情報処理学会論文誌　データベース　Vol.14　No.4　8–23 (Oct. 2021)

# 階層コード表現学習による上位下位関係の識別

水木 栄[1,a)]　岡崎 直観[1,b)]

受付日 2021年3月4日, 採録日 2021年7月2日

**概要**：単語を，意味的な上位下位関係を保ちながら離散コードで表すような表現学習は可能だろうか？具体的には，上位語と下位語は先頭桁が一致するようにしつつ，M 進 N 桁のコードで表す単語埋め込みを獲得することは可能だろうか．もしもそのような表現が獲得できるならば，コードを比較するだけで単語間の上位下位関係が推論できるはずである．このような動機に基づき，本研究では，分散表現を階層性のあるコード表現に変換するアーキテクチャと，任意のコード対の包含関係を微分可能な形式で計量する計算方法を提案する．具体的には，語彙資源から得られる大量の上位下位語ペアを教師信号とすることにより，単語間の上位下位関係を考慮しつつ，単語分散表現を，連続緩和を適用したコード表現へ変換する統計モデルを学習する仕組みを提示する．提案手法により得られたコード表現を用いていくつかの上位下位識別タスクを解いたところ，上位下位関係の分類タスクにおいて，Semantic Specialization や Order Embeddings に基づく既存手法を上回る性能を達成した．また提案手法の有効性分析を実施し，コード表現から元の分散表現を復元する補助タスクや，意味的類似性の高い非上位下位語ペアを負例に用いることが性能向上に寄与することを確認した．

**キーワード**：コード学習，表現学習，上位下位関係識別，Semantic Specialization，Order Embeddings

# Learning Hierarchical Code Representation for Hypernymy Detection

Sakae Mizuki[1,a)]　Naoaki Okazaki[1,b)]

Received: March 4, 2021, Accepted: July 2, 2021

***Abstract:*** Is it possible to learn the word representations using taxonomically consistent discrete codes that reflects the hypernymy relation? Specifically, is it possible to represent any word as an M-ary N-digit code while the hypernym and hyponym share the first n digits? Once such a representation is learned, hypernymy relation detection can be done by simply comparing the codes of word pairs. Motivated by this concept, we propose an architecture for transforming distributed representations into hierarchical code representations and a differentiable metric that quantifies the degree of inclusion relation of an arbitrary pair of codes. Concretely, we present a methodology for training an encoder that transforms any word embeddings into the code representations with continuous relaxation while considering hypernymy relations among words using large-scale hypernym-hyponym word pairs extracted from the lexical resource as a supervision signal. Accordingly, we applied the learned hierarchical codes to solve the hypernymy relation detection tasks. As a result, we confirmed that the proposed method outperformed existing methods based on Semantic Specialization and Order Embeddings in the hypernymy classification tasks. We also conducted an ablation study and confirmed that the auxiliary task of reconstructing the input word embeddings from the codes and negative sampling using non-hypernymy but semantically similar word pairs contributed to the performance improvement.

***Keywords:*** code learning, representation learning, hypernymy detection, Semantic Specialization, Order Embeddings

## 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 [6] やテキスト生成 [2] など幅広いタスクで用いられる．

[1] 東京工業大学
Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
[a)] sakae.mizuki@nlp.c.titech.ac.jp
[b)] okazaki@c.titech.ac.jp

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 [4]．こうしたなか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 [5]．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．

ところで Word2Vec [20] に代表される単語分散表現は，ニューラル自然言語処理の基本要素として幅広く利用されている．分散表現は一般に数百次元のベクトルだが，コード学習の手法を用いることで，ベクトル空間上での類似性を保ったまま M 進 N 桁の離散コードに教師なし学習で変換できることが報告されている [25]．たとえば cat を $(3,7,1,4)$，dog を $(3,7,4,4)$ に変換する $\cdots$ といった具合である．

このようなコード学習の手法を発展させて，分散表現を階層性のあるコード表現に変換することはできないだろうか？　たとえばシラバスの授業コードは，先頭桁を見れば授業の上位概念である学年や学部などが分かるようにコードが体系化されている．同様の発想で，上位語と下位語は先頭桁が一致するような変換を実現できないだろうか．たとえば animal は cat および dog の上位語なので，これらと先頭桁が一致するように $(3,7,0,0)$ に変換する，といった具合である．階層性のあるコード表現の世界では，上位下位関係はコードの包含関係に帰着される．したがって正確な変換規則を作ることができれば，もはや特別な識別器を用意せずとも，関心のある単語のコードを比較するだけで上位下位関係の有無が識別できるはずだ．

問題は，このような都合の良い（animal のコードが dog や cat のコードを包含するような）変換規則を，人間による明示的な体系化が与えられずとも機械的に構築できるか，ということである．人手で 1 つ 1 つコードを割り当てていくことは現実的ではないが，WordNet [8] を参照したり，上位下位関係を抽出するルールベース手法（Hearst Pattern [10]）をコーパスに適用 [23] すれば，上位下位語ペア，たとえば (animal, cat) や (animal, dog) といった単語ペアが大量に入手できる．こうした語彙知識を教師信号として変換器の機械学習に用いれば，コード体系を明示的に与えずとも適切な変換規則を獲得し，語彙知識に含まれない単語ペアの上位下位関係を高精度に推論できるのではないだろうか？

本論文で提案するのは，語彙資源から得られる大量の上位下位語ペアを教師信号とすることにより，学習ずみ単語分散表現を階層性のあるコード表現に変換するという表現学習の手法である．また，連続緩和したコード表現を用いることにより，単語ペアの上位下位関係を微分可能な形式で計量する方法を提示する．提案手法の模式図を**図 1** に示すとともに，実際に提案手法により得られた階層コードの具体例を**表 1** に示す．また実験を通じて，提案手法が上位下位識別タスクの一部で既存手法を上回ったことを報告する．

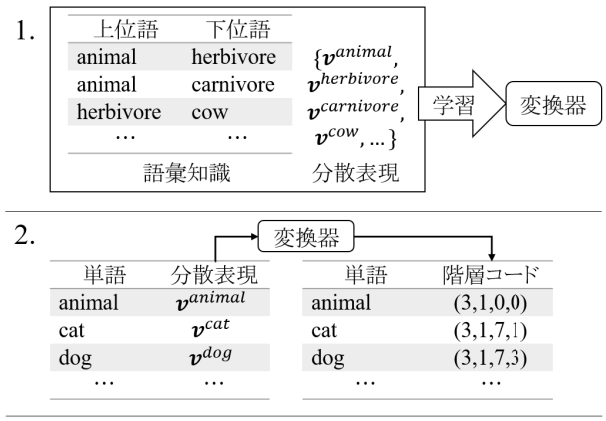


**図 1**　提案手法の模式図．1. 語彙知識と分散表現を用いて変換器を学習，2. 学習した変換器を用いて分散表現を階層コード表現に変換，3. 階層コード間の包含関係を用いて上位下位関係を識別

**Fig. 1**　Overall schema of the proposed method. 1. Train the encoder using lexical knowledge and word embeddings. 2. Convert the word embeddings into hierarchical code representations using the trained encoder. 3. Detect hypernymy relation based on the inclusion relation between the hierarchical code pair.

**表 1**　提案手法により得られた 8 進 16 桁の階層コード[*1]．任意の単語分散表現を階層コードに変換できる

**Table 1**　The 16-digit 8-ary hierarchical codes obtained by the proposed method. Proposed method enables transformation of any word embeddings into hierarchical codes.

| 単語 | 階層コード |
|---|---|
| animal | (2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0) |
| mammal | (2,2,2,2,2,2,2,4,0,0,0,0,0,0,0,0) |
| carnivore | (2,2,2,2,2,2,2,4,0,0,0,0,0,0,0,0) |
| cat | (2,2,2,2,2,2,2,4,4,5,1,1,7,4,2,2) |
| dog | (2,2,2,2,2,2,2,4,5,5,1,1,7,4,4,2) |
| mouse | (2,2,2,2,2,2,2,4,5,5,1,2,7,4,2,2) |

本研究の提案手法は，Semantic Specialization および Order Embeddings [26] の長所を統合統合する手法とも位置づけられる．近年の研究では，語彙知識を用いた単語埋め込みの表現学習を行う方法として，Semantic Specialization および Order Embeddings の 2 つが注目を集めている．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，精度だけでなく，推論結果が推移性や反対称性といった望ましい性質を満たすことが重要だと指摘されている [5]．精度については，単語分散表現による意味的類似性

[*1] 実際の計算では離散コードではなく連続緩和したコードを用いる（3.3 節を参照）のだが，ここでは分かりやすさのために各桁の argmax をとって離散化したコードを示す．

を活用できる Semantic Specialization がこれまでの最高精度を報告している [28]．一方で推移性や反対称性については，順序関係を定義可能な Order Embeddings が有望である．こうしたなかで，両者を兼ね備えた手法はいまだ提案されていない．これに対して本研究の提案手法は，単語分散表現を変換してコード表現を得ることにより Semantic Specialization の長所を，包含関係を定義可能な階層性のあるコード表現を用いることにより Order Embeddings の長所を取り入れようとする手法である．

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アーキテクチャの提案．
- コードのペアに対して，包含関係の程度を微分可能な形式で計量する計算方法の提案．これにより，誤差逆伝搬による最適化を可能にしたこと．
- 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

本研究における上位下位識別のタスク設定は，WordNet のような大規模な語彙資源から得られる知識，特に上位下位語ペアの集合を教師信号とする表現学習である．すなわち単語分散表現のコサイン類似度を直に用いて意味的類似度を推論するのと同様に，獲得した表現を（識別器の特徴量として用いるのではなく）直に用いて上位下位関係を推論するという設定である．同様のタスク設定に従う既存研究は，Order Embeddings，Semantic Specialization 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．

Order Embeddings [26] 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[*2]および反対称性[*3]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 [1] や超直方体 [17] などを用いる手法が提案されている．Order Embeddings の短所は，表現学習の対象が語彙資源により制約されることである．各単語の埋め込み表現は語彙資源から得られる上位下位語ペアを用いて学習されるため，語彙資源に含まれない単語については表現が与えられないし，上位下位関係を推論することもできない．これに対して提案手法の相違点は，埋め込み表現（すなわち，階層コード）を直接学習するのではなく，単語分散表現を埋め込み表現に変換する方法を学習することである．このため提案手法は語彙資源により制約されず，原理的には分散表現が所与の単語すべてを扱うことができる．

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 [21]．Vulic ら [28] はこの表現方法にのっとり，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これまでの最高精度を報告している．これに対して本研究の相違点は，単語分散表現を，順序関係を定義可能な表現形式に変換する手法を提案することである．ベクトル空間上でのノルム長さおよびコサイン類似度は推移性および反対称性を必ずしも満たさないが，提案手法が用いる階層コード上の包含関係は，これらを満たすことができる．

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 [25]，クラスタリング [11]，および類似文書検索 [24], [32] に応用されている．一方で，本研究のように上位下位語ペアを教師信号として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．

本研究のタスク設定とは異なるが，上位下位識別を教師あり分類タスクとして解くアプローチも存在する．主な特徴量として用いられるのは Word2Vec [20] や fastText [3] などの単語分散表現である．上位下位関係は意味関係の一種であるため，意味的類似度をとらえる単語分散表現は有効な特徴量だという期待がある．しかし実際には，教師あり学習の枠組みで単語分散表現を特徴量とするアプローチは，単語間の関係ではなく上位語らしさを記憶する傾向があるため，汎化性能に問題があると報告されている（Lexical Memorization [16]）．このため近年の研究では，意味関係を考慮して写像した分散表現を用いるといった複雑な手法が提案されている [29]．

## 3. 提案手法

分散表現を階層コードに変換する変換器を訓練する手法を提案する．手法の概要は以下のとおりである．

(1) 分散表現を変換器に入力して，階層コードの確率分布を出力する．
(2) Gumbel-Softmax Trick を用いて，確率分布からサンプリングする．これにより，各桁の one-hot vector を連続緩和した形式で，階層コードのサンプルを得る．
(3) 得られたサンプルを用いて，上位下位識別，再構築損失，および相互情報量からなる目的関数を計算する．上位下位識別は，上位下位語ペアを正例，非上位下位語ペアを負例とする二値分類である．

[*2] $s \prec t$ かつ $t \prec u$ ならば $s \prec u$ が成立する性質のこと．
[*3] $s \prec t$ ならば $t \not\prec s$ が成立する性質のこと．

(4) 目的関数に対するモデルパラメータの勾配を計算する．
(5) 勾配を用いて，変換器のモデルパラメータを更新する．

手法の特徴は，本来の問題を連続緩和して扱うことである．本来の階層コードは，各桁が離散値で表される．しかし離散値のままでは，勾配降下法による最適化が困難である．そこで変換器の訓練および上位下位関係の推論を行うための計算過程では，つねに各桁の one-hot vector を連続緩和した形式を用いて近似計算を行う．すなわち提案手法は，離散コードをサンプリングするのではなく連続緩和したコードを出力し，各桁は互いに独立に，連続緩和したコードを分布パラメータとするカテゴリカル分布に従うようにモデル化する．

以下，それぞれの計算を詳述する．

### 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，ただしひとたび 0 が出現したら後続桁はすべて 0 となる離散ベクトルである．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．ここから直感的に分かるように，階層コードを用いて包含関係を定義できる．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) 双方ともに非ゼロのすべての桁について，値が一致する関係であると定義すればよい．たとえば $((3,1,0,0),(3,1,7,0))$ は包含関係である．

### 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 $\boldsymbol{v}$ を $M$ 進 $N$ 桁の階層コード $\boldsymbol{C}=(C_0,C_1,\ldots,C_{N-1})$ の確率分布 $P(\boldsymbol{C})$ に変換する関数である．階層コードでは，上位桁が下位桁に影響を及ぼす．そこで LSTM による再帰的計算および Gumbel-Softmax Trick [18] によるサンプリングを用いて，下位桁の確率分布が上位桁の値に依存する関数をモデリングする．アーキテクチャの模式図を**図 2** に掲載する．

$d$ 桁目の階層コードを表す確率変数およびその値を $C_d$ および $a\in\{0,1,\ldots,M-1\}$ とする．ひとたび 0 が出現したら後続桁はすべて 0 となるという制約より，上位 $d-1$ 桁 $\boldsymbol{C}_{<d}$ を所与とした $C_d$ の条件付分布は

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

と表される．ここで $\mathbb{1}_{\{.\}}$ は指示関数を表す．また右辺第 1 項および第 2 項はそれぞれ，直前桁がゼロの場合および非ゼロの場合に対応している．

直前桁が非ゼロの場合の確率分布 $P(C_d=a|C_{d-1}\neq 0,\boldsymbol{C}_{<d})$ のカテゴリカル分布パラメータ $\boldsymbol{\pi}'_d\in\mathbb{\Delta}^{M-1}$（ただし $\mathbb{\Delta}^{M-1}$ は $M-1$ 次元の単体を表す[*4]）は，LSTM を用いてモデリングする．LSTM の入力は，直前桁の値および，変換対象の単語分散表現である．

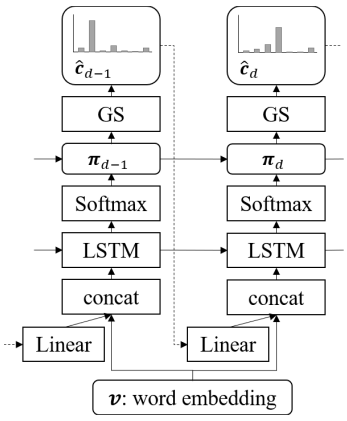


**図 2**　変換器のアーキテクチャ．角丸は変数，直角は関数，点線は誤差逆伝搬しないことを表す．記号の定義は本文を参照

**Fig. 2**　Encoder architecture. Rectangle indicate functions, Rectangle with rounded corners indicate variables, and dotted lines indicate no back propagation. Refer to the body text for definitions of symbols.

$$\boldsymbol{\pi}'_d=\mathrm{Softmax}(\mathrm{Linear}(\boldsymbol{h}_d))\tag{2}$$

$$\boldsymbol{h}_d=\mathrm{LSTM}([\boldsymbol{v};\mathrm{Linear}(\mathrm{detach}(\hat{\boldsymbol{c}}_{d-1}))],\boldsymbol{h}_{d-1})\tag{3}$$

ここで $\boldsymbol{v}$ は単語分散表現，$\boldsymbol{h}_d$ は $d$ 回目の LSTM の隠れ状態ベクトル，; はベクトルの連結，Linear は線形変換層を表す．ただし線形変換層への入力については，誤差逆伝搬をしない[*5]（式中 detach）．

$\hat{\boldsymbol{c}}_{d-1}\in\mathbb{\Delta}^{M-1}$ は，$d-1$ 桁目の値の one-hot vector を連続緩和したものである．すなわち $\hat{\boldsymbol{c}}_{d-1}$ は M 次元ベクトルであり，$a$ 番目の要素は $d-1$ 桁目が $a$ をとる割合を示している．したがって $0\leq\hat{c}_{d-1,a}\leq 1$ かつ $\sum_{a=0}^{M-1}\hat{c}_{d-1,a}=1$ を満たす．

$\hat{\boldsymbol{c}}_{d-1}$ の計算は，訓練時は Gumbel-Softmax Trick（式中 GS）によるサンプリング，推論時はカテゴリカル分布パラメータをそのまま用いる．

$$\hat{\boldsymbol{c}}_{d-1}=\begin{cases}\mathrm{GS}(\boldsymbol{\pi}_{d-1}) & \text{訓練時}\\ \boldsymbol{\pi}_{d-1} & \text{推論時}\end{cases}\tag{4}$$

最後に，式 (1) に従って $\boldsymbol{\pi}'_d$ を補正し，条件付確率 $P(C_d=a|C_{<d})$ のカテゴリカル分布パラメータ $\boldsymbol{\pi}_d\in\mathbb{\Delta}^{M-1}$ を求める．

[*4] $M-1$ 次元の単体とは，すべての要素がゼロ以上かつ，かつ要素の合計が 1 をとる $M$ 次元の実数ベクトルの集合である．Softmax 関数の出力，カテゴリカル分布のパラメータ，および Gumbel-Softmax Trick の出力はすべて，単体上の点である．

[*5] 予備実験において，誤差逆伝搬を有効にすると学習が不安定になることを確認したため．なお上位桁への誤差逆伝搬は $\boldsymbol{h}_d$ および $\boldsymbol{\pi}'_d$ を経由する．

**Algorithm 1** 階層コードの関係を判定する関数

**Require:** $(\boldsymbol{C}^s, \boldsymbol{C}^t)$: 候補 $s$ および候補 $t$ の階層コード
**Output:** コードペアの関係．1:包含, 0:一致, −1:その他
  **function** RELATION$(\boldsymbol{C}^s, \boldsymbol{C}^t)$
    **for** $d = 0$ to $N-1$ **do**
      **if** $C^s_d = 0 \wedge C^t_d \neq 0$ **then**
        **return** 1
      **else if** $C^s_d = C^t_d \wedge C^s_d \neq 0 \wedge C^t_d \neq 0$ **then**
        **continue**
      **else if** $C^s_d = 0 \wedge C^t_d = 0$ **then**
        **return** 0
      **else**
        **return** −1
      **end if**
    **end for**
    **return** 0
  **end function**

$$\pi_{d,0} = \pi_{d-1,0} + (1 - \pi_{d-1,0})\pi'_{d,0} \tag{5}$$

$$\boldsymbol{\pi}_{d,>0} = \frac{(1-\pi_{d,0})}{(1-\pi'_{d,0})}\boldsymbol{\pi}'_{d,>0} \tag{6}$$

ベクトルの添字 $d,>0$ は，先頭の次元を除くことを表す．

以上により，階層コードの確率分布 $P(\boldsymbol{C})$ は

$$P(\boldsymbol{C}) = \prod_{d=0}^{N-1} \mathrm{Cat}(C_d; \boldsymbol{\pi}_d) \tag{7}$$

と定義される．ただし $\mathrm{Cat}(;\boldsymbol{\pi})$ は，$\boldsymbol{\pi}$ をパラメータとするカテゴリカル分布である．また $\boldsymbol{\pi}$ は連続緩和したコードでもある．なお訓練時には Gumbel-Softmax Trick を適用するため，$\boldsymbol{\pi}$ は確率的であることに注意されたい．推論時は確定的である．

### 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 $(\boldsymbol{v}^s, \boldsymbol{v}^t)$ を変換した階層コード $(\boldsymbol{C}^s, \boldsymbol{C}^t)$ を用いる．ただし $s$ および $t$ はそれぞれ，上位語候補および下位語候補を表す．2 つの階層コードを比較すると，その関係はかならず，包含・一致・その他のいずれかに分類できる[*6]．3.1 節で述べた包含関係の定義をアルゴリズムの形式で書き下したものが，Algortithm 1 に示す判定関数 RELATION である．たとえば包含（= 1）を返す If ブロックの条件式：$C^s_d = 0 \wedge C^t_d \neq 0$ は，3.1 節で述べた定義文 “上位より下位のほうが非ゼロ桁数が多く” に対応している．

判定関数 RELATION を用いて，任意の単語ペアに対して上位下位関係の計量ができる．具体的には，上位下位関係にある確率 $P(s \prec t)$ は，RELATION 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 $P(s \not\prec t)$ は，RELATION がその他（= −1）を返す期待値となる．

[*6] 本研究では使用しなかったが，原理的には同位関係の分類も可能である．同位関係を扱うことはタクソノミの自動構築において重要と考えられるため，今後の研究課題としたい．

$$P(s \prec t) = \mathop{\mathbb{E}}_{P(\boldsymbol{C}^s),P(\boldsymbol{C}^t)}[\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = 1] \tag{8}$$

$$P(s \not\prec t) = \mathop{\mathbb{E}}_{P(\boldsymbol{C}^s),P(\boldsymbol{C}^t)}[\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = -1] \tag{9}$$

ただし $P(\boldsymbol{C}^s)$ および $P(\boldsymbol{C}^t)$ は，式 (7) により定義された，上位語候補 $s$ および下位語候補 $t$ の階層コードが従う確率分布である．

実際には，訓練時には $P(\boldsymbol{C})$ の分布パラメータ $\{\boldsymbol{\pi}_d\}_{d=0}^{N-1}$ は確率的であるため，上述の期待値は解析解を計算できない．このため，変換器に含まれるモデルパラメータの勾配を求めることができない．また階層コードの実現値を離散値の形式で（連続緩和せずに）サンプリングしてモンテカルロ近似する方法も，計算コストや勾配の推定精度に問題がある．そこで Variational AutoEncoder [15] の方法論にならって，式 (4) で得られるサンプル $\hat{\boldsymbol{C}} = (\hat{\boldsymbol{c}}_0, \hat{\boldsymbol{c}}_1, \ldots, \hat{\boldsymbol{c}}_{N-1})$ を用いて近似計算する．具体的には，各桁が独立かつ，$\hat{\boldsymbol{c}}_d$ をパラメータとするカテゴリカル分布 $\mathrm{Cat}(C_d; \hat{\boldsymbol{c}}_d)$ を用いて

$$\hat{P}(\boldsymbol{C}) = \prod_{d=0}^{N-1} \mathrm{Cat}(C_d; \hat{\boldsymbol{c}}_d) \tag{10}$$

と近似する．そのうえで，式 (8) の $P(\boldsymbol{C}^s)$ および $P(\boldsymbol{C}^t)$ をそれぞれ $\hat{P}(\boldsymbol{C}^s)$ および $\hat{P}(\boldsymbol{C}^t)$ で置き換えて，期待値を計算する．

$$P(s \prec t) \approx \mathop{\mathbb{E}}_{\hat{P}(\boldsymbol{C}^s),\hat{P}(\boldsymbol{C}^t)}[\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = 1] \tag{11}$$

$$P(s \not\prec t) \approx \mathop{\mathbb{E}}_{\hat{P}(\boldsymbol{C}^s),\hat{P}(\boldsymbol{C}^t)}[\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = -1] \tag{12}$$

$P(\boldsymbol{C})$ を $\hat{P}(\boldsymbol{C})$ で置き換えて期待値を計算することは，$P$ の分布パラメータを 1 個の Gumbel-Softmax Trick のサンプルで近似していることになる．

訓練時の $\hat{P}(\boldsymbol{C})$ および，推論時の $P(\boldsymbol{C})$ は，いずれも各桁が独立である．このため判定関数 RELATION の期待値は解析解が計算できる．以下では計算方法の概要のみ述べる．詳細な計算式は付録 A.1.1 に示す．なお以下の説明では $\hat{\boldsymbol{c}}_d$ を分布パラメータとするが，推論時は $\boldsymbol{\pi}_d$ と読み替えればよい．

$d$ 桁目の値 $C_d$ は $\mathrm{Cat}(C_d; \hat{\boldsymbol{c}}_d)$ に従う．したがって，判定関数 RELATION の If 文は，成立・不成立の二値ではなく，成立する割合（確率）を実数として評価できる．たとえば包含（= 1）を返す If ブロックの条件式：$C^s_d = 0 \wedge C^t_d \neq 0$ が成立する割合は

$$P(C^s_d = 0 \wedge C^t_d \neq 0) = \hat{c}^s_{d,0}\left(\sum_{a \neq 0} \hat{c}^t_{d,a}\right) \tag{13}$$

と評価する．If 文は桁数ぶん反復する For ループで囲まれているため，判定関数 RELATION が包含（= 1）を返す割合は，$d-1$ 桁目までは For ループ継続（= continue）の条件式が成立し，$d$ 桁目ではじめて包含（= 1）の条件式が成立する割合を，すべての $d$ について合計した値となる．

ここまでに例示したとおり，サンプルを用いた近似計算のもとでは，解析解およびモンテカルロ近似の短所を回避しつつ，$P(s \prec t)$ および $P(s \not\prec t)$ についてモデルパラメータの勾配を計算することができる．

### 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とともに，分散表現空間上の類似性が階層コードに反映されやすくする [11], [25]．

$$L = L_{\mathrm{h}} + \alpha L_{\mathrm{reconst}} + \beta L_{\mathrm{mi}} \tag{14}$$

上位下位関係の識別に対する目的関数 $L_{\mathrm{h}}$ は，上位下位語ペアを正例（$y=1$），非上位下位語ペアを負例（$y=0$）とする二値分類に対するクロスエントロピー誤差として定義する．

$$L_{\mathrm{h}} = \sum_{(s,t,y)\in\mathbb{H}^+\cup\mathbb{H}^-} y \ln P(s \prec t) + (1-y)\ln P(s \not\prec t) \tag{15}$$

ただし $\mathbb{H}^+ = \{(s,t,y=1)\}$ は，語彙資源から抽出した上位下位語ペアの集合である．同様に $\mathbb{H}^- = \{(s,t,y=0)\}$ は，$\mathbb{H}^+$ から機械的に生成する非上位下位語ペアの集合である（生成方法については 3.5 節で述べる）．

再構築損失に対する目的関数 $L_{\mathrm{reconst}}$ 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．ただし分散表現の再構築には非ゼロ桁のみが寄与するように制約する．具体的には

$$L_{\mathrm{reconst}} = \sum_{w\in\mathbb{V}} ||\hat{\boldsymbol{v}}^w - \boldsymbol{v}^w|| \tag{16}$$

$$\hat{\boldsymbol{v}} = \frac{||\boldsymbol{v}||}{||\hat{\boldsymbol{v}}'||}\hat{\boldsymbol{v}}',\ \hat{\boldsymbol{v}}' = \sum_{d=0}^{N-1}\sum_{a=1}^{M-1} \hat{c}'_{d,a}\boldsymbol{e}_{d,a} \tag{17}$$

$$\hat{c}'_{d,a} = \hat{c}_{d,a}\left(\prod_{k=0}^{d-1}(1-\hat{c}_{k,0})\right) \tag{18}$$

と定義する[*7]．ただし $\mathbb{V}$ は，単語分散表現の語彙である．また $\boldsymbol{e}_{d,a}$ は，$d$ 桁目の値 $a(\neq 0)$ 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．

[*7] 上付き添字 $w$ が明らかな場合は添字を省略した．

**表 2**　非上位下位語ペアの生成．イタリック体は乱択された単語を表す

**Table 2**　Generation of the non-hyponymy word pairs. Italic face indicates the randomly sampled word.

| 操作 | 上位語 ($s$) | 下位語 ($t$) |
|---|---|---|
| 正例 | animal | dog |
| 順序反転 | dog | animal |
| 上位語を最近傍から乱択 | *dog_food* | dog |
| 上位語乱択+順序反転 | dog | *dog_food* |
| 下位語を全語彙から乱択 | animal | *captain* |
| 下位語乱択+順序反転 | *captain* | animal |

非ゼロ桁数と分散表現の相互情報量に対する目的関数 $L_{\mathrm{mi}}$ 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．

$$L_{\mathrm{mi}} = -\mathrm{I}(\mathrm{length}(\boldsymbol{C}); V) \tag{19}$$

$$P(V=\boldsymbol{v}) \approx \frac{1}{|\mathbb{V}|}\sum_{w\in\mathbb{V}} \mathbb{1}\{\boldsymbol{v}=\boldsymbol{v}^w\} \tag{20}$$

ただし length(.) は，階層コードの非ゼロ桁数を返す関数である．つまり $\mathrm{length}(\boldsymbol{C})$ は，$\{0,1,\ldots,N\}$ のいずれかをとる確率変数である．また $V$ は，単語分散表現を表す確率変数であり，経験分布で近似する．

相互情報量を求めるためには非ゼロ桁数および単語分散表現の確率分布を用いて期待値を計算する必要があるが，前者を解析的に求めることはできない．そこで上位下位関係の計量と同様に，式 (10) で定義した確率分布 $\hat{P}(\boldsymbol{C})$ を用いて近似的に計算する．詳細な計算式は付録 A.1.2 に示す．

### 3.5　非上位下位語ペア（負例）の生成

語彙資源から得られるのは上位下位語ペアのみ，つまり正例のみだが，負例がなくては変換器をうまく最適化することはできない．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[*8]．これにより，関係性に乏しい単語ペアだけでなく，意味的類似性は高いが上位下位関係ではない単語ペアも生成されるようにする．非上位下位語ペアの生成例を**表 2** に示す．

## 4.　実験

### 4.1　評価タスク・データセット・推論方法

先行研究 [21], [28] にならい，分類タスク 3 種類および，ランキングタスク 1 種類を用いて，上位下位識別タスクに

[*8] ただし，対象語の上位語および下位語は除外する．

表 3　提案手法の性能および先行研究との比較．提案手法は初期パラメータを変えて 5 回実験した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

**Table 3**　Performance of the proposed method and previous studies. The proposed method reports the mean and standard deviation (numbers in parentheses) of five trials with different model parameter initialization. Bold figures indicate the proposed method outperforms the best result of previous studies.

| 手法 | 語彙資源 | 訓練に用いる意味関係 | BLESS-hyponymy | WBLESS | BIBLESS | HyperLex |
|---|---|---|---|---|---|---|
| Nickel+ [22] | WordNet | 上位下位 | - | 0.86 | - | 0.512 |
| Athiwaratkun+ [1] | WordNet | 上位下位 | - | - | - | 0.59 |
| Dash+ [7] | Hearst Patterns | 上位下位 | 0.97 | 0.91 | 0.87 | - |
| Nguyen+ [21] | WordNet | 上位下位 | 0.92 | 0.87 | 0.81 | 0.54 |
| Vulic+ [28] | 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 | **0.984** (0.004) | 0.919 (0.004) | **0.886** (0.007) | 0.539 (0.012) |

おける提案手法の性能を評価する．

分類タスクは，BLESS-hyponymy [14]，WBLESS [30]，BIBLESS [14] 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 値分類を解くタスクである．推論は，式 (8) で定義した上位下位関係にある確率 $P(s \prec t)$ を用いて行う．BLESS-hyponymy は $P(s \prec t)$ と $P(t \prec s)$ を比較して大きい方を上位語と判定，WBLESS は $P(s \prec t)$ がしきい値以上ならば上位下位関係と判定，BIBLESS はまず $\max\{P(s \prec t), P(t \prec s)\}$ をしきい値と比較して，しきい値以上ならば $P(s \prec t)$ と $P(t \prec s)$ を比較して上位下位か下位上位かを判定する．しきい値は開発データに最適化する．また開発データ・テストデータ分割は [21] と同様に，データセットからそれぞれ 2%，98%にランダム分割する[*9]．評価指標はテストデータの正解率（accuracy）である．ただしデータセット分割におけるランダムネスの影響を排除するため，開発データ・テストデータ分割を 1,000 回実施して平均値を求める．

ランキングタスクは，HyperLex [27] を用いる．HyperLex は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 $P(s \prec t)$ 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．

### 4.2　学習方法

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 [21] にならって，WordNet から抽出した上位下位関係および，事前学習ずみ 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．たとえば (animal, cat)，(mammal, cat)，(carnivore, cat) などの単語ペアが得られる．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．

事前学習ずみ fastText モデルは，Mikolov ら [19] が配布するサブワード情報付きモデル[*10]を用いる．次元数は 300，語彙サイズは 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とる．

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．上位下位識別 $L_{\mathrm{h}}$ のミニバッチサンプル数は，正例 200 件・負例 1,000 件である．再構築損失 $L_{\mathrm{reconst}}$ および相互情報量 $L_{\mathrm{mi}}$ 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 [9] を用いる．Gumbel-Softmax Trick の温度パラメータは 1.0 とする[*11]．温度パラメータのスケジューリングは使用しない．式 (14) で示した目的関数の重み付き和は $\alpha = 5.0, \beta = 0.05$ とする．階層コードの進数 $M$ および桁数 $N$ は，8 進 16 桁とする．

### 4.3　実験結果

**表 3** 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た．特に BLESS-hyponymy および BIBLESS では，Dash らおよび Vulic らによる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．実験に用いられた語彙知識の違いを考慮に入れて比較しても，提案手法は先

[*9] 訓練データは必要としないため，開発データ・テストデータのみ用意すればよい．2 章で述べたとおり，本研究のタスク設定は，獲得した表現を直に用いて上位下位関係を推論するというものである．xBLESS データセットを用いて識別器を訓練するわけではない．

[*10] https://fasttext.cc/docs/en/english-vectors.html

[*11] 1.0 は，PyTorch:gumbel_softmax 関数のデフォルト値である．なお予備実験では，温度パラメータの影響は些少であった．

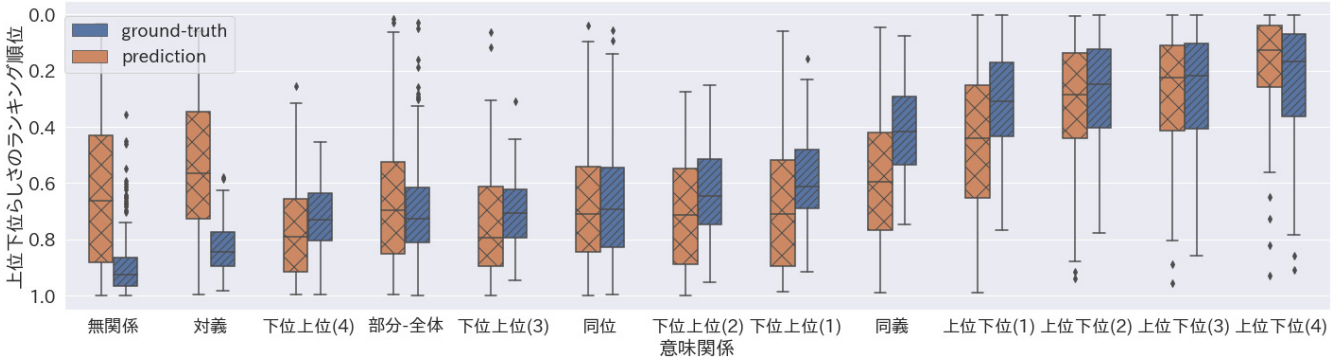


**図 3**　HyperLex データセットの意味関係ごとの順位分布．順位は全サンプル数で正規化．意味関係のカッコ内の数値は上位語・下位語間のホップ数を表す

**Fig. 3**　Distribution of the ranks grouped by each semantic relations for HyperLex task. The ranks are normalized by the total number of samples. The number in parentheses of semantic relations indicate the number of hops of hyponymy relations.

行研究に対して優位性のある結果になっている．具体的には，Nguyen ら [21] が実験で用いた語彙知識は本研究とほぼ同様であるが，彼らの手法と比較すると提案手法は 5 から 7 ポイントの性能改善を示している．また Vulic ら [28] が実験で用いた語彙知識は本研究よりも多い（Vulic らは上位下位だけでなく，同義・対義関係も用いている）が，彼らの報告と比較しても，提案手法は同程度あるいは 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，提案手法は与えられた語彙知識を表現学習に活用する効率が高いことを示唆している．

ランキングタスク（HyperLex）では，提案手法は Vulic らが報告した最高精度を 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 に極力近づけようとする．このため，上位下位関係らしさの推論結果は二極化しやすい．これに対して，Vulic らの手法ではヒンジ損失の最小化を用いる（すなわち，マージン以下の誤差は許容される）ため，二極化は起きにくいと考えられる．また Vulic らの研究では，前述したとおり，本研究よりも多くの語彙知識を用いていることにも留意が必要である．

## 5. 考察

### 5.1 分類タスクの誤り分析

WBLESS および BIBLESS のデータセットに含まれる非上位下位語ペア（負例）は，同位・全体–部分など，上位下位以外の意味関係を持つ単語ペアで構成されている．したがって正しく負例と判定できた単語ペアの割合を意味関係ごとに算出すれば，どのような意味関係を上位下位関係と誤認しやすいのかを把握できる．この発想に基づき，意味関係ごとに正解率を求めた結果を**表 4** に示す．提案手法は，BIBLESS の全体–部分関係および部分–全体関係にある単語ペアの正解率が相対的に低いことが分かる．すなわち提案手法は (fox, mouth) や (radio, wire) のような単語ペアを，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，部分語が上位語候補の場合は誤認しにくいことが分かる．したがって，全体語および部分語をそれぞれ意味階層の上位（非ゼロ桁数が小）および下位（非ゼロ桁数が大）だととらえることはできているが，両者に上位下位関係がないことはとらえられていないといえる．この要因としては，非上位下位語ペアを生成する際に，全体–部分関係が出現しにくいことが影響していると考えられる．非上位下位語ペアをなす単語は，語彙全体またはもう一方の単語の最近傍から乱択される．このため非上位下位語ペアは，意味的関係がないか，または意味的類似性が高いかのいずれかになりやすいが，全体–部分関係は必ずしもどちらにも該当しないためである．

**表 4**　分類タスク（WBLESS および BIBLESS）の意味関係ごとの正解率

**Table 4**　Accuracy by each semantic relations for classification tasks (WBLESS and BIBLESS).

| | WBLESS | | BIBLESS | |
|---|---|---|---|---|
| 意味関係 | 正解クラス | 正解率 | 正解クラス | 正解率 |
| 上位–下位 | 上位下位 | 0.901 | 上位下位 | 0.893 |
| 下位–上位 | その他 | 0.976 | 下位上位 | 0.907 |
| 同位 | その他 | 0.914 | その他 | 0.873 |
| 全体–部分 | その他 | 0.838 | その他 | 0.843 |
| 部分–全体 | その他 | 0.966 | その他 | 0.805 |
| ランダム | その他 | 0.930 | その他 | 0.915 |
| すべて | - | 0.919 | - | 0.886 |

### 5.2 ランキングタスクの誤り分析

HyperLex のデータセットは（xBLESS と同様に）さまざまな意味関係を持つ単語ペアから構成されており，なおかつ意味関係ごとに上位下位関係らしさの度合いに一定の

表 5　提案手法と，手法の一部を無効化した場合との比較．アブレーション列は無効化した対象を示す．角カッコ内の数値は提案手法との accuracy の差を報告．平均寄与度は全タスクの accuracy 差の平均値を報告

**Table 5**　Ablation study of the proposed method. The leftmost column indicates the disabled method or subtask. The numbers in square brackets report the difference in accuracy between the proposed method and ablation. The average contribution is the mean value of the difference in accuracy of all tasks.

| アブレーション | BLESS-hyponymy | WBLESS | BIBLESS | HyperLex | 平均寄与度 |
|---|---|---|---|---|---|
| 提案手法 | 0.984 | 0.919 | 0.886 | 0.539 | - |
| -補助目的関数; 再構築損失 | 0.988 [+0.3 pt] | 0.918 [−0.1 pt] | 0.870 [−1.5 pt] | 0.556 [+1.7 pt] | 0.1 |
| -補助目的関数; 相互情報量 | 0.980 [−0.5 pt] | 0.921 [+0.2 pt] | 0.880 [−0.5 pt] | 0.539 [−0.0 pt] | −0.2 |
| -負例生成; 最近傍から乱択 | 0.979 [−0.6 pt] | 0.913 [−0.6 pt] | 0.869 [−1.7 pt] | 0.530 [−0.9 pt] | −0.9 |

傾向があると報告されている [27]．したがって推論した順位と，アノテータが付与した正解の順位をそれぞれ意味関係ごとにグループ化して比較すれば，どのような意味関係で推論と正解の不一致が大きいのかを把握できる．この発想に基づき，意味関係ごとに順位の分布を可視化した結果を**図 3** に示す．提案手法は，上位–下位・同義・下位–上位については，平均的には正解と同様の順序付けができているが，しばしば正解よりもばらつきが大きいことが分かる．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分かる．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．

表 6　BIBLESS における意味関係ごとの，提案手法と手法の一部を無効化した場合との性能差．数値は提案手法との accuracy の差（単位はポイント）を報告

**Table 6**　Detailed result of the ablation study highlighted by each semantic relations for the BIBLESS task. The numbers indicate the difference in accuracy (in points) between the proposed method and ablation.

| 意味関係 | -再構築損失 | -最近傍から乱択 |
|---|---|---|
| 全サンプル | −1.5 | −1.7 |
| 上位–下位 | 0.7 | 0.2 |
| 下位–上位 | 0.5 | −0.5 |
| 同位 | −7.9 | −12.4 |
| 全体–部分 | −8.0 | −1.1 |
| 部分–全体 | −4.9 | −1.8 |
| ランダム | −1.9 | −0.2 |

### 5.3　提案手法の有効性分析

提案手法では，階層コードへの変換器を学習するための手段として，補助目的関数および非上位下位語ペアの生成を用いている．これらの手段が上位下位識別タスクの性能に寄与しているかどうかを調べるために，アブレーションによる有効性分析を実施する．提案手法と，手法の一部を無効化した場合との性能差を**表 5** に示す．この結果から，最近傍からの乱択による非上位下位語ペアの生成が，上位下位識別タスク全般に有効であることが分かる．また BIBLESS に限定すれば，階層コードから元の分散表現を再構築させることが，分類精度の向上に有効であることが分かる．これらはいずれも，意味的類似度が高い単語に対して，異なるコードの割当てを促す効果がある．これにより，上位下位関係とその他の意味関係の識別に寄与するものと考えられる．実際に BIBLESS で意味関係ごとの性能差を調べたところ（**表 6**），補助目的関数または最近傍からの乱択を無効化すると，同位関係および全体–部分関係の性能が 8.0 から 12.4 ポイント低下する一方で，ランダムな単語ペアでの性能差は 2 ポイント未満であることが確認できた．したがって前述の考察は妥当だといえる．

再構築損失が有効であることは，Order Embeddings の実現方法として階層コード表現を用いることの長所とも解釈できる．再構築損失は Denoising AutoEncoder の枠組みで導入できるため，変換器（エンコーダ）・逆変換器（デコーダ）アーキテクチャの設計が必要である．コード表現の場合は，Shu [25] や本研究の提案手法（式 (17)）のように，各桁の値に固有の基底ベクトルを導入することで容易に逆変換器が設計できる．一方で，Order Embeddings の既存研究では確率分布 [1] や超直方体 [17] などを用いる手法が提案されているが，これらの表現に対する逆変換器の設計は，コード表現の場合ほど直接的ではない．

### 5.4　階層コードの割当て特性の分析

提案手法は，単語分散表現を $M$ 進 $N$ 桁の階層コードに変換する．実際には各桁は連続緩和されている（3 章）が，仮に各桁で最大値をとる要素を選択すれば，単語を $M^N$ 通りのいずれかに割り当てる，クラスタリングの一種と見なせる．一方で WordNet についても，語義（Synset）を基準とする単語のクラスタリングと見なせる．そこで本節では，提案手法による階層コードをクラスタリング手法と見なす場合に，どのような割当て特性を持つのかを分析する．具体的には WordNet の語義によるクラスタリングとの比

表 7　階層コードクラスタおよび WordNet 語義クラスタの割当て特性の比較．階層コードは初期パラメータを変えて 5 回実験した平均および標準偏差（カッコ内の数値）を報告

**Table 7**　Comparison of the clustering characteristics of hierarchical codes and WordNet synsets. Hierarchical code reports the mean and standard deviation (numbers in parentheses) of five trials with different model parameter initialization.

| クラスタ | クラスタ数 | 平均単語数 | ARI | 意味階層的類似度（WuP） | | 分散表現類似度（cos） | |
|---|---|---|---|---|---|---|---|
| | | | | クラスタ内 | クラスタ間 | クラスタ内 | クラスタ間 |
| WordNet 語義 | 70,959 | 1.64 | - | 1.00 | 0.23 | 0.62 | 0.32 |
| 階層コード | 71,564 | 1.64 | 0.0024 | 0.53 | 0.23 | 0.73 | 0.32 |
| | (7,120) | (0.15) | (0.0011) | (0.11) | (0.00) | (0.11) | (0.01) |

較および，階層コードの進数・桁数による影響を調べる．

### 5.4.1　分析方法

クラスタリングの対象とする語彙は，本実験で用いた語彙知識（WordNet から抽出した上位下位語ペアの集合．詳細は 4.2 節を参照）に出現するすべての単語を用いる．ただし異なる品詞は異なる単語として扱う．たとえば名詞の “play” と動詞の “play” は異なる単語とする．語彙サイズは 116,510 件（名詞が 105,048 件，動詞が 11,462 件）である．

階層コードによるクラスタの割当ては，各桁で最大値をとる要素を選択する．すなわちクラスタ ID を $z_0 z_1 \dots z_d \dots z_{N-1}$ として

$$z_d = \mathrm{argmax}_a(\pi_{d,a} | a \in \{0, 1, \dots, M-1\}) \qquad (21)$$

と割り当てる．なお $\boldsymbol{\pi}_d$ は，$d$ 桁目の値が従うカテゴリカル分布のパラメータである（式 (7)）．WordNet の語義によるクラスタの割当ては，Synset ID をクラスタ ID とする．ただし多義語の場合は，1 番目の Synset（いわゆる WordNet first Synset．first Synset は出現頻度が最も高いと期待される語義である [13]）を選択する．

割当て特性の評価指標は，クラスタの統計量，クラスタリングの精度，およびクラスタ内・クラスタ間の単語類似度の 3 種類を用いる．クラスタの統計量は，クラスタ数およびクラスタ別の平均単語数を用いる．クラスタリングの精度は，WordNet 語義クラスタを正解として，Adjusted Rand Index（ARI）[*12] [12] を用いる．

単語類似度は，クラスタ内の場合は同一クラスタから，クラスタ間の場合は異なるクラスタから，それぞれ同一品詞の単語ペアを乱択して評価する．サンプル数は 10,000 件である．単語類似度の指標は，WordNet の上位下位関係の階層構造における類似度および，単語分散表現の類似度の 2 種類を用いる．計量指標はそれぞれ，Wu-Palmar（WuP）類似度[*13] [31] および，cosine 類似度を用いる．なお本節での略称はそれぞれ，意味階層的類似度および分散表現類似度とする．

[*12] 部分集合群への割当てかたの近さをとらえる指標．でたらめな割当てならば 0，完全一致ならば 1 となる．
[*13] 木構造におけるノード間の近さをとらえる指標．値域は 0 から 1 の範囲で，一致ならば 1 となる．

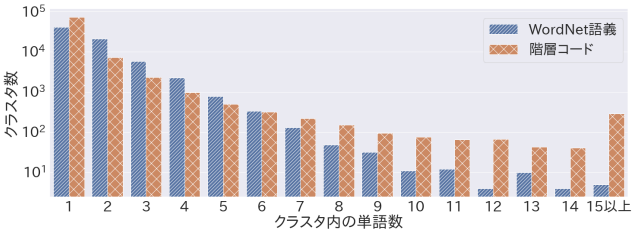


図 4　クラスタ内単語数のヒストグラム

**Fig. 4**　Histogram of the cluster frequency distribution based on the number of words in cluster.

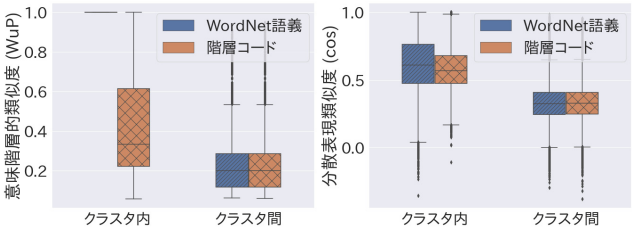


図 5　単語類似度の分布．左：意味階層的類似度（Wu-Palmar），右：分散表現類似度（cosine）

**Fig. 5**　Distribution of the similarity between word pairs. Left: semantic hierarchy-aware similarity (Wu-Palmar). Right: distributional similarity (cosine).

モデル訓練時の統計誤差の影響を考慮するため，階層コードによるクラスタの割当ては，モデルパラメータの初期値を変えて 5 回試行する．

### 5.4.2　WordNet 語義との比較

階層コード学習に用いる語彙知識は，WordNet から抽出した上位下位語ペアである．したがって，階層コードの割当ては，WordNet の上位下位関係の階層構造と関連することが期待される．実用的にも，もしも意味階層的な類似性が高い単語に同一のコードが割り当てられる傾向が確認できれば，同義関係や同位関係の識別可能性に対する示唆にもなる．こうした動機から，階層コードによるクラスタリングの特性を，WordNet の語義によるクラスタリングの特性と比較する．比較結果の一覧を，**表 7** に示す．またクラスタ内単語数のヒストグラムを**図 4** に，サンプリングした単語ペアの単語類似度の分布を**図 5** に示す．

クラスタ数および平均単語数は，階層コードの特性が

表 8　同一の階層コードが割り当てられた単語群の例．クラスタ内単語数が $\{2, 4, 6, 8\}$ 個のクラスタを無作為に抽出

**Table 8**　Examples of word clusters assigned the same hierarchical code. We randomly pick-up the clusters with $\{2, 4, 6, 8\}$ words in the cluster.

| クラスタ ID | 単語数 | 単語群 |
|---|---|---|
| Noun-a | 2 | goldfinch, limpkin |
| Noun-b | 4 | Rodentia, Cricetus, Choloepus, Sarcocephalus |
| Noun-c | 6 | genus_Tympanuchus, genus_Pyrocephalus, genus_Rhyncostylis, genus_Streptococcus, genus_Eucinostomus, genus_Stenopterygius |
| Noun-d | 8 | positional_representation_system, ship-towed_long-range_acoustic_detection_system, concentration, localisation, genetic_marker, feasibility, naval_tactical_data_system, fixed-point_representation_system |
| Verb-a | 2 | come_upon, look_upon |
| Verb-b | 4 | cause, raiment, be_active, evict |
| Verb-c | 6 | solemnize, compartmentalise, analogize, allegorize, sermonize, literalize |
| Verb-d | 8 | derogate, enounce, posit, indite, reveal, declaim, attaint, outguess |

WordNet 語義の特性とよく一致した．たとえば平均単語数は，階層コードと WordNet 語義のいずれも，1.64 件となった．一方でヒストグラムの比較からは，階層コードは，一部のクラスタに割当てが集中する傾向が強いことが分かった．たとえば 15 単語以上が属するクラスタ数は，階層コードでは 284 個だが，WordNet 語義では 5 個のみであった．また階層コードによるクラスタ数の標準偏差は平均の約 10%であり，試行（すなわち，変換器の最適化）によるばらつきがやや大きいことを示唆している．

WordNet 語義との割当ての一致度を表す指標である ARI は，0.0024 であった．この数値は，でたらめな割当てよりも明らかに良い[*14]が，完全一致には程遠いという水準である．すなわち，割り当てた階層コードの同一性を根拠として同義関係を推論することは，きわめて困難であるといえる．

単語類似度は，クラスタ内の意味階層的類似度を除き，階層コードの特性が WordNet 語義の特性とよく一致した．また意味階層的類似度と分散表現類似度のいずれについても，クラスタ間よりもクラスタ内の類似度のほうが高い．すなわち，同一の階層コードが割り当てられる単語群は，異なる階層コードが割り当てられる単語群と比べて，意味階層上の配置と単語分散表現がともに似通るといえる．したがって，割り当てた階層コードが一致するか否かは，単語間の同義関係や同位関係を識別する手がかりになりうると示唆される．ただし ARI の水準がほぼゼロであることおよび，意味階層的類似度が 0.53 であり，同義関係（つまり 1.0）との差が 47 ポイントあることを考慮すると，改良の余地は非常に大きいといえる．

同一の階層コードが割り当てられる単語群の実例を，**表 8** に示す．1 つめの特徴として，複合語の場合は一部の単語が共通することがあげられる．たとえば Noun-d 単語群は *system* が共通している．複合語の分散表現は全単語の算術平均をとる（4.2 節）ため，おのずと分散表現類似度が高くなるのが要因であろう．2 つめの特徴として，同位関係が散見されることがあげられる．たとえば Noun-b 単語群の (Rodentia, Cricetus) は間接の同位関係（WuP=0.625），Noun-c 単語群の (genus_Tympanuchus, genus_Pyrocephalus) は直接の同位関係（WuP=0.875）である．もちろんすべてが同位関係ではないが，クラスタ間よりもクラスタ内の意味階層的類似度が高いという結果と整合的である．

### 5.4.3　進数および桁数の影響

階層コードがとりうる場合の数は，$M$ 進 $N$ 桁のとき，$M^N$ 通りである．したがって進数および桁数を変化させると，クラスタの割当て特性が変化することが予想される．こうした動機から，8 進 16 桁のデフォルト設定をベースラインとして，進数および桁数をそれぞれ 2 倍または半減させた場合の割当て特性を計測し，ベースラインとの差および，差の統計的な有意性を評価する．また参考として，上位下位識別タスクの精度についても評価する．実験結果を**表 9** に示す．

クラスタ数は，進数・桁数の増減と連動して増減する結果となった．提案手法の目的関数は，再構築損失および相互情報量を補助的に併用しているため，とりうる場合の数に連動して，コードの分散・集中を促す度合いが上下動するのだと考えられる．

単語類似度は，進数を 2 倍にすると，統計的に有意ではないものの，クラスタ内の類似度が上昇した．すなわち，進数を増やすと，同一の階層コードが割り当てられる単語どうしの類似性が高まる傾向があるようだ．また ARI も，0.1 ポイントと微量ながら上昇している．したがって，進数を増やすと階層コードの割当て特性が WordNet の上位下位関係の階層構造に近づく傾向があると示唆される．

上位下位識別タスクの精度は，HyperLex の一部を除き，ベースラインとの差は 1 ポイント以内であった．またベースラインとの統計的な有意差があるのは 16 通りのうち 3

[*14] でたらめな割当てによる ARI のシミュレーション値は $2.6 \times 10^{-7}$ であった．

**表 9**　進数および桁数の影響．アスタリスク（*）はデフォルト設定との差が Welch の t 検定（両側検定）により $p < 0.05$ で有意．見出しの “C” は “クラスタ” の略記

**Table 9**　Effect of the number of digit and ary. Asterisks (*) indicate that the difference with respect to the default setting is statistically significant at $p < 0.05$ by Welch's t-test (two-tailed test). The “C” in the header is the abbreviation of “cluster”.

| 設定 | クラスタ数 | 平均単語数 | ARI | 意味階層的類似度 | | 分散表現類似度 | | 上位下位識別タスク | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | C 内 | C 間 | C 内 | C 間 | B-hyp | WB | BIB | HLex |
| デフォルト ($M = 8$, $N = 16$) | 71,564 | 1.64 | 0.0024 | 0.53 | 0.23 | 0.73 | 0.32 | 0.984 | 0.919 | 0.886 | 0.539 |
| 桁数 2 倍 $N = 32$ | 103,914* | 1.17* | 0.0008* | 0.48 | 0.24 | 0.69 | 0.33 | 0.984 | 0.916 | 0.873 | 0.519 |
| 進数 2 倍 $M = 16$ | 101,835* | 1.15* | 0.0036 | 0.60 | 0.24 | 0.83 | 0.33 | 0.985 | 0.924 | 0.889 | 0.540 |
| 桁数半減 $N = 8$ | 6,150* | 19.43* | 0.0007* | 0.42 | 0.23 | 0.53* | 0.32 | 0.984 | 0.925* | 0.887 | 0.564* |
| 進数半減 $M = 4$ | 19,333* | 6.15* | 0.0009* | 0.41 | 0.23 | 0.54* | 0.31 | 0.980* | 0.921 | 0.880 | 0.538 |

通りのみであった．したがって進数および桁数は，本タスクに強く影響するパラメータではないといえる．

以上の知見に基づき，適切な進数および桁数を探索する方法を考察する．仮に上位下位識別タスクの精度のみが関心事の場合は，進数・桁数の影響は小さいため，幅広く探索する意義は小さいといえる．一方で階層コードの割当て特性を WordNet の上位下位関係の階層構造に近づけることが関心事の場合は，進数および桁数を小さな値から始めて徐々に大きくしてゆき，クラスタ内単語ペアの Wu-Palmar 類似度が低下に転じるか飽和する周辺で探索を止めるのがよいだろう．

### 5.5　間接的な上位下位語ペアの影響分析

階層コードへの変換器（エンコーダ）の訓練に用いる上位下位語ペアは，直接・間接いずれの上位下位関係をも利用できる．このため本実験（4.3 節）では，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使用する（4.2 節）という設定での実験結果を報告した．一方で，提案手法は包含関係を定義可能な階層コードという表現を用いるため，原理的には，直接の “is-a” 関係のみ与えて訓練すれば，間接的な上位下位関係を推論できる[*15]．したがって，訓練データに含める間接的な上位下位語ペアを意図的に制限する場合に，上位下位識別タスクの性能がどのような影響を受けるのかは，興味深い問いである．仮に間接的な上位下位語ペアを制限しても十分な精度が得られるならば，訓練に要する計算負荷の削減や，推移性も含めた上位下位語ペアの網羅性に対する要求の緩和に寄与するためである．特に後者の観点は，網羅性が保証されない機械的な収集 [10], [23], [33] を前提とする場合に，実用的にも重要である．したがって本節では，訓練データに含める間接的な上位下位語ペアを制限する場合の影響を実験的に評価する．具体的には，すべての単語ペアを用いる場合をベースラインとして，“is-a” 関係のホップ数が指定値以下の上位下位語ペアのみを用いる場合のタスク性能を計測し，ベースラインとの精度の差および，差の統計的な有意性を評価する．なおモデル訓練時の統計誤差の影響を考慮するため，各設定に対してモデルパラメータの初期値を変えて 5 回試行する．実験結果を**表 10** に示す．

ホップ数の範囲を大きくするにつれて，すなわち訓練に用いる間接的な上位下位語ペアを増やすにつれて，すべてのタスクで精度が向上した．精度向上幅はほぼ単調増加で，ベースラインに漸近する傾向を示した．また 5・7・9 ホップ以下の設定ではベースラインとの統計的な有意差が認められないが，精度向上の傾向は飽和していない．したがって，上位下位識別タスクの性能を高くするためには，間接的な上位下位語ペアを網羅的に訓練データに含めること，すなわち仮に直接の上位下位語ペアのみが与えられた場合は，有向非巡回グラフを構築して，間接的な単語ペアを網羅的に生成することが望ましいといえる．

タスクごとにホップ数の影響を比較すると，精度向上幅が最も小さいタスクは BLESS-hyponymy で，最も大きいタスクは BIBLESS である．この結果から得られる示唆は，ホップ数が大きい上位下位語ペアは，意味階層の深さ（コードの非ゼロ桁数）の学習にはさほど貢献しないが，意味階層の包含関係（非ゼロ桁の値の一致）の学習に貢献する，ということである．階層コードの値を学習する（$N$ 桁それぞれについて $M$ 個の選択肢がある）ことは，非ゼロ桁数を学習する（はじめてゼロを出力する桁を $N$ 個の選択肢から選ぶ）ことよりも難しいというのが直観的な解釈だが，本分析の示唆はこの直観と整合的である．

### 5.6　非上位下位語ペアの生成に関するハイパーパラメータの影響分析

提案手法に含まれる各種のハイパーパラメータは，上

[*15] 非上位下位関係については必ずしもそうではない．なぜならば，訓練時に非上位下位語ペアを生成する際に，本来は間接的な上位下位関係にある単語ペアが誤って生成されうるためである（正例になければ棄却できない）．本節の実験ではこの問題を捨象しているため，やや厳しい評価であることに留意されたい．

**表 10**　訓練データに含める間接的な上位下位語ペアの影響．丸カッコ内の数値は全ペア数に対する比率．角カッコ内の数値はデフォルト設定との accuracy の差．アスタリスク（*）は accuracy の差が Welch の t 検定（両側検定）により $p < 0.05$ で有意

**Table 10**　Effect of the indirect hyponymy word pairs in the trainset. The numbers in round brackets of the setting names are the ratio of the number of word pairs against all pairs. The numbers in square brackets are the difference in accuracy between the default setting and each settings. Asterisks (*) indicate that the difference in accuracy with respect to the default setting is statistically significant at $p < 0.05$ by Welch's t-test (two-tailed test).

| ホップ数の範囲 | BLESS-hyponymy | WBLESS | BIBLESS | HyperLex |
|---|---|---|---|---|
| 制限なし (100.0%) | 0.984 | 0.919 | 0.886 | 0.539 |
| 1 ホップ (14.2%) | 0.952 [−3.3 pt]* | 0.808 [−11.1 pt]* | 0.710 [−17.6 pt]* | 0.447 [−9.2 pt]* |
| 2 ホップ以下 (27.9%) | 0.980 [−0.5 pt] | 0.858 [−6.0 pt]* | 0.783 [−10.2 pt]* | 0.501 [−3.7 pt]* |
| 3 ホップ以下 (41.8%) | 0.983 [−0.2 pt] | 0.893 [−2.6 pt]* | 0.838 [−4.8 pt]* | 0.526 [−1.3 pt] |
| 4 ホップ以下 (54.7%) | 0.983 [−0.1 pt] | 0.904 [−1.5 pt]* | 0.857 [−2.9 pt]* | 0.537 [−0.2 pt] |
| 5 ホップ以下 (66.9%) | 0.984 [−0.0 pt] | 0.910 [−0.9 pt] | 0.868 [−1.8 pt] | 0.532 [−0.7 pt] |
| 7 ホップ以下 (85.8%) | 0.983 [−0.1 pt] | 0.917 [−0.1 pt] | 0.879 [−0.7 pt] | 0.536 [−0.3 pt] |
| 9 ホップ以下 (94.9%) | 0.982 [−0.2 pt] | 0.923 [+0.4 pt] | 0.886 [−0.0 pt] | 0.546 [+0.7 pt] |

**表 11**　最近傍からの乱択に関するハイパーパラメータの影響．角カッコ内の数値はデフォルト設定との accuracy の差．アスタリスク（*）は accuracy の差が Welch の t 検定（両側検定）により $p < 0.05$ で有意

**Table 11**　Effect of hyperparameters of the random sampling from nearest neighbor words. The numbers in square brackets are the difference in accuracy between the default setting and each settings. Asterisks (*) indicate that the difference in accuracy with respect to the default setting is statistically significant at $p < 0.05$ by Welch's t-test (two-tailed test).

| 設定 | BLESS-hyponymy | WBLESS | BIBLESS | HyperLex |
|---|---|---|---|---|
| デフォルト ($q = 30\%$, $k = 100$) | 0.984 | 0.919 | 0.886 | 0.539 |
| 最近傍から乱択する確率 ($q$) を変更 | | | | |
| $q = 0\%$ [*16] | 0.979 [−0.6 pt] | 0.913 [ 0.6 pt]* | 0.869 [−1.7 pt]* | 0.530 [−0.9 pt] |
| $q = 50\%$ | 0.983 [−0.1 pt] | 0.918 [−0.1 pt] | 0.880 [−0.6 pt] | 0.525 [−1.4 pt] |
| $q = 70\%$ | 0.984 [−0.1 pt] | 0.910 [−0.9 pt] | 0.871 [−1.5 pt] | 0.531 [−0.8 pt] |
| $q = 100\%$ | 0.985 [+0.1 pt] | 0.904 [−1.5 pt]* | 0.857 [−2.9 pt]* | 0.501 [−3.8 pt]* |
| 最近傍の単語数 ($k$) を変更 | | | | |
| $k = 10$ | 0.988 [+0.3 pt] | 0.922 [+0.3 pt] | 0.888 [+0.2 pt] | 0.532 [−0.7 pt] |
| $k = 30$ | 0.983 [−0.2 pt] | 0.922 [+0.3 pt] | 0.888 [+0.2 pt] | 0.535 [−0.4 pt] |
| $k = 300$ | 0.986 [+0.1 pt] | 0.922 [+0.3 pt] | 0.890 [+0.4 pt] | 0.541 [+0.2 pt] |
| $k = 500$ | 0.981 [−0.3 pt] | 0.919 [−0.0 pt] | 0.876 [−0.9 pt] | 0.540 [+0.1 pt] |

位下位識別タスクの性能に影響すると考えられる．また 5.3 節では，最近傍からの乱択による非上位下位語ペア（負例）の生成がタスクに有効とする考察を得た．したがって本節では，最近傍からの乱択に関するハイパーパラメータの影響を実験的に評価する．具体的には，デフォルトの設定（3.5 節）をベースラインとして，最近傍から乱択する確率 $q$ および最近傍の単語数 $k$ を変更した設定におけるタスクの性能を計測し，ベースラインとの精度の差および，差の統計的な有意性を評価する．なおモデル訓練時の統計誤差の影響を考慮するため，各設定に対してモデルパラメータの初期値を変えて 5 回試行する．実験結果を**表 11** に示す．

最近傍の単語数は，すべての設定においてベースラインとの精度差は統計的に有意ではなかった．したがって，提案手法は最近傍の単語数に対して鈍感だといえる．

最近傍から乱択する確率は，0%（最近傍からの乱択が無効）および 100%（全語彙からの乱択が無効）の場合に，デフォルト設定の精度を統計的に有意に下回った．特に 100%の場合は，デフォルト設定を 1.5 ポイントから 3.8 ポイント下回り，大きく精度が低下した．確率を 50%および 70%に増やす場合は，統計的に有意ではないものの，デフォルト設定を 0.1 ポイントから 1.5 ポイント下回った．以上の結果から，最近傍から乱択する確率は，上位下位識

[*16] $q = 0\%$ の設定は，最近傍からの乱択の無効化と同義である．このため，アブレーションによる有効性分析（表 5）の "-負例生成; 最近傍から乱択" の実験結果を再利用した．

別タスクの性能に相応に影響するといえる．最近傍からの乱択を完全に有効化あるいは無効化すると，いずれも精度が低下する．また最近傍から乱択する確率の最適値は，デフォルト設定のとおり，30%前後であると示唆される．

## 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するアーキテクチャと，階層コードのペアに対して包含関係らしさを微分可能な形式で計量する手法を提案した．そのうえで，大量の上位下位語ペアを教師信号とすることにより，上位語と下位語が包含関係のあるコードに変換されるような変換器を学習できることを示した．また提案手法の有効性を示すため，変換により得られた階層コードを用いていくつかの上位下位識別タスクを解いた．その結果，提案手法は上位下位関係の分類タスクにおいて，Semantic Specialization や Order Embeddings による既存手法を上回る性能であることを報告した．

タスクの誤り分析からは，全体–部分関係および対義関係の誤差が大きいことが分かった．これらは手法そのものの性質というよりも，最適化における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．また提案手法の有効性分析からは，再構築損失による補助タスクおよび最近傍語による負例の生成といった工夫が，性能向上への寄与が大きいことが分かった．これらの工夫が性能向上に寄与する理由は，上位下位関係とその他の意味関係との識別を改善するためであることが示唆された．

最後に，今後の課題を述べる．本研究で提案した手法は，単語ごとに 1 つのコードを与える．しかし単語はしばしば複数の語義を持つため，語義に応じて異なるコードを与えるのが理想的である．たとえば mouse という単語は，device の文脈と animal の文脈とで異なるコードを与えられてよいはずである．近年は ELMo や BERT のように，文脈に依存する単語分散表現の計算方法が多数提案されている．これらの分散表現を提案手法と組み合わせることにより，文脈に依存する階層コード表現の学習が可能かを検証したい．

**謝辞**　本研究は JSPS 科研費 19H01118 の助成を受けた．

### 参考文献


[1] Athiwaratkun, B. and Wilson, A.G.: Hierarchical Density Order Embeddings, *Proc. 6th International Conference on Learning Representations* (2018).

[2] Biran, O. and McKeown, K.R.: Classifying Taxonomic Relations between Pairs of Wikipedia Articles, *Proc. 6th International Joint Conference on Natural Language Processing*, pp.788–794 (2013).

[3] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching Word Vectors with Subword Information, *Trans. Association for Computational Linguistics*, Vol.5, pp.135–146 (2017).

[4] Bordea, G., Lefever, E. and Buitelaar, P.: SemEval-2016 Task 13: Taxonomy Extraction Evaluation (TExEval-2), *Proc. 10th International Workshop on Semantic Evaluation*, pp.1081–1091 (2016).

[5] Camacho-Collados, J.: Why we have switched from building full-fledged taxonomies to simply detecting hypernymy relations, *CoRR*, Vol.abs/1703.04178 (2017).

[6] Dagan, I., Roth, D., Sammons, M. and Zanzotto, F.M.: *Recognizing Textual Entailment: Models and Applications*, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2013).

[7] Dash, S., Chowdhury, M.F.M., Gliozzo, A., Mihindukulasooriya, N. and Fauceglia, N.R.: Hypernym Detection Using Strict Partial Order Networks, *Proc. 34th AAAI Conference on Artificial Intelligence*, pp.7626–7633 (2020).

[8] Fellbaum, C.: *WordNet: An Electronic Lexical Database*, The MIT Press (1998).

[9] Foret, P., Kleiner, A., Mobahi, H. and Neyshabur, B.: Sharpness-Aware Minimization for Efficiently Improving Generalization, *CoRR*, Vol.abs/2010.01412 (2020).

[10] Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora, *14th International Conference on Computational Linguistics*, pp.539–545 (1992).

[11] Hu, W., Miyato, T., Tokui, S., Matsumoto, E. and Sugiyama, M.: Learning Discrete Representations via Information Maximizing Self-Augmented Training, *Proc. 34th International Conference on Machine Learning*, pp.1558–1567 (2017).

[12] Hubert, L. and Arabie, P.: Comparing partitions, *Journal of classification*, Vol.2, No.1, pp.193–218 (1985).

[13] Jurafsky, D. and Martin, J.H.: *Speech and Language Processing* (*2nd Edition*), chapter 20, pp.678–679, Prentice-Hall, Inc. (2009).

[14] Kiela, D., Rimell, L., Vulic, I. and Clark, S.: Exploiting Image Generality for Lexical Entailment Detection, *Proc. 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).

[15] Kingma, D.P. and Welling, M.: Auto-Encoding Variational Bayes, *2nd International Conference on Learning Representations, Conference Track Proceedings* (2014).

[16] Levy, O., Remus, S., Biemann, C. and Dagan, I.: Do Supervised Distributional Methods Really Learn Lexical Inference Relations?, *Proc. 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.970–976 (2015).

[17] Li, X., Vilnis, L., Zhang, D., Boratko, M. and McCallum, A.: Smoothing the Geometry of Probabilistic Box Embeddings, *Proc. 7th International Conference on Learning Representations* (2019).

[18] Maddison, C.J., Mnih, A. and Teh, Y.W.: The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables, *Proc. 5th International Conference on Learning Representations* (2017).

[19] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. and Joulin, A.: Advances in Pre-Training Distributed Word Representations, *Proc. 11th International Conference on Language Resources and Evaluation* (2018).

[20] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neu-*

*ral Information Processing Systems 26*, pp.3111–3119 (2013).
[21] Nguyen, K.A., Köper, M., im Walde, S.S. and Vu, N.T.: Hierarchical Embeddings for Hypernymy Detection and Directionality, *Proc. 2017 Conference on Empirical Methods in Natural Language Processing*, pp.233–243 (2017).
[22] Nickel, M. and Kiela, D.: Poincaré Embeddings for Learning Hierarchical Representations, *Advances in Neural Information Processing Systems 30*, pp.6338–6347 (2017).
[23] Roller, S., Kiela, D. and Nickel, M.: Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora, *Proc. 56th Annual Meeting of the Association for Computational Linguistics*, pp.358–363 (2018).
[24] Shen, D., Su, Q., Chapfuwa, P., Wang, W., Wang, G., Henao, R. and Carin, L.: NASH: Toward End-to-End Neural Architecture for Generative Semantic Hashing, *Proc. 56th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), pp.2041–2050 (2018).
[25] Shu, R. and Nakayama, H.: Compressing Word Embeddings via Deep Compositional Code Learning, *Proc. 6th International Conference on Learning Representations* (2018).
[26] Vendrov, I., Kiros, R., Fidler, S. and Urtasun, R.: Order-Embeddings of Images and Language, *4th International Conference on Learning Representations, Conference Track Proceedings* (2016).
[27] Vulic, I., Gerz, D., Kiela, D., Hill, F. and Korhonen, A.: HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment, *Computational Linguistics*, Vol.43, No.4, pp.781–835 (2017).
[28] Vulic, I. and Mrksic, N.: Specialising Word Vectors for Lexical Entailment, *Proc. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.1134–1145 (2018).
[29] Wang, C. and He, X.: BiRRE: Learning Bidirectional Residual Relation Embeddings for Supervised Hypernymy Detection, *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, pp.3630–3640 (2020).
[30] Weeds, J., Clarke, D., Reffin, J., Weir, D.J. and Keller, B.: Learning to Distinguish Hypernyms and Co-Hyponyms, *Proc. 25th International Conference on Computational Linguistics: Technical Papers*, pp.2249–2259 (2014).
[31] Wu, Z. and Palmer, M.: Verb Semantics and Lexical Selection, *32nd Annual Meeting of the Association for Computational Linguistics*, pp.133–138 (1994).
[32] Zheng, L., Su, Q., Shen, D. and Chen, C.: Generative Semantic Hashing Enhanced via Boltzmann Machines, *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, pp.777–788 (2020).
[33] 隅田飛鳥，吉永直樹，鳥澤健太郎：Wikipedia の記事構造からの上位下位関係抽出，自然言語処理，Vol.16, No.3, pp.3–24 (2009).

# 付　　録

## A.1　詳細な計算式

本文では省略した，上位下位関係の確率および，非ゼロ桁数と分散表現の相互情報量の詳細な計算式を示す．

### A.1.1　上位下位関係の確率

単語ペアの上位下位関係 $P(s \prec t)$ および，非上位下位関係 $P(s \not\prec t)$ の計量は，式 (11) の期待値を解析的に計算して求める．なお以下の説明では訓練時を想定して $\hat{\boldsymbol{c}}_d$ を分布パラメータとして数式を記述するが，推論時は $\hat{\boldsymbol{c}}_d$ を $\boldsymbol{\pi}_d$ と読み替えればよい（式 (4) を参照）．

期待値計算に用いる確率分布 $\boldsymbol{C} \sim \hat{P}(\boldsymbol{C})$ は，式 (10) を用いる．$\hat{P}(\boldsymbol{C})$ は各桁が互いに独立なので，$d$ 桁目の値が $a$ をとる確率は $P(C_d = a) = \hat{c}_{d,a}$ となる．これを利用して，RELATION の If ブロック条件式が成立する割合（確率）を求める．たとえば条件式：$C^s_d = 0 \wedge C^t_d \neq 0$ が成立する割合は $\hat{c}^s_{d,0}(\sum_{a\neq 0} \hat{c}^t_{d,a})$ と計算する．If 文は桁数ぶん反復する For ループで囲まれているため，RELATION が $\{-1, 0, 1\}$ を返す割合は，$d$ 桁目ではじめて該当する If ブロックの条件式が成立する割合を，すべての桁について合計した値となる．

ところで $\hat{P}(\boldsymbol{C})$ は各桁が互いに独立なので，階層コードの定義を満たさない事象，たとえば $(3, 1, 0, 1)$ のように，ゼロの後続桁で非ゼロが出現する事象に非ゼロの確率を与える．すなわち $\hat{P}(\boldsymbol{C})$ は，厳密には階層コードのみを台とする確率分布ではない．しかし上位下位関係の計量においては，後述のとおり，ゼロが出現する（すなわち，For ループが終了する）桁以降は計量に寄与しない．つまり階層コードの定義を満たす事象のみが上位下位関係の計算対象になるため，階層コードの定義との矛盾はない．

さて，まず $P(s \prec t)$ すなわち RELATION が 1 を返す場合を定式化する．$\beta^{st}_d, \gamma^{st}_d, \delta^{st}_d$ を，それぞれ 1 を返す If ブロック，For ループ継続の If ブロック，0 を返す If ブロックの条件式が成立する割合

$$\beta^{st}_d := P(C^s_d = 0 \wedge C^t_d \neq 0) \tag{A.1}$$
$$\gamma^{st}_d := P(C^s_d = C^t_d \wedge C^s_d \neq 0 \wedge C^t_d \neq 0) \tag{A.2}$$
$$\delta^{st}_d := P(C^s_d = 0 \wedge C^t_d = 0) \tag{A.3}$$

と定義すると，これらの変数の値は

$$\beta^{st}_d = \hat{c}^s_{d,0}(1 - \hat{c}^t_{d,0}) \tag{A.4}$$
$$\gamma^{st}_d = \sum_{a=1}^{M-1} \hat{c}^s_{d,a}\hat{c}^t_{d,a} \tag{A.5}$$
$$\delta^{st}_d = \hat{c}^s_{d,0}\hat{c}^t_{d,0} \tag{A.6}$$

と計算できる．なお $\gamma^{st}_d$ の計算式における添字 $a$ が 1 から始まる理由は，条件式：$C^s_d \neq 0 \wedge C^t_d \neq 0$ を満たすには $a = 0$ をとりえないためである．

これを $d = 0, 1, \ldots, N-1$ 桁目まで合計することで $P(s \prec t)$ は

$$P(s \prec t) = \sum_{d=0}^{N-1} \beta_d^{st} \left( \prod_{d'=-1}^{d-1} \gamma_{d'}^{st} \right);\ \gamma_{-1}^{st} = 1 \qquad (A.7)$$

と計算できる．

次に $P(s = t)$ すなわち RELATION が 0 を返す場合については

$$P(s = t) = \prod_{d=0}^{N-1} \gamma_d^{st} + \sum_{d=0}^{N-1} \delta_d^{st} \left( \prod_{d'=-1}^{d-1} \gamma_{d'}^{st} \right) \qquad (A.8)$$

と計算できる．

最後に，非上位下位関係の確率は

$$P(s \not\prec t) = 1 - (P(s \prec t) + P(s = t)) \qquad (A.9)$$

と計算できる．

### A.1.2　非ゼロ桁数と分散表現の相互情報量

非ゼロ桁数と分散表現の相互情報量 $\mathrm{I}(\mathrm{length}(\boldsymbol{C}); V)$ は，非ゼロ桁数の確率分布 $P(\mathrm{length}(\boldsymbol{C}) = n|V)$ および，分散表現の確率分布 $P(V)$ を定義すれば計算できる．非ゼロ桁数の確率分布は，分散表現を変換器に入力して得られる階層コードの確率分布から求められるため，非ゼロ桁数と分散表現は相互に依存している．

まず，相互情報量を確率分布を用いた記法に変換しておく．また記述を簡略化するため，以下の式では $\mathrm{length}(\boldsymbol{C}) = S$ と表記する．相互情報量，エントロピー，周辺化分布の定義より

$$\mathrm{I}(S; V) = \mathrm{H}(S) - \mathrm{H}(S|V) \qquad (A.10)$$

$$\mathrm{H}(S) = -\sum_{n=0}^{N} P(S = n) \ln P(S = n) \qquad (A.11)$$

$$\mathrm{H}(S|V) = \mathop{\mathbb{E}}_{P(V)} \left[ \sum_{n=0}^{N} P(S = n|V) \ln P(S = n|V) \right] \qquad (A.12)$$

$$P(S = n) = \mathop{\mathbb{E}}_{P(V)} \left[ P(S = n|V) \right] \qquad (A.13)$$

となる．すなわち $\mathrm{I}(S; V)$ は，非ゼロ桁数について，無条件確率分布および分散表現による条件付確率分布のエントロピーの差である．

次に，非ゼロ桁数の確率分布を定義する．非ゼロ桁数 $n \in \{0, 1, \ldots, N\}$ ははじめて値がゼロになる桁位置に等しい（最後までゼロが出現しなければ $n = N$ になる）．上位下位関係の計量と同様に，階層コードの確率分布を $\hat{P}(\boldsymbol{C})$（式 (10)）を用いて近似すると，$d$ 桁目の値がゼロになる確率は $\hat{c}_{d,0}$ なので

$$P(S = n|V = \boldsymbol{v}) \approx P(S = n|\hat{\boldsymbol{C}}) = \hat{c}_{n,0} \prod_{d=-1}^{n-1} (1 - \hat{c}_{d,0})$$

$$\text{ただし}\quad \hat{c}_{-1,0} = 0,\ \hat{c}_{N,0} = 1 \qquad (A.14)$$

と計算できる．

最後に，分散表現の確率分布を定義する．これは経験分布を用いて

$$P(V = \boldsymbol{v}) \approx \frac{1}{|\mathbb{V}|} \sum_{w \in \mathbb{V}} \mathbb{1}\{\boldsymbol{v} = \boldsymbol{v}^w\} \qquad (A.15)$$

と近似する．ただし $\mathbb{V}$ は，単語分散表現の語彙である．

以上を組み合わせると，式 (A.10) で定義した条件付エントロピーおよび周辺分布は

$$\mathrm{H}(S|V) \approx \frac{1}{|\mathbb{V}|} \sum_{w \in \mathbb{V}} \sum_{n=0}^{N} P(S^w = n) \ln P(S^w = n) \qquad (A.16)$$

$$P(S = n) \approx \frac{1}{|\mathbb{V}|} \sum_{w \in \mathbb{V}} P(S^w = n) \qquad (A.17)$$

と近似計算できる．ただし $P(S = n|V = \boldsymbol{v}^w)$ を $P(S^w = n)$ と略記した．


**水木 栄**　（学生会員）

1984 年生．2007 年名古屋大学工学部機械・航空工学科卒業．2009 年東京大学大学院工学系研究科修士課程修了．金融機関を経て，2015 年株式会社ホットリンク入社．自然言語処理技術によるソーシャルメディアへの応用研究に従事．2018 年東京工業大学情報理工学院博士課程入学，現在に至る．言語処理学会会員．

**岡崎 直観**　（正会員）

1979 年生．2007 年東京大学大学院情報理工学系研究科博士課程修了．2007 年同大学大学院情報理工学系研究科・特任研究員．2011 年東北大学大学院情報科学研究科准教授．2017 年東京工業大学情報理工学院教授．自然言語処理の研究に従事．言語処理学会，人工知能学会，ACL 各会員．


（担当編集委員　吉田 稔）