

# 知識グラフ埋め込みによる候補削減と定義文に基づく 言語モデルを用いた二段階オントロジー補完

## Two-Stage Ontology Completion with KGE-based Candidate Reduction and Definition-based LM Reranking

高橋颯<sup>1\*</sup> 兼岩憲<sup>1</sup>  
Hayase Takahashi<sup>1</sup> Ken Kaneiwa<sup>1</sup>

<sup>1</sup> 電気通信大学大学院 情報理工学研究科 情報・ネットワーク工学専攻

<sup>1</sup> Department of Computer and Network Engineering, Graduate School of Informatics  
and Engineering, The University of Electro-Communications

**Abstract:** Ontology completion is an important task for automating the enrichment of ontologies, which are often incomplete due to the high cost of manual construction. While recent BERT-based methods achieve high prediction accuracy for the task, they have high computational costs. To address this, we propose a two-stage architecture that integrates BERT-based methods with KGE models to achieve both high accuracy and low computational cost. The architecture first uses a KGE model for efficient candidate selection, reducing the search space. Second, a BERT model reranks only this reduced candidate set. Furthermore, we introduce two techniques to improve the model’s performance: a KGE-aided negative sampling strategy for fine-tuning, and an input tokenization method using ‘definition sentences’. Evaluation experiments on two ontologies show that our proposed method exceeds the accuracy of existing methods, while significantly reducing computational costs compared to conventional BERT-based approaches.

## 1 はじめに

オントロジーは、特定分野の概念や知識を体系的に整理し、共通の語彙と理解を共有するための基盤であり、セマンティックウェブや人工知能の分野で広く利用されている [1]。しかし、手動での構築、維持には多大なコストがかかるため、多くのオントロジーは不完全である。そのためオントロジー補完タスクは、オントロジーの品質と実用性を高める上で極めて重要な研究課題である。

近年は、オントロジーを低次元ベクトルに埋め込み、その埋め込みを活用してオントロジーの補完を行う手法が主流である。TransE[2], RotatE[9], HAKE[10] のような知識グラフ埋め込みによる手法や、ELEm[6] のような記述論理に対する埋め込み手法が存在する。また、OWL2Vec\*[3] は論理構造、グラフ構造、アノテーションのテキスト情報の3つを統合した学習コーパスを対象に、Word2Vec[8] を適用してオントロジー埋め込みを実現する。飯島ら [11] は RotatE をベース

として、関係によるエンティティの回転移動に半径の拡大縮小を加えることと、OWL2Vec\*による埋め込み初期化を併用して、高い性能を記録している。

しかし、従来の OWL オントロジー埋め込み手法は、BERT[5] などの Transformer ベースの文脈依存のモデルに比べて性能が劣る。これは OWL2Vec\*などが文脈に依存しない静的な単語埋め込みモデルを採用しているからである。この課題に対し、BERTSubs[4] はオントロジー補完タスクを BERT の下流分類タスクとしてモデル化し、高い予測精度を達成している。しかし、BERT のような文脈を考慮した大規模言語モデルは、予測精度の向上に寄与する一方で、計算コストが増大してしまう。特にオントロジータスクの1つである包含関係の補完では、あるクラスの包含関係判定に、他の候補となる数万の全クラスとのペアについて個別に評価する必要があるため、計算コストがオントロジーのサイズに比例して増大する。

そこで、本研究では包含関係の補完タスクにおいて、高速に動作する知識グラフ埋め込みモデル (KGE) と BERT を用いたオントロジー補完を組み合わせた、予測精度の高さと計算コストの低さを両立する手法を提案する。第1段階として、KGE モデルを用いて膨

\*連絡先: 電気通信大学大学院 情報理工学研究科  
〒182-8585 東京都調布市調布ヶ丘 1-5-1  
E-mail: takahashi@sw.cei.uec.ac.jp

大な候補集合をより有力な定数個の候補のみの集合へと選定する．次に第2段階として，この定数個の候補を対象に，BERT を用いて高精度なリランキングを行う．これにより，BERT の評価のステップ数がオントロジーのサイズに依存しない計算量に減少する．

さらに，BERT を用いたオントロジー補完の性能向上のため，ファインチューニングにおいて KGE モデルで予測した候補によるネガティブサンプリングと，定義文を活用した入力トークン列の作成の2つの工夫を導入する．本研究の新規性は以下の3点である．

- KGE による候補選定と BERT によるリランキングを組み合わせた二段階オントロジー補完アーキテクチャ
- BERT のファインチューニングにおける，KGE の予測により選定したネガティブサンプリング
- BERT を用いたオントロジー補完タスクにおける，定義文を活用した入力トークン列の作成

評価実験では，2つのオントロジーでの補完タスクにおいて，既存手法の精度を提案手法が上回ることを示す．また，従来の BERT を使用する手法に比べて計算コストが低いことを示す．

本稿の構成は以下の通りである．2章では，OWL オントロジーの概要と，関連研究について述べる．3章では問題設定と，提案手法の詳細について述べる．4章では，提案モデルの評価実験を行い，既存モデルとの結果の比較および分析を行う．5章では本稿の結論を述べる．

## 2 準備と関連研究

### 2.1 OWL オントロジー

オントロジーをウェブ上で記述するための言語の一つが OWL である．OWL オントロジーは，クラス (概念)，プロパティ (関係)，インスタンス (個体) などの要素を用いて記述され，各要素は IRI (Internationalized Resource Identifier) によって一意に識別される．IRI は，ウェブ上のリソースに与えられるユニークな ID であり，例えば [http://purl.obolibrary.org/obo/FOODON\\_03400819](http://purl.obolibrary.org/obo/FOODON_03400819) は “unleavened bread, crisp bread and rusk” というクラスを指す．より形式的には，オントロジーは TBox と ABox という2つの公理集合から構成される．ここで，クラスの集合を  $\mathbb{C}$ ，プロパティの集合を  $\mathbb{R}$ ，インスタンスの集合を  $\mathbb{I}$  とする．TBox は，ドメインにおける概念間の普遍的な関係を定義する部分であり， $C, D \in \mathbb{C}$  を用いた包含関

係  $C \sqsubseteq D$  などが含まれる．ABox は  $a, b \in \mathbb{I}, R \in \mathbb{R}$  を用いたインスタンス公理  $C(a)$  や二項関係  $R(a, b)$  などから構成される．

### 2.2 知識グラフ埋め込み

エンティティの集合を  $\mathbb{E}$ ，関係の集合を  $\mathbb{R}$  とし， $h, t \in \mathbb{E}, r \in \mathbb{R}$  としたとき知識グラフはトリプル  $(h, r, t)$  の有限集合として表現される．知識グラフ埋め込みは，これらのエンティティおよび関係を低次元のベクトル空間に写像する手法である．知識グラフ埋め込みモデル  $m$  は，各トリプル  $(h, r, t)$  の妥当性を評価するためのスコア関数  $d_m(h, r, t)$  を定義する．

TransE は，知識グラフにおける関係  $r$  を，ヘッド  $h \in \mathbb{E}$  からテール  $t \in \mathbb{E}$  へのベクトル空間上での平行移動として解釈する埋め込みモデルである．エンティティと関係の埋め込みをそれぞれ  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$  としたとき，トリプル  $(h, r, t)$  に対して， $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$  が成立する．このとき，トリプルの妥当性を評価する距離ベースのスコア関数 (値が小さいほど妥当) は以下のように定義される．

$$d_{\text{TransE}}(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_p} \quad (1)$$

ここで， $\|\cdot\|_{L_p}$  は L1 または L2 ノルムを表す．

RotatE は，TransE が苦手とする複雑な関係パターンを表現するために，関係を複素ベクトル空間における回転としてモデル化している．エンティティを複素ベクトル  $\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$ ，関係を大きさが1の複素ベクトル  $\mathbf{r} \in \mathbb{C}^d$  ( $|\mathbf{r}_i| = 1$ ) で表現するとき，トリプル  $(h, r, t)$  に対して， $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$  が成立する．ここで  $\circ$  はアダマール積を意味する．スコア関数は以下のように定義される．

$$d_{\text{RotatE}}(h, r, t) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| \quad (2)$$

飯島らは，RotatE をベースに半径の拡大縮小を取り入れたモデルを提案した．本稿では飯島らのモデルを owlRotatE<sup>HAKÉ</sup> と表す．極形式で各ベクトルの  $j$  番目の要素を  $\mathbf{h}[j] = \mathbf{m}_h[j]e^{i\theta_h[j]}$ ， $\mathbf{t}[j] = \mathbf{m}_t[j]e^{i\theta_t[j]}$ ， $\mathbf{r}[j] = \mathbf{m}_r[j]e^{i\theta_r[j]}$  と表す．このとき，スコア関数は通常の RotatE との違いを明確にするため，角度と半径が分かれた極形式で以下のように定義される．

$$d_{\text{owlRotatE}^{\text{HAKÉ}}}(h, r, t) = \sum_{j=1}^d |\mathbf{m}_h[j]\mathbf{m}_r[j]e^{i(\theta_h[j]+\theta_r[j])} - \mathbf{m}_t[j]e^{i\theta_t[j]}| \quad (3)$$

ここで，リレーションの係数  $\mathbf{m}_r[j]$  は積によって半径の縮小を，角度  $\theta_r[j]$  は加算によって回転を表す．

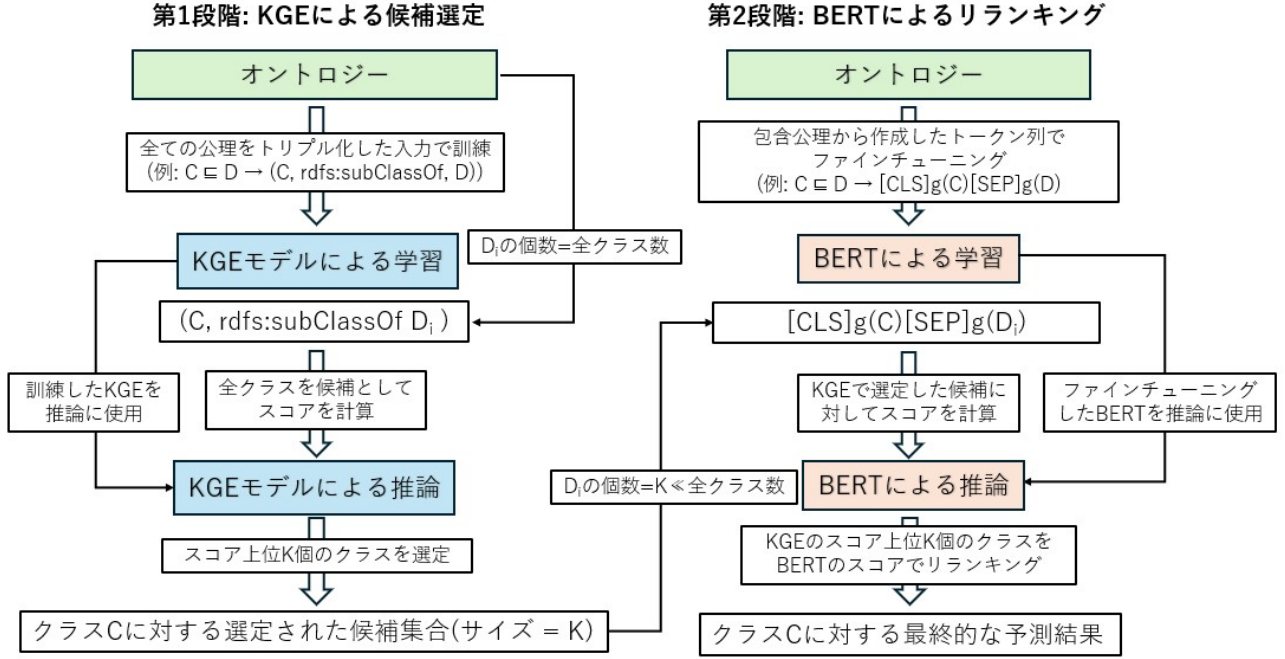


図 1: 提案するアーキテクチャの概要図

## 2.3 BERT によるオントロジー補完

BERT は, Transformer アーキテクチャに基づく大規模言語モデルである. Attention を用いた構造と, 双方向性の学習および大規模な事前学習コーパスによって, 高度な文章表現の獲得を可能とする. BERT への入力, 文全体の表現を集約する特殊トークン [CLS] と, 文の区切りを示す [SEP] を用いてフォーマットされる.

BERTSubs は, BERT を用いてオントロジーの補完を行う手法である. BERTSubs は, オントロジー内の包含関係  $C \subseteq D$  から  $C, D$  を記述するテキストラベルを入力トークン列に変換して BERT に入力し, [CLS] トークンに対応する最終層の出力ベクトルを分類器に適用することで,  $C$  と  $D$  の間の包含関係をスコアリングする. 事前学習済み BERT エンコーダと, 追加された分類器はクロスエントロピー損失によって最適化される.

BERTSubs は, BERT の持つ文脈理解能力と事前知識を活用することで高い予測精度を達成している. しかしその一方で, 計算コストが高く, 数万のクラスを持つような実用オントロジーに対して適用が難しいと考えられる.

## 3 二段階オントロジー補完

本研究では, 知識グラフ埋め込み (KGE) と大規模言語モデル (BERT) を組み合わせた, 効率と精度

を両立する二段階のオントロジー補完アーキテクチャを提案する. さらに, BERT の性能を最大化するため, KGE を用いたネガティブサンプリングと定義文を活用した入力トークン列の生成を提案する.

### 3.1 問題設定

本研究ではオントロジー補完タスクのうち, スーパークラス予測を扱う. オントロジーのクラスの集合を  $\mathbb{C}$  とする. あるクラス  $C \in \mathbb{C}$  が与えられたとき, 包含関係  $C \subseteq D$  が成立する  $D \in \mathbb{C} \setminus \{C\}$  を予測するタスクは, 包含関係の妥当性を示すスコア関数  $S(C \subseteq D)$  を最大化するスーパークラス  $D^*$  を求める問題として以下のように定式化できる.

$$D^* = \arg \max_{D_i \in \mathbb{C} \setminus \{C\}} S(C \subseteq D_i) \quad (4)$$

### 3.2 KGE と BERT によるリランキング

図 1 で示すように提案アーキテクチャは, (1) KGE モデルによる高速な候補選定, (2) BERT による高精度なリランキング, という二つの段階で構成される.

#### 3.2.1 KGE による候補選定

第 1 段階では, KGE モデルを用いて, 与えられたクラス  $C$  に対するスーパークラスの候補を, 自身以

外の全クラス集合  $\mathbb{C} \setminus \{C\}$  からより小さい候補集合へ選定する。

初めに KGE を用いてオントロジーを学習するために, TBox と ABox に含まれる各公理をエンティティ  $h, t (\in \mathbb{E})$  間の関係  $r (\in \mathbb{R})$  としてトリプルに変換する. 例えば包含関係  $C \sqsubseteq D$  や二項関係  $R(a, b)$  は,  $(C, \text{rdfs:subClassOf}, D)$ ,  $(a, R, b)$  というトリプルに変換される. 本研究では OWL2Vec\* と同様の変換規則を用いる. ただし, 関係が *owl:sameAs* であるものは, 学習の妨げになるため除外する. この変換規則によって得られたトリプルの集合を  $\mathbb{T}$  とする.

第 1 段階では, KGE モデルを使用してトリプル  $(h, r, t)$  の妥当性を評価する. KGE モデル  $m$  の各エンティティの埋め込みは, 以下の損失関数  $\mathcal{L}_{\text{KGE}}$  を最小化するように学習される.

$$\mathcal{L}_{\text{KGE}} = \sum_{(h,r,t) \in \mathbb{T}} \sum_{(h',r',t') \in \mathbb{T}'} \max(0, \gamma + d_m(h, r, t) - d_m(h', r', t')) \quad (5)$$

ここで,  $\gamma > 0$  はマージンである. また,  $\mathbb{T}'$  は正例トリプル  $(h, r, t)$  のヘッド  $h$  またはテール  $t$  をランダムに置換した負例トリプルの集合である.

オントロジーにより学習された KGE モデルを用いて, すべての候補クラス  $D_i \in \mathbb{C} \setminus \{C\}$  を評価して  $C$  のスーパークラスを予測する. このとき  $C \sqsubseteq D_i$  の妥当性を示すスコア  $d_m(C, \text{rdfs:subClassOf}, D_i)$  に基づき, 上位  $K$  個のクラスを第 2 段階の評価対象となる候補集合  $\mathbb{C}_{\text{top-}K}(C)$  として選出する.

$$\mathbb{C}_{\text{top-}K}(C) = \underset{D_i \in \mathbb{C} \setminus \{C\}}{\text{top-}K} \{d_m(C, \text{rdfs:subClassOf}, D_i)\} \quad (6)$$

ここで  $K$  はハイパーパラメータであり, 通常, 全クラス数  $|\mathbb{C}|$  よりもはるかに小さい定数値 ( $K \ll |\mathbb{C}|$ ) に設定される.

### 3.2.2 BERT によるリランキング

第 2 段階では, KGE で選定された  $\mathbb{C}_{\text{top-}K}(C)$  を対象として, BERT のファインチューニングによるリランキングを行う. 本研究では BERTSubs と同様に, オントロジーのスーパークラス予測タスクを一般的な二値分類タスクとみなす. オントロジーの包含関係  $C \sqsubseteq D$  を BERT で処理するため, 以下のようにトークン列を構築する.

$$x_{(C \sqsubseteq D)} = [\text{CLS}] g(C) [\text{SEP}] g(D) \quad (7)$$

ここで  $g(\cdot)$  は, クラスの IRI やラベルをトークン列に変換する関数である.

このトークン列  $x_{(C \sqsubseteq D)}$  を BERT に入力することで,  $[\text{CLS}]$  トークンに対応する最終層の出力ベクトル  $\text{BERT}(x_{(C \sqsubseteq D)}) \in \mathbb{R}^d$  を得る. このベクトル  $\text{BERT}(x_{(C \sqsubseteq D)})$  を MLP に通して得られる出力により,  $C \sqsubseteq D$  の妥当性を示すスコアを定義する.

$$S_{\text{BERT}}(C \sqsubseteq D) = \text{MLP}(\text{BERT}(x_{(C \sqsubseteq D)})) \quad (8)$$

BERT のファインチューニングは, 以下の損失関数により行われる.

$$\mathcal{L}_{\text{BERT}} = - \sum_{(C \sqsubseteq D) \in \mathcal{D}} \left[ y_{(C \sqsubseteq D)} \log(\sigma(S_{\text{BERT}}(C \sqsubseteq D))) + (1 - y_{(C \sqsubseteq D)}) \log(1 - \sigma(S_{\text{BERT}}(C \sqsubseteq D))) \right] \quad (9)$$

ここで  $\sigma(\cdot)$  はシグモイド関数,  $\mathcal{D}$  は訓練データの集合,  $y_{(C \sqsubseteq D)}$  は  $C \sqsubseteq D$  が正例なら 1, 負例なら 0 となるラベルである. この損失関数により, BERT のファインチューニングで正例のスコアを最大化し, 負例のスコアを最小化する.

$\mathbb{C}_{\text{top-}K}(C)$  に対するリランキングは,  $C$  と全ての候補スーパークラス  $D_i \in \mathbb{C}_{\text{top-}K}(C)$  との間の包含関係  $C \sqsubseteq D_i$  について, 式 (8) のスコアにより順序付けられる.  $S_{\text{BERT}}(C \sqsubseteq D_i)$  のスコアを最大化する  $D_i$  を最終的な  $C$  のスーパークラス  $D^*$  とする.

以上により 3.1 節の式 (4) における探索空間を  $\mathbb{C} \setminus \{C\}$  から  $\mathbb{C}_{\text{top-}K}(C)$  に置き換えられ, 計算コストの高い BERT の評価回数を  $|\mathbb{C}|$  から  $K$  にまで削減できる.

### 3.3 KGE によるネガティブサンプリング

本節では, BERT のファインチューニングで, 1 段階目の KGE の予測を用いたネガティブサンプリングを導入する. 訓練データ  $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$  のうち, 正例の集合  $\mathcal{D}^+$  は, オントロジーの TBox に存在する単純なクラスによる既知の包含関係  $C \sqsubseteq D$  の集合として定義される.

$$\mathcal{D}^+ = \{(C \sqsubseteq D) \in \text{TBox}\} \quad (10)$$

BERT では負例の集合  $\mathcal{D}^-$  を構築する際, TBox に存在しない包含関係を, 成り立たない負例として全てサンプリングする方が精度が良い. しかし  $\mathcal{D}^+$  と比べて  $\mathcal{D}_{\text{all}}^-$  のサイズが膨大になってしまう.

$$\mathcal{D}_{\text{all}}^- = \{C \sqsubseteq D' | (C \sqsubseteq D) \in \mathcal{D}^+, D' \in \mathbb{C} \setminus \{C, D\}\} \quad (11)$$

そのため先行研究の BERTSubs では 1 つの正例  $C \sqsubseteq D$  に対して,  $D$  をランダムに入れ替えた 1 つの  $D'$  のみ

表 1: OWL2Vec\*の設定におけるスーパークラス予測タスクの結果

|                                    | FoodOn       |              |              | GO           |              |              |
|------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                    | Hits@1       | Hits@5       | Hits@10      | Hits@1       | Hits@5       | Hits@10      |
| OWL2Vec*                           | 0.105        | 0.236        | 0.318        | 0.057        | 0.207        | 0.316        |
| TransE                             | 0.065        | 0.154        | 0.192        | 0.14         | 0.381        | 0.475        |
| RotatE                             | 0.108        | 0.184        | 0.21         | 0.191        | 0.447        | 0.528        |
| owlRotatE <sup>HAKE</sup>          | 0.307        | 0.523        | 0.595        | 0.199        | 0.403        | 0.482        |
| TransE-RoBERTa                     | 0.125        | 0.228        | 0.252        | 0.331        | 0.542        | 0.58         |
| RotatE-RoBERTa                     | 0.161        | 0.257        | 0.274        | 0.368        | 0.564        | 0.602        |
| owlRotatE <sup>HAKE</sup> -RoBERTa | <b>0.464</b> | <b>0.681</b> | <b>0.718</b> | <b>0.401</b> | <b>0.608</b> | <b>0.641</b> |

からなる負例の集合  $\mathcal{D}_{rand}^-$  を使用し,  $|\mathcal{D}^+| = |\mathcal{D}_{rand}^-|$  としている. しかしサイズの大きい  $\mathbb{C}$  からランダムに入れ替えた  $D'$  の大半は  $C$  と全く無関係なクラスで,  $C \sqsubseteq D'$  はいわゆる簡単すぎる負例となり, その学習は非効率的である.

KGE 予測によるネガティブサンプリングで, 各正例  $C \sqsubseteq D$  から負例  $C \sqsubseteq D'$  を作成する際の  $D'$  の母集団を  $\mathbb{C}$  から, 式 (7) で求めた  $\mathbb{C}_{\text{top-}K}(C)$  に変更する.

$$\mathcal{D}^- = \{(C \sqsubseteq D') \in \mathcal{D}_{all}^- | D' \in \mathbb{C}_{\text{top-}K}(C)\} \quad (12)$$

このサンプリング戦略により, BERT が類似したクラスの分類を効率的に学習できる.

### 3.4 定義文による入力トークン列の生成

スーパークラス予測タスクに BERT を適用する際, 式 (7) で示すように入力公理  $C \sqsubseteq D$  をから入力トークン列を作成する. BERTSubs では, 各クラスを表すトークンに `http://www.w3.org/2000/01/rdf-schema#label` でアノテーションされるテキストラベル (単語列) を入力していた.

BERT の表現能力を発揮するためには, 各クラスの意味内容を十分に表現した入力が必要である. クラスの IRI (例: `http://purl.obolibrary.org/obo/FOODON_03400819`) やラベル (例: “unleavened bread, crisp bread and rusk”) のみでは, 文脈情報が不足している. そこで, クラスのテキストラベルに加えて, `http://purl.obolibrary.org/obo/IAO_0000115` でアノテーションされるクラスの定義文を活用する. 例えば, 上述のクラス `http://purl.obolibrary.org/obo/FOODON_03400819` に対しては, 英単語 89 ワードからなる定義文が追加できる.

具体的には, 式 (7) の関数  $g(C)$  を, クラス  $C$  の定義文  $D(C)$  をトークン化したものとして定義する. ただし, 定義文のないクラスはテキストラベル  $L(C)$

表 2: 1 つのテストデータに対するスーパークラス予測の実行時間 (秒)

|                                    | FoodOn                | GO                    |
|------------------------------------|-----------------------|-----------------------|
| owlRotatE <sup>HAKE</sup>          | $5.51 \times 10^{-3}$ | $3.10 \times 10^{-3}$ |
| RoBERTa                            | $5.80 \times 10^2$    | $9.48 \times 10^2$    |
| RoBERTa( $K = 50$ )                | $1.01 \times 10^0$    | $1.06 \times 10^0$    |
| owlRotatE <sup>HAKE</sup> -RoBERTa | $1.01 \times 10^0$    | $1.06 \times 10^0$    |

をトークン化し, 両方ともないクラスは IRI をトークン化する.

$$g(C) = \text{Tokenize} \left( \begin{cases} D(C) & \text{if } D(C) \text{ is available} \\ L(C) & \text{else if } L(C) \text{ is available} \\ \text{IRI}(C) & \text{otherwise} \end{cases} \right) \quad (13)$$

ここで  $\text{Tokenize}(\cdot)$  は, BERT の辞書に基づき, テキストをトークン列に変換する関数である.

IRI やクラスラベルよりも自然言語の文章に近い定義文を活用することで, BERT が事前学習で獲得した高度な文脈知識を十全に発揮できる.

## 4 実験

オントロジー GO と FoodOn を対象に, スーパークラス予測タスクにおける提案手法の性能を評価する. GO は遺伝子情報に関するオントロジー, FoodOn は食べ物に関するオントロジーである.

### 4.1 実験設定

OWL2Vec\* と BERTSubs のそれぞれと同様の設定で各オントロジーの実験を行った. OWL2Vec\* の設定は, オントロジー内の包含関係を訓練 (70%), 検証 (10%), テスト (20%) に分割し, 検証データとテストデータにおいて予測を行う際は自身以外の全クラスを予測候補とする. 一方, BERTSubs の設定は,

表 3: BERTSubs の設定におけるスーパークラス予測タスクの結果

|                                    | FoodOn       |              |              | GO           |              |              |
|------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                    | Hits@1       | Hits@5       | Hits@10      | Hits@1       | Hits@5       | Hits@10      |
| BertSubs(IC)                       | 0.483        | 0.832        | 0.931        | 0.408        | 0.825        | <b>0.937</b> |
| BertSubs(PC)                       | 0.491        | 0.829        | 0.932        | 0.453        | 0.806        | 0.927        |
| BertSubs(BC)                       | 0.459        | 0.824        | 0.935        | 0.429        | 0.767        | 0.907        |
| owlRotatE <sup>HAKE</sup> -RoBERTa | <b>0.680</b> | <b>0.902</b> | <b>0.940</b> | <b>0.640</b> | <b>0.874</b> | 0.901        |
| -w/o KGE ネガティブサンプリング               | 0.483        | 0.843        | 0.917        | 0.491        | 0.830        | 0.891        |
| -w/o 定義文の追加                        | 0.678        | 0.899        | 0.935        | 0.590        | 0.846        | 0.893        |

オントロジー内の包含関係を訓練 (80%), 検証 (5%), テスト (15%) に分割して使用し, 検証データとテストデータにおいて予測候補を最大 50 個までに制限している. 予測候補は, 正解クラスとその近傍クラスから選択され, テストデータにおいては, FoodOn が平均 36.2 個, GO が平均 44.9 個である.

評価指標として, Hits@ $n$  ( $n = 1, 5, 10$ ) を用いる. Hits@ $n$  は予測結果の上位  $n$  件に正解のスーパークラスが含まれていた割合を示す. また計算効率の評価指標として, 1つのテストデータに対するスーパークラスの予測の実行時間を計測する.

各 KGE モデルについては, 埋め込み次元  $d \in \{50, 100, 200\}$ , マージン  $\gamma \in \{1, 6, 12\}$ , 正例 1 に対する負例の数  $n_{neg} \in \{1, 8, 16\}$  について, グリッドサーチによるハイパーパラメータチューニングを行った. 知識グラフ埋め込みモデルを用いて選定する候補数  $K$  は, OWL2Vec\* の設定では  $K=50$ , BERTSubs の設定では  $K=25$  とした. なお本研究では BERT ベースエンコーダとして RoBERTa[7] を採用している.

## 4.2 実験結果

OWL2Vec\* の設定による実験結果を表 1 に示す. 既存手法の中で最も精度の高い owlRotatE<sup>HAKE</sup> の推論結果を, 提案アーキテクチャに基づきランキングしたときが, 全指標で最も高い精度を記録した. また第 1 段階モデルの精度が比較的高くない TransE と RotatE についても, 精度の伸びは小さいが, リランキングによって改善している. このことから提案アーキテクチャは, KGE モデルによる候補選定の精度が良いほどリランキングの効果が高くなる.

1つのテストデータに対してスーパークラス予測を行う際の各モデルの実行時間を表 2 に示す. 提案アーキテクチャによって実行時間が FoodOn では約  $\frac{1}{574}$ , GO では約  $\frac{1}{903}$  倍になっており, BERT を用いたオントロジー補完の計算コストを劇的に削減させている. また, FoodOn のクラス数は 28182, GO のクラス数は 44244 であり, 従来の BERT による予測の実行時

間はクラス数に応じて長くなっている. 一方, 提案アーキテクチャは 2つのオントロジーの実行時間がほぼ等しく, 予測候補の選定によって, オントロジーのクラス数に関係なく計算量が一定に保たれていることがわかる. また, KGE モデルの owlRotatE<sup>HAKE</sup> は非常に高速で, 候補選定の計算コストは低いことがわかる.

BERTSubs の設定による実験結果を表 3 に示す. 両方のオントロジーにおいて, 多くの指標で提案手法が BERTSubs を上回る精度を記録した. また, KGE モデルの予測によるネガティブサンプリングが精度向上に大きく寄与している. 定義文の追加に関しては, FoodOn では精度への影響が小さく, GO では精度の向上に寄与している. これは, GO は全てのクラスに定義文が存在するが, FoodOn は約 4 割のクラスにしか定義文が存在していないことが原因と考えられる.

## 5 まとめ

本研究では, オントロジーの包含関係予測タスクに対し, KGE モデルによる候補選定と BERT によるリランキングを組み合わせた, 効率と精度を両立する二段階のアーキテクチャを提案した. また BERT によるリランキングの精度向上のため, KGE モデルの予測を用いたネガティブサンプリングと, 定義文を活用した入力トークン列の生成を提案した. 評価実験では, 2つのオントロジーにおけるスーパークラス予測において, 提案手法が既存手法の精度を上回ることを示した. また, KGE モデルによる候補選定が BERT を用いたスーパークラス予測の計算コストを大幅に削減することを示した. 今後の展望としては, 他のオントロジー補完タスクへの適用や, 複雑なクラスを含む公理の活用などが考えられる.

## 参考文献

- [1] S Bechhofer *et al.* OWL Web Ontology Language Reference. Technical report, World Wide

Web Consortium (W3C), 2004.

- [2] A Bordes *et al.* Translating embeddings for modeling multi-relational data. *NIPS'13*, page 2787–2795, 2013.
- [3] J Chen *et al.* Owl2vec\*: embedding of owl ontologies. *Machine Learning*, 110:1813–1845, 2021.
- [4] J Chen *et al.* Contextual semantic embeddings for ontology subsumption prediction. *World Wide Web*, 2023.
- [5] J Devlin *et al.* 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*, pages 4171–4186, 2019.
- [6] M Kulmanov *et al.* El embeddings: Geometric construction of models for the description logic el++. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 2019.
- [7] Y Liu *et al.* Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [8] T Mikolov *et al.* Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.
- [9] Z Sun *et al.* Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.
- [10] Z Zhang *et al.* Learning hierarchy-aware knowledge graph embeddings for link prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:3065–3072, 2020.
- [11] 飯島彩友. オントロジーの論理構造と語彙情報を考慮した概念階層の埋め込み学習. 2025.