldsymbol{h}_t^{\text{lstm}} = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t] \in \mathbb{R}^{2d}$ を得る．

最後に，ベクトル系列 $\boldsymbol{h}_{0:T}^{\text{lstm}} = (\boldsymbol{h}_0^{\text{lstm}}, \boldsymbol{h}_1^{\text{lstm}}, \ldots, \boldsymbol{h}_T^{\text{lstm}})$ の各ベクトルを変換し，式 4 中の表現 $\boldsymbol{h}_t^{\text{dep}} = \boldsymbol{W}^{\text{dep}} \boldsymbol{h}_t^{\text{lstm}}$ と $\boldsymbol{h}_t^{\text{head}} = \boldsymbol{W}^{\text{head}} \boldsymbol{h}_t^{\text{lstm}}$ を得る．ここで，$\boldsymbol{W}^{\text{dep}} \in \mathbb{R}^{d \times 2d}$ と $\boldsymbol{W}^{\text{head}} \in \mathbb{R}^{d \times 2d}$ はそれぞれ，$\boldsymbol{h}_t^{\text{dep}} \in \mathbb{R}^d$ と $\boldsymbol{h}_t^{\text{head}} \in \mathbb{R}^d$ に変換するための重み行列である．実験に使用したハイパーパラメータは表 6 に記す．

A.3 結果の詳細

学習	重み				事例	
推論	重み		事例		事例	
類似度	dot	cos	dot	cos	dot	cos
ID	(α)	(β)	(a)	(b)	(c)	(d)
PTB	96.4	96.4	96.4	85.6	96.4	96.2
UD-Arabic	87.9	87.8	87.9	77.7	88.0	88.0
UD-Basque	85.1	85.4	85.1	80.2	85.3	85.0
UD-Chinese	85.9	85.5	85.9	79.3	86.1	85.7
UD-English	90.8	90.9	90.8	85.8	91.1	90.8
UD-Finnish	89.4	89.3	89.4	84.2	89.5	89.5
UD-Hebrew	89.7	89.4	89.7	82.6	89.9	89.5
UD-Hindi	94.8	94.9	94.8	90.0	94.9	94.8
UD-Italian	94.3	94.0	94.3	86.5	94.3	94.2
UD-Japanese	94.3	94.4	94.3	91.0	94.6	94.5
UD-Korean	87.9	88.0	88.0	85.1	88.2	88.2
UD-Russian	94.2	94.2	94.2	54.6	94.3	94.1
UD-Swedish	90.3	90.3	90.3	87.8	90.6	90.2
UD-Turkish	73.1	73.5	73.1	70.7	73.7	73.6
UD (avg.)	89.1	89.0	89.1	81.2	89.3	89.1

表 7: 各テストデータにおける正解率．

表 7 に全テストデータの正解率を記す．各正解率は，異なるランダムシードを用いて独立に学習した 3 つのモデルの平均正解率である．

13） https://universaldependencies.org/#download
14） PTB と UD ともに，データセットですでにトークナイズされている系列を使った．
15） 学習時に更新せず，固定の表現を使用する．
16） 各トークン $x_t \in X$ をサブワード分割してから BERT でエンコードする．ここで，各トークン x_t の先頭のサブワードに対するベクトル表現（BERT の最終層の表現）を $\boldsymbol{b}_t$ とする．この表現は学習時に更新せず，固定のものを使用する．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).