

# 階層コード表現を用いた上位下位関係の識別

水木 栄

東京工業大学情報理工学院  
sakae.mizuki@nlp.c.titech.ac.jp

岡崎 直観

東京工業大学情報理工学院  
okazaki@c.titech.ac.jp

## 1 はじめに

単語間の上位下位関係を表す語彙知識は、含意関係認識 [1] やテキスト生成 [2] など幅広いタスクで用いられる。また、多数の上位下位関係を階層構造に集約すると、タクソノミを構築できる [3]。こうした中で上位下位関係識別は、タクソノミの自動構築を実現する基礎技術として位置付けられている [4]。

タクソノミ構築を志向して上位下位識別タスクを解くうえでは、識別精度だけでなく、推論結果が推移性や反対称性を満たすことが重要である [4]。前者は単語分散表現と語彙知識を併用できる Semantic Specialization が、後者は順序関係を定義可能な Order Embeddings が有望であるが、両者を兼ね備えた手法は提案されていない。

本研究では学習済み単語分散表現を階層性のあるコード表現に変換して、単語ペアの上位下位関係らしさを計量する方法を提案する。提案手法により得られた階層コードの例を、付録 A の表 4 に示す。これにより、Semantic Specialization の長所である単語分散表現の活用と、Order Embeddings の長所である推移性・反対称性の両立を志向する。また WordNet から得られる語彙知識を用いて学習することにより、上位下位関係識別タスクにおいて既存手法を上回る性能が実現できることを報告する。本研究の貢献は次の 3 点である。分散表現を、階層性を備えたコード表現に変換する手法の提案。コード表現を用いて、単語ペアの上位下位関係の程度を計量する手法の提案。提案手法の上位下位関係識別タスクへの有効性を、実験的に示したこと。

## 2 既存研究

上位下位関係識別とは、与えられた単語ペア、たとえば (animal,dog) が上位下位関係か否かを識別するタスクである。本研究の設定は、WordNet のような汎用的かつ大規模な語彙資源から得られる知識を訓練データとする表現学習である。獲得した表現

を直接用いて上位下位関係を推論する。同様の設定に従う既存研究は、Semantic Specialization, Order Embeddings の 2 つである。また、本研究の提案手法と関連性が深いのはコード学習の手法である。

Semantic Specialization とは、語彙資源から得られる単語間の意味関係を単語分散表現に反映する手法の総称である。その長所は、文脈類似性に基づく意味的類似度と、語彙資源に基づく詳細な意味関係の両方を獲得できることである。特に上位下位関係の場合は、分散表現のノルム長さにより意味階層を、コサイン類似度により関係の強さを表現する方法が提案されている [5]。Vulic ら [6] はこの表現方法に則り、上位下位・同義・対義関係の語彙知識を反映した単語分散表現を得る手法を提案し、上位下位関係識別タスクで最高精度を報告している。

Order Embeddings [7] とは、順序関係を定義可能な表現形式または空間による埋め込み表現の総称である。その長所は、順序集合の定義により推移性<sup>1)</sup>および反対称性<sup>2)</sup>という、タクソノミが一般に有する性質がおのずと満たされることである。既存研究では確率分布 [8] や超直方体 [9] などを用いる手法が提案されている。

コード学習とは、連続空間上のベクトルを M 進 N 桁の離散ベクトルに変換する手法である。この手法の長所は、連続空間上の類似性を維持したまま離散空間上でコンパクトな表現を得ることである。既存研究では、モデル圧縮 [10] やクラスタリング [11] に応用されている。一方で、本研究が目指す階層性を備えたコードを学習する手法は提案されていない。

## 3 提案手法

### 3.1 階層コードおよび上位下位関係の定義

本論文の階層コードの定義は M 進 N 桁、ただしひとつたび 0 が出現したら後続桁はすべて 0 となる離

- 1)  $s < t$  かつ  $t < u$  ならば  $s < u$  が成立する性質のこと。
- 2)  $s < t$  ならば  $t \neq s$  が成立する性質のこと。

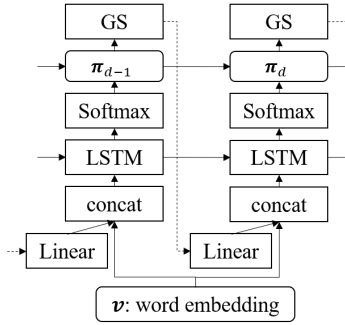


図1 変換器のアーキテクチャ。角丸は変数，直角は関数，点線は誤差逆伝搬なしを表す。記号は数式に準拠

散ベクトルである。たとえば  $[3, 1, 0, 0]$  は階層コードである。深さ  $N$ ・最大幅  $M$  の木構造は，幅優先符号化により階層コードに変換できる。ここから直感的にわかるように，階層コードを用いて上位下位関係を定義できる。すなわち a) 上位より下位のほうが非ゼロ桁数が多く，かつ b) 非ゼロ桁の値がすべて一致する 関係であると定義すればよい。たとえば  $([3, 1, 0, 0], [3, 1, 7, 0])$  は上位下位関係である。

### 3.2 変換器 (エンコーダ)

変換器 (エンコーダ) は，単語分散表現  $\mathbf{v}$  を  $M$  進  $N$  桁の階層コード  $\mathbf{C}$  の確率分布  $P(\mathbf{C})$  に変換する関数である。LSTM による再帰的計算および Gumbel-Softmax Trick [12] による擬似サンプリングを用いて，下位桁が上位桁の値に依存するような関数をモデリングする。アーキテクチャの模式図を図1に掲載する。

$d$  桁目の階層コードを表す確率変数を  $C_d$  とする。階層コードの定義 3.1 より  $C_d$  の条件付分布は

$$\begin{aligned} P(C_d = a|C_{<d}) &= 1\{a = 0\}P(C_{d-1} = 0|C_{<d-1}) \\ &+ P(C_d = a|C_{d-1} \neq 0, C_{<d})P(C_{d-1} \neq 0|C_{<d-1}) \quad (1) \end{aligned}$$

と表される。つまりゼロを出力する確率は桁が進むにつれて単調増加する。そこで LSTM を用いて， $d-1$  桁目がゼロでない場合の条件付確率  $P(C_d|C_{d-1} \neq 0, C_{<d})$  のカテゴリカル分布パラメータ  $\pi'_d$  をモデリングする。

$$\pi'_d = \text{Softmax}(\text{Linear}(\mathbf{h}_d)) \quad (2)$$

$$\mathbf{h}_d = \text{LSTM}([\mathbf{v}; \text{Linear}(\hat{\mathbf{c}}_{d-1})], \mathbf{h}_{d-1}) \quad (3)$$

ここで  $\mathbf{v}$  は単語分散表現， $;$  はベクトルの連結，Linear は線形変換層を表す。

$\hat{\mathbf{c}}_{d-1}$  は， $d-1$  桁目の値の one-hot vector を連続近似したものである。訓練時は Gumbel-Softmax Trick

### Algorithm 1 単語ペアの関係を判定する関数

**Require:**  $(C^s, C^t)$ : 候補  $s$  および候補  $t$  の階層コード

**Ensure:** 単語ペアの関係。1:包含, 0:一致, -1:その他

```
function RELATION( $C^s, C^t$ )
  for  $C_d^s, C_d^t$  in zip( $C^s, C^t$ ) do
    if  $C_d^s = 0 \wedge C_d^t \neq 0$  then
      return 1
    else if  $C_d^s = C_d^t \wedge C_d^s \neq 0 \wedge C_d^t \neq 0$  then
      continue
    else if  $C_d^s = 0 \wedge C_d^t = 0$  then
      return 0
    else
      return -1
  end if
end for
return 0
end function
```

(式中 GS) による擬似サンプリング (ただし誤差逆伝搬はしない: 式中 detach), 推論時はカテゴリカル分布パラメータをパススルーで与える。

$$\hat{\mathbf{c}}_{d-1} = \begin{cases} \text{detach}(\text{GS}(\pi_{d-1})) & \text{訓練時} \\ \pi_{d-1} & \text{推論時} \end{cases} \quad (4)$$

最後に，式 1 に従って  $\pi'_d$  を補正し，条件付確率  $P(C_d|C_{<d}) = \pi_d$  を求める。

$$\begin{aligned} \pi_{d,0} &= \pi_{d-1,0} + (1 - \pi_{d-1,0})\pi'_{d,0} \\ \pi_{d,>0} &= \frac{(1 - \pi_{d,0})}{(1 - \pi'_{d,0})}\pi'_{d,>0} \end{aligned} \quad (5)$$

### 3.3 上位下位関係の計量

上位下位関係の計量は，単語分散表現ペア  $(\mathbf{v}^s, \mathbf{v}^t)$  を変換した階層コード  $(C^s, C^t)$  を用いる。ふたつの階層コードを比較すると，その結果はかならず，包含・一致・その他のいずれかになる。上位下位関係の定義 3.1 より，判定関数 RELATION は Algorithm 1 として実装できる。したがって上位下位関係にある確率  $P(s < t)$  は，判定関数が包含 (= 1) を返す期待値となる。同様に，非上位下位関係の確率  $P(s \neq t)$  は，判定関数がその他 (= -1) を返す期待値となる。

$$P(s < t) = \mathbb{E}_{P(C^s), P(C^t)} [\text{RELATION}(C^s, C^t) = 1] \quad (6)$$

$$P(s \neq t) = \mathbb{E}_{P(C^s), P(C^t)} [\text{RELATION}(C^s, C^t) = -1]$$

ただし実際には全桁の同時分布の期待値は扱えないため，式 4 の擬似サンプルで近似的に計算する。

### 3.4 目的関数

変換器 (エンコーダ) を最適化する際の目的関数は，上位下位関係の識別，再構築損失，非ゼロ桁数

と分散表現の相互情報量の重み付き和とする。原理的には上位下位関係の識別のみを最適化すればよいが、再構築損失および相互情報量を補助的に併用することで、コードの分散を促す [10, 11].

$$L = L_h + \alpha L_{\text{reconst}} + \beta L_{\text{mi}} \quad (7)$$

上位下位関係の識別に対する目的関数  $L_h$  は、クロスエントロピー誤差として定義する。

$$L_h = \sum_{s_i, t_i, y_i \in \mathbb{H}^+ \cup \mathbb{H}^-} y_i P(s < t) + (1 - y_i) P(s \not< t) \quad (8)$$

ただし  $\mathbb{H}^+ = (s_i, t_i, y_i = 1)$  は、語彙知識に含まれる上位下位語ペアである。また  $\mathbb{H}^- = (s_i, t_i, y_i = 0)$  は、 $\mathbb{H}^+$  から機械的に生成した非上位下位語ペアである (3.5 節で述べる)。

再構築損失に対する目的関数  $L_{\text{reconst}}$  は、階層コードから再構築した分散表現と、オリジナルの分散表現との L2 誤差として定義する。

$$L_{\text{reconst}} = \sum_{i \in \mathbb{V}} \|\hat{v}_i - v_i\| \quad (9)$$

$$\hat{v} = \frac{\|\mathbf{v}\|}{\|\hat{\mathbf{v}}'\|} \hat{\mathbf{v}}', \quad \hat{\mathbf{v}}' = \sum_{d=0}^{N-1} \sum_{a=0}^{M-1} \hat{c}_{d,a} \mathbf{e}_{d,a} \quad (10)$$

ただし  $\mathbb{V}$  は、単語分散表現の語彙である。また  $\mathbf{e}_{d,a}$  は、 $d$  桁目の値  $a$  に割り当てた基底ベクトルである。基底ベクトルは他のモデルパラメータと同様に、訓練時に最適化する。

非ゼロ桁数と分散表現の相互情報量に対する目的関数  $L_{\text{mi}}$  は、階層コードから求まる非ゼロ桁数と、単語分散表現との相互情報量として定義する。

$$L_{\text{mi}} = -I(S; V), \quad S = \text{length}(\hat{C}) \quad (11)$$

ただし  $\text{length}(\cdot)$  は、非ゼロ桁数を返す関数である。

### 3.5 非上位下位語ペア（負例）の生成

語彙知識から得られるのは上位下位語ペアのみである。そこで順序反転および乱択を用いて、個別の上位下位語ペアから 5 つの非上位下位語ペアを生成する。順序反転は単語順序の交換、乱択は片方の単語をランダムサンプリングした単語と交換する操作である。乱択は、30% の確率で分散表現の最近傍 100 語から、70% の確率で全語彙からサンプリングする。非上位下位語ペアの生成例を表 1 に示す。

## 4 実験

先行研究 [5, 6] に倣い、分類タスク 3 種類：BLESS-hyponymy, WBLESS, BIBLESS および、ラン

表 1 非上位下位語ペアの生成

操作	上位語 ( $s$ )	下位語 ( $t$ )
正例	animal	dog
順序反転	dog	animal
上位語乱択	captain	dog
上位語乱択+順序反転	dog	captain
下位語乱択 (最近傍)	animal	dog_food
下位語乱択+順序反転	dog_food	animal

キングタスク 1 種類：HyperLex を用いて提案手法を評価する。これらの 4 つのタスク詳細および評価方法を付録 B に掲載した。また、提案手法のモデルの学習における詳細な実験設定を付録 C に掲載した。

表 2 に、提案手法の性能および先行研究との比較を示す。提案手法は、分類タスクにおいて高い性能を示し、特に BLESS-hyponymy および BIBLESS では先行研究の最高精度をそれぞれ 1.4 ポイントおよび 0.6 ポイント上回った。Nguyen ら [5] は本研究とよく似た語彙知識 (WordNet から抽出した上位下位関係) を実験に用いているが、彼らの手法をベースラインとすると、提案手法は 5 から 7 ポイントの性能改善を示している。また Vulic ら [6] の実験は上位下位に加えて同義・対義関係も利用しているが、提案手法は彼らと同程度あるいは 1 から 2 ポイントの性能改善を示している。これらの結果は、語彙知識の量が同等ならば、提案手法は語彙知識を活用する効率が高いことを示唆している。

一方、ランキングタスク (HyperLex) では提案手法は先行研究の最高精度を 10 ポイント以上下回った。この要因としては、最適化に用いる目的関数の影響が考えられる。提案手法は上位下位関係識別のクロスエントロピー誤差を最小化するため、正例は 1、負例は 0 に近づけようとする。一方で Athiwaratkun ら [8] や Vulic らの手法ではヒンジ損失を最小化するため、マージン以下の誤差は許容される。また Vulic らの実験は、前述のとおり同義・対義関係も用いることに留意が必要である。

## 5 考察

### 5.1 分類タスクの誤り分析

WBLESS および BIBLESS の負例は、同位・全体—部分など様々な意味関係の単語ペアから構成されている。意味関係ごとに評価した性能を、表 3 に示す。提案手法は、全体—部分関係および部分—全体関係の正解率が相対的に低いことがわかる。すなわち提案手法は (fox,mouth) や (radio,wire) のような単

表 2 提案手法の性能および先行研究との比較. 提案手法は 5 回の試行による平均および標準偏差 (カッコ内の数値) を報告. 太字は先行研究の精度を上回る事例.

手法	語彙資源	訓練に用いる意味関係	BLESS-hyponymy	WBLESS	BIBLESS	HyperLex
Nickel+[13]	WordNet	上位下位	-	0.86	-	0.512
Athiwaratkun+[8]	WordNet	上位下位	-	-	-	0.59
Dash+[14]	Hearst Patterns	上位下位	0.97	0.91	0.87	-
Nguyen+[5]	WordNet	上位下位	0.92	0.87	0.81	0.54
Vulic+[6]	WordNet, Roget	上位下位・同義・対義	0.96	0.92	0.88	0.686
提案手法	WordNet	上位下位	<b>0.984</b> (0.004)	0.919 (0.004)	<b>0.886</b> (0.007)	0.539 (0.012)

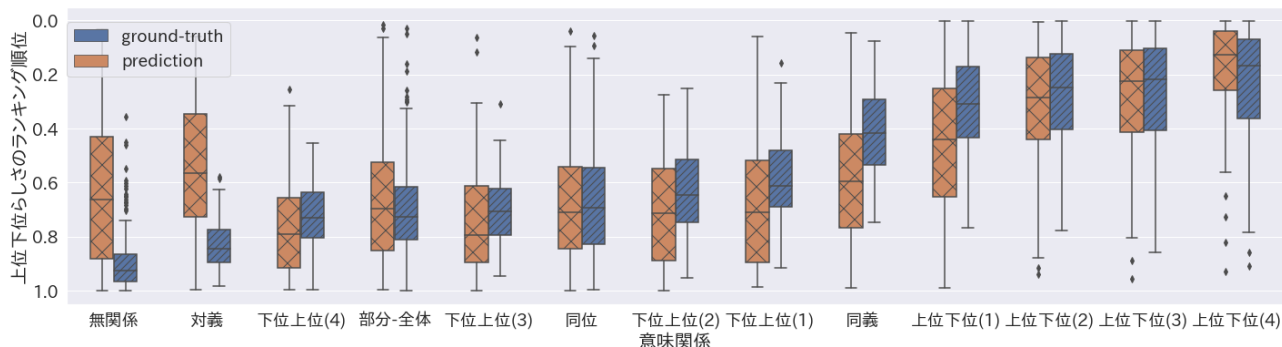


図 2 ランキングタスクの意味関係ごとの順位分布. 順位は全サンプル数で正規化. 意味関係のカッコ内の数値は上位語・下位語間のホップ数を表す

表 3 分類タスクの意味関係ごとの性能 (正解率)

意味関係	WBLESS	BIBLESS
全サンプル	0.919	0.886
上位-下位	0.901	0.893
下位-上位	0.976	0.907
同位	0.914	0.873
全体-部分	0.838	0.843
部分-全体	0.966	0.805
ランダム	0.930	0.915

語ペアを, 上位下位関係と誤認しやすいのである. なお WBLESS の部分—全体関係は精度が高いことから, 全体語を部分語よりも意味階層の上位だと捉える (=非ゼロ桁数を少なくする) ことはできているが, 誤って同一のコードを割り当ててていることが示唆される. この要因としては, 非上位下位語ペアの生成で全体—部分関係が出現しにくいことが考えられる. 乱択により生成される単語ペアは無関係または意味的類似性が高いかのいずれかだが, 全体—部分関係は必ずしもこれらに当てはまらない.

## 5.2 ランキングタスクの誤り分析

HyperLex の単語ペアは (xBLESS と同様に) 様々な意味関係の単語ペアから構成されている. 意味関係ごとにランキング順位の分布を可視化した結果を, 図 2 に示す. 提案手法は, 上位—下位・同義・下位—上位については, 平均的には正しい順序付け

ができていて, しばしば正解よりもばらつきが大きいことがわかる. また対義関係については, 上位下位らしさを過大評価していることがわかる. この要因としては, 分類タスクの誤り分析と同様に, 非上位下位語ペアの生成で対義関係が出現しにくいことが考えられる. Vulic らの実験のように, 対義関係を明示的な負例に用いることで, 性能改善の余地があるかもしれない.

## 6 まとめ

本研究では, 単語分散表現を階層コード表現に変換して, コードを比較して単語ペアの上位下位関係らしさを計量する手法を提案した. また提案手法は上位下位関係の分類タスクで既存手法を上回る性能であることを報告した. また誤り分析からは全体—部分関係および対義関係の誤差が大きいことがわかった. これらは手法そのものよりも, 非上位下位語ペアの与え方が要因になっている可能性がある. したがって, 提案手法をさらに多様なデータセットおよび語彙知識を用いて評価していきたい.

謝辞 本研究は JSPS 科研費 19H01118 の助成を受けた.

## 参考文献

- [1] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. *Recognizing Textual Entailment: Models and*

- Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2013.
- [2] Or Biran and Kathleen R. McKeown. Classifying taxonomic relations between pairs of wikipedia articles. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pp. 788–794, 2013.
- [3] Georgeta Bordea, Els Lefever, and Paul Buitelaar. Semeval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 1081–1091, 2016.
- [4] José Camacho-Collados. Why we have switched from building full-fledged taxonomies to simply detecting hypernymy relations. *CoRR*, Vol. abs/1703.04178, , 2017.
- [5] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 233–243, 2017.
- [6] Ivan Vulic and Nikola Mrksic. Specialising word vectors for lexical entailment. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1134–1145, 2018.
- [7] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.
- [8] Ben Athiwaratkun and Andrew Gordon Wilson. Hierarchical density order embeddings. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.
- [9] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. Smoothing the geometry of probabilistic box embeddings. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [10] Raphael Shu and Hideki Nakayama. Compressing word embeddings via deep compositional code learning. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.
- [11] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1558–1567, 2017.
- [12] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- [13] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30*, pp. 6338–6347, 2017.
- [14] Sarthak Dash, Md. Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, and Nicolas Rodolfo Fauceglia. Hypernym detection using strict partial order networks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 7626–7633.
- [15] Douwe Kiela, Laura Rimell, Ivan Vulic, and Stephen Clark. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 119–124, 2015.
- [16] Julie Weeds, Daoud Clarke, Jeremy Reffin, David J. Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pp. 2249–2259, 2014.
- [17] Ivan Vulic, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, Vol. 43, No. 4, 2017.
- [18] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhresch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.
- [19] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *CoRR*, Vol. abs/2010.01412, , 2020.

表 4 提案手法により得られた階層コード

単語	階層コード
animal	[2,2,2,2,2,2,2,0,0,0,0,0]
mammal	[2,2,2,2,2,2,2,4,0,0,0,0]
carnivore	[2,2,2,2,2,2,2,4,0,0,0,0]
cat	[2,2,2,2,2,2,2,4,4,5,1,1,7]
dog	[2,2,2,2,2,2,2,4,5,5,1,1,7]
mouse	[2,2,2,2,2,2,2,4,5,5,1,2,7]

## A 階層コードの実例

提案手法により得られた階層コードの一部を、表 4 に示す。提案手法を用いると、任意の単語分散表現を階層コードに変換して、上位下位関係を推論できるようにする。実際に単語ペアの上位下位関係を推論する際には各桁の条件付確率  $P(C_d|C_{<d})$  を用いる (3.3 節を参照) のだが、ここではわかりやすさのために  $\text{argmax}$  を取って離散化したものを示している。

## B 分類タスクおよび評価方法

分類タスクは、BLESS-hyponymy [15], WBLESS [16], BIBLESS の 3 種類を用いる。BLESS-hyponymy は上位下位語ペアのうちどちらが上位語かの 2 値分類, WBLESS は上位下位関係・その他の 2 値分類, BIBLESS は上位下位関係・下位上位関係・その他の 3 値分類の精度を評価する。推論は、式 6 で定義した上位下位関係にある確率  $P(s < t)$  を用いて行う。BLESS-hyponymy は  $P(s < t)$  と  $P(t < s)$  の大きい方を採択, WBLESS は  $P(s < t)$  をしきい値と比較, BIBLESS は最初に  $\max\{P(s < t), P(t < s)\}$  をしきい値と比較してから、 $P(s < t)$  と  $P(t < s)$  の大きい方を採択する。しきい値は validation data に最適化する。また validation/test split は、[5] と同様に、データセットからそれぞれ 2%, 98% を乱択する。評価指標は test data の accuracy である。validation/test split を 1,000 回実施して平均値を報告する。

ランキングタスクは、HyperLex [17] を用いる。与えられた単語ペアを上位下位関係らしさの高い順に順序付けして、アノテータが付与した順序との一致度を評価するタスクである。推論は  $P(s < t)$  の値が大きい順に順序付けする。評価指標はスピアマンの順位相関である。

## C 提案手法の学習

階層コードへの変換器 (エンコーダ) を訓練するための語彙知識および単語分散表現は、先行研究 [5] に倣って、WordNet から抽出した上位下位関係および、事前学習済み fastText モデルを用いる。

上位下位関係は、WordNet 上で直接・間接の "is-a" 関係にあるすべてのレンマのペアを抽出する。そのうえで、評価タスクのデータセット (xBLESS および HyperLex) と重複するペアを削除する。サンプル数は、名詞ペアが 2,158,824 件、動詞ペアが 162,706 件となった。

事前学習済み fastText モデルは、Mikolov ら [18] が配布するサブワード情報付きモデル<sup>3)</sup>を用いる。次元数は 300、語彙数は 100 万である。レンマから単語分散表現への変換は、大文字・小文字を区別する。また "dog\_food" のようにレンマが複数の単語からなる場合は、全単語の算術平均を取る。

目的関数の最適化は、ミニバッチによる確率勾配法を用いる。 $L_h$  のミニバッチサンプル数は、正例 200 件・負例 1,000 件である。 $L_{\text{reconst}}, L_{\text{mi}}$  のミニバッチサンプル数は、1,000 件である。最適化アルゴリズムは Sharpness-aware Minimization Optimizer [19] を用いる。式 7 で示した目的関数の重み付き和は  $\alpha = 5.0, \beta = 0.05$  とする。階層コードの容量は、8 進 16 桁とする。

3) <https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M-subword.vec.zip>