

Graph Neural Networkを用いた未知エンティティの表現獲得について

濱口 拓男^{1,a)} 大岩 秀和^{2,b)} 新保 仁^{1,c)} 松本 裕治^{1,d)}

概要：知識ベース補完は知識ベース内で欠落している情報を推定するタスクである．本論文では知識ベース補完における未知 entity 問題を扱う．未知 entity 問題とは，訓練データに含まれない entity が予測時に与えられた場合，その entity に関する知識をどのようにして獲得するのかという問題である．埋め込みに基づいた知識ベース補完の既存手法は，予測時に与えられる entity はすべて訓練データ中に出現している仮定を置いているため，未知 entity の表現をどう獲得するかは明らかではなかった．今回我々は再度モデルを学習するということなく，この未知 entity 問題を解決する．具体的には，未知 entity に関する補助的な知識を用いることで知識グラフ上の Graph neural network を構築し，既存の表現から知識を転用することで未知 entity に関する表現を得る．我々は未知 entity の実験において提案する手法の効果を示した．また WordNet データを用いた標準的な知識ベース補完の設定の下でも，先行研究に比較して良い精度を示した．

TAKUO HAMAGUCHI^{1,a)} HIDEKAZU OIWA^{2,b)} MASASHI SHIMBO^{1,c)} YUJI MATSUMOTO^{1,d)}

1. 序論

WordNet [15] や Freebase [1] といった知識ベースは，情報抽出や質問応答，文章理解など様々な応用で活用されている．我々は，多くの先行研究と同じように，2 項関係によって構成される知識ベースに注目する．2 項関係とは 3 つ組 (h, r, t) で表されるデータであり， h, r, t はそれぞれ head entity，関係，tail entity である．このような 3 つ組の例として，*(Philip-K.-Dick, write, Do-Androids-Dream-of-Electric-Sheep?)* や *(Do-Androids-Dream-of-Electric-Sheep?, is-a, Science-fiction)* を挙げることができる．知識ベースは数多くの 3 つ組を含むが，しかし全ての情報が記録されてはならず，その知識は不完全であることが知られている [17]．知識ベース補完は，それら知識ベースに欠落した情報の予測を目標とするものである．

本論文において，我々は知識ベース補完における未知 entity 問題という，新しい問題に対する解法を提案する．この問題は，新しく観測され知識ベースに存在しない entity (未

知 entity) に関する問い合わせが予測時に与えられた場合に，これをどのように扱うかという問題である．現実世界においてそれまでに存在しなかった新しい entity に関する知識が求められる場面は決して少なくない．例えば知識ベースをどのように更新するのかという問題は未知 entity に関する問題である．どのように日々生産され続けている新しい entity を扱うのかという問題は情報抽出において重要な問題である．

本論文で扱われる未知 entity の問題がどのようなものであるかを，次の事例を挙げて説明する．我々は新しい entity “Blade-Runner” を新しい 3 つ組 *(Blade-Runner, based-on, Do-Androids-Dream-of-Electric-Sheep?)* の中に見つけたとする．このように我々の設定では，未知の entity Blade-Runner は新しい知識の一部として与えられる．我々は，新しい entity に関する質問，例えば “Is Blade Runner a science fiction?” のような質問に答えたい．これは 3 つ組 *(Blade Runner, is-a, science fiction)* が正しいかどうかを識別する問題と同じである．直感的な話として，既存の知識ベースが 3 つ組 *(Do-Androids-Dream-of-Electric-Sheep?, is-a, Science-fiction)* を含んでいるのであれば，上記の質問の答えは Yes であることが推定できる．つまり，2 つの 3 つ組 *(Blade-Runner, based-on, Do-Androids-Dream-of-Electric-Sheep?)* と *(Do-Androids-Dream-of-Electric-Sheep?, is-a, Science-fiction)* を含んでいるのであれば，上記の質問の答えは Yes であることが推定できる．つまり，2 つの 3 つ組 *(Blade-Runner, based-on, Do-Androids-Dream-of-Electric-Sheep?)* と *(Do-Androids-Dream-of-Electric-Sheep?, is-a, Science-fiction)* を含んでいるのであれば，上記の質問の答えは Yes であることが推定できる．

¹ 奈良先端科学技術大学院大学 Nara Institute of Science and Technology

² Recruit Institute of Technology

^{a)} takuo-h@is.naist.jp

^{b)} oiwa@recruit.ai

^{c)} shimbo@is.naist.jp

^{d)} matsu@is.naist.jp

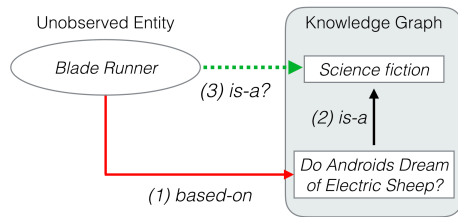


図 1: 未知 entity 問題の例示．赤い実矢印 (1) で示される 3 つ組が予測時に利用可能な新しい知識である．その 3 つ組は、知識ベースには存在しない entity “Blade Runner” を含んでおり、その未知の entity に関する知識をどのように抽出するかが本論文の主題である．この例では、黒の実矢印 (2) で示される知識 (*Blade Runner, is-a, science fiction*) が既存の知識ベースに含まれていると仮定されている．ただし既存の知識ベースは灰色の領域で表され、内部の entity はそれに含まれるとする．この既知の知識 (2) と新しい entity に関連する知識 (1) を用いることによって、例えば緑の破線矢印で示される問 “Is Blade Runner a science fiction?” の答えが Yes だと推測できる．

is-a, Science-fiction) から、(*Blade Runner, is-a, science fiction*) の情報が推定できる．このように、我々は未知の entity に関する情報も間接的に推定することが可能である．図 1 はこの例を図示したものである．

一般的な問題として、我々は既存の知識から未知 entity へどう知識の転移をすればよいかという問題に直面する．この問題の解決方法の 1 つは、複雑で反復的なシステム [7], [8] を構築するである．しかし、これらのモデルはウェブ情報のクローリングや人手でのアノテーションといった外部リソースを必要とするためコストが高い．またそれらモデルが置いている前提と今回の知識ベース補完は性質の違ったものである．本論文で扱う知識ベース補完は知識ベースのみを用いて欠落した知識を補完するものである．そして興味深いことに、上で挙げた *Blade Runner* の例は、そのような追加的なコストをかけずに新しい entity に関する知識を獲得できることを示唆している．このような問題、つまりどのように未知 entity を扱うべきなのかや、どのように既知の知識ベースから未知の entity に知識を転移すればいいのかという問題が、本論文で扱う未知 entity 問題である．

近年、分散表現を用いた埋め込みに基づく手法 [2], [22] が知識ベース補完で提案されている．埋め込みに基づく手法は暗黙的な情報を扱えるなどの利点を持ち、実験的にも良い性能を示すことが報告されている．これらの手法は予測すべき 3 つ組の entity が訓練データに含まれることを仮定しており、その訓練で学習された結果を欠落した情報の予測に用いている．この、事前に全ての entity を訓練できるという仮定を置いているために、それら先行研究では訓練データに含まれない未知の entity をどう扱うかが明らかではない．この問題は特に埋め込みに基づく手法では避け

がたい問題である．我々は埋め込みに基づく手法の利点を活用でき、かつ未知 entity に関する問題を解決する手法を提案する．

我々は前述の解決するために、Graph Neural Networks(Graph-NN) [13], [18] を知識グラフに適用した手法を提案する．知識グラフとは知識ベースに含まれる 3 つ組 (h, r, t) から構成されるラベル付き有向グラフであり、グラフの辺は節点 t から節点 h に向かう、ラベル r を持つものとして定義される．また Graph-NN とはグラフ上に定義された neural network の一種であり、伝播モデルと出力モデルの 2 モデルから構成される．伝播モデルは、既存の知識ベースから未知 entity への知識の転移を可能にすることで未知 entity の表現を獲得可能にするものである．より具体的に言えば、節点 e の表現ベクトルは、 e の近傍にある節点の埋め込みベクトルを neural network を用いて変換し、変換されたベクトルを集めた集合を平均や max などの演算を用いてプールして 1 つのベクトルに変換することで獲得される．この、各 entity の表現が近傍の表現ベクトルから計算される仕組みによって、未知 entity の表現ベクトルも計算することができる．もう 1 つのモデルである出力モデルは、それら伝播モデルによって獲得された表現ベクトルを用い、タスクに応じた目的関数を定めるものである．どういう目的関数を設定するかには自由度があり、本論文の目的である知識ベース補完のため、先行研究の知識ベース補完のモデルをそのまま用いることが可能である．本論文では最も基本的な手法の 1 つである TransE [2] を出力モデルとして用いる．

- 我々の主たる貢献は次のようにまとめることができる．
- 知識ベース補完における未知 entity 問題を定式化した．
 - 未知 entity を扱う新しい Graph-NN を提案した．
 - 未知 entity 問題と通常の設定で提案手法の有用性を示した．

2. 知識ベース補完における未知 entity 問題

2.1 知識グラフ

entity の集合 $\mathcal{E} = \mathcal{E}(G)$ と関係の集合 $\mathcal{R} = \mathcal{R}(G)$ が与えられたとき、知識ベース G は $h, t \in \mathcal{E}, r \in \mathcal{R}$ であるような 3 つ組 (h, r, t) の集合として定義される．ここで \mathcal{E} に含まれる全ての entity と \mathcal{R} に含まれる全ての関係は、 G 中の 3 つ組の中に少なくとも一度は含まれるとする．知識ベースはしばしば知識グラフと呼ばれる．それは、個々の 3 つ組内の entity を節点に、関係を辺のラベルと見做すことで、3 つ組の集合に対応するラベル付き有向グラフを構成できるためである．

2.2 知識ベース補完: 3 つ組分類

3 つ組分類は、先行研究である [19] によって紹介された典型的な知識ベース補完のタスクであり、標準的な評価手

法として用いられている [16], [20] .

3 つ組分類では知識ベースは不完全であることが仮定されている . この不完全性のために , 知識ベースに含まれない 3 つ組には 2 種類のもの存在する . 1 つは本来は単純に正しくない知識であるため知識ベースに記録されていないものであり , もう 1 つは正しい知識であるが , 知識ベースの不完全性ゆえに正しいけれど存在しないものである . この結果として , 知識ベースに含まれない 3 つ組が与えられた時 , それどちらに所属するかは分からないという問題が発生する . 3 つ組分類は , 正しいかどうか不明な 3 つ組がどちらに所属する 3 つ組かを識別する問題である . つまり機械学習の問題として見れば , 3 つ組分類は訓練データとテストデータに分けられた 3 つ組の集合を用いた分類タスクであると言える .

3 つ組分類の最終的な目標は完全な知識ベース $\mathcal{G}_{\text{gold}} \supset \mathcal{G}$ を特定することである . ここで $\mathcal{G}_{\text{gold}}$ は観測された \mathcal{G} の entity と関係の全ての可能な組み合わせの部分集合である . これは $\mathcal{G}_{\text{gold}}$ の全ての entity と関係は \mathcal{G} で既に観測されたものであることを意味し , 3 つ組分類を解決することは本研究の目的である既知の知識ベースから欠落した 3 つ組を補完することを意味する .

2.3 未知 entity 問題

我々は未知 entity 問題と呼ぶ新しい知識ベース補完のタスクを紹介する .

この問題設定では 3 種類の 3 つ組の集合 , 訓練データ , テストデータ , 補助データ \mathcal{G}_{aux} が存在する . まず $\mathcal{E}(\mathcal{G}_{\text{aux}}) \not\subseteq \mathcal{E}(\mathcal{G})$ であり , 補助データに含まれる 3 つ組には既存の知識ベースに含まれない entity が存在する . 未知 entity の集合を , これら表記を用いて $\mathcal{E}_{\text{unobserved}} = \mathcal{E}(\mathcal{G}_{\text{aux}}) \setminus \mathcal{E}(\mathcal{G})$ と定義する . また $\mathcal{R}(\mathcal{G}_{\text{aux}}) \subseteq \mathcal{R}(\mathcal{G})$ であり , 新しい関係性は登場しないものとする . 訓練データは 1 つも未知 entity を含まない 3 つ組から構成され , テストデータは最低 1 つは未知 entity を含む 3 つ組から構成される . また補助データ \mathcal{G}_{aux} に含まれる 3 つ組は正確に 1 つの未知 entity $\mathcal{E}_{\text{unobserved}}$ と , 既知の entity $\mathcal{E}(\mathcal{G})$ を含んでいるとする . この補助データは予測時にのみ与えられるとする . 補助データは \mathcal{G}_{aux} は , $\mathcal{E}(\mathcal{G})$ と $\mathcal{E}_{\text{unobserved}}$ を橋渡しする役目を持つ . つまり補助的な知識 \mathcal{G}_{aux} を手がかりとして既知の知識 \mathcal{G} から知識を転移し , 未知の entity $\mathcal{E}_{\text{unobserved}}$ に関する知識を獲得するタスクを未知 entity 問題と呼ぶ .

3. 提案モデル

3.1 Graph Neural Networks

Graph Neural Network(Graph-NN) はグラフ構造の上に定義された neural network である . Graph-NN の中にはグラフ全体を 1 つのベクトルに変換するモデルも存在するが [3], [4], 我々は節点や辺それぞれをベクトルに変換する手

法に注目する . これは知識ベース補完により適しているためである .

先行研究である [13], [18] に従えば , Graph-NN は伝播モデルと出力モデルの 2 モデルから構成される . 伝播モデルは節点の間でどのように情報が伝播するかを決め , 出力モデルはベクトル表現された節点や辺を用いてタスクに応じた目的関数を定義する . 本論文で我々は伝播モデルを知識グラフに適応するように改変し , 出力モデルを埋め込みに基づく知識ベース補完の手法である TransE [2] を用いたものに改変する .

3.2 知識グラフ上の伝播モデル

\mathcal{G} を知識グラフ , $e \in \mathcal{E}(\mathcal{G})$ を entity , $\mathbf{v}_e \in \mathbb{R}^d$ を d 次元で表現された entity e の表現ベクトルとする . 先行研究である [13] は伝播モデルを次式で定義する .

$$\mathbf{v}_e = \sum_{(h,r,e) \in \mathcal{N}_{\text{head}}(e)} T_{\text{head}}(\mathbf{v}_h; h, r, e) + \sum_{(e,r,t) \in \mathcal{N}_{\text{tail}}(e)} T_{\text{tail}}(\mathbf{v}_t; e, r, t) \quad (1)$$

ここで head の近傍 $\mathcal{N}_{\text{head}}$ と tail の近傍 $\mathcal{N}_{\text{tail}}$ はそれぞれ , 知識グラフ \mathcal{G} における $\mathcal{N}_{\text{head}}(e) = \{(h, r, e) \mid (h, r, e) \in \mathcal{G}\}$ と $\mathcal{N}_{\text{tail}}(e) = \{(e, r, t) \mid (e, r, t) \in \mathcal{G}\}$ である . また $T_{\text{head}}, T_{\text{tail}} : \mathbb{R}^d \times \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}^d$ は遷移関数であり , 節点間の関係に基づいて近傍節点のベクトルを変換するものである .

我々はこの伝播モデルを , pooling 関数を用いて次のように一般化する .

$$S_{\text{head}}(e) = \{T_{\text{head}}(\mathbf{v}_h; h, r, e) \mid (h, r, e) \in \mathcal{N}_h(e)\}, \quad (2)$$

$$S_{\text{tail}}(e) = \{T_{\text{tail}}(\mathbf{v}_t; e, r, t) \mid (e, r, t) \in \mathcal{N}_t(e)\}, \quad (3)$$

$$\mathbf{v}_e = P(S_{\text{head}}(e) \cup S_{\text{tail}}(e)), \quad (4)$$

ここで $S_{\text{head}}(e)$ と $S_{\text{tail}}(e)$ はそれぞれ , 近傍の表現ベクトル $\mathcal{N}_{\text{head}}(e)$ と $\mathcal{N}_{\text{tail}}(e)$ を含むものである . 先行研究の式 (1) と我々の式 Eqs. (2)–(4) の違いは , 総和を取る箇所に pooling 関数を使うかどうかであり , pooling 関数に sum を用いれば先行研究のモデルと一致する . 以下では , どのような遷移関数 $T_{\text{head}}, T_{\text{tail}}$ や pooling 関数 P を用いることができ , 我々がどの遷移関数と pooling 関数を用いるかを説明する .

3.2.1 遷移関数

遷移関数 T_{head} と T_{tail} は , 近傍の節点と現在の節点にある関係性を表現ベクトルに反映することである . まず関係性に依存しない遷移関数として次のようなものを挙げることができる .

$$T(\mathbf{v}) = \mathbf{v}, \quad (\text{Identity})$$

$$T(\mathbf{v}) = \tanh(\mathbf{A}\mathbf{v}), \quad (\text{single tanh layer})$$

$$T(\mathbf{v}) = \text{ReLU}(\mathbf{A}\mathbf{v}), \quad (\text{single ReLU layer})$$

ここで $\mathbf{A} \in \mathbb{R}^{d \times d}$ はモデルのパラメータである行列であ

り, \tanh と ReLU は要素ごとの hyperbolic tangent と rectified linear unit 関数である. これらに加えて, 他の neural network の技術, 例えば batch-normalization [9] や residual connection [6] といったものを用い, より複雑な遷移関数を設計することも可能である.

上記の遷移関数を拡張し, 3 つ組に依存した次のような遷移関数を考えることができる.

$$T_{\text{head}}(\mathbf{v}_h; h, r, e) = \tanh(\mathbf{A}_{(h,r,e)}^{\text{head}} \mathbf{v}_h),$$

$$T_{\text{tail}}(\mathbf{v}_t; e, r, t) = \tanh(\mathbf{A}_{(e,r,t)}^{\text{tail}} \mathbf{v}_t).$$

ここでパラメータ行列 \mathbf{A} は節点 (entity) e , 近傍 h もしくは t , それらの関係 r に依存するものになっている.

実験では次の遷移関数を用いる.

$$T_{\text{head}}(\mathbf{v}_h; h, r, e) = \text{ReLU}(\text{BN}(\mathbf{A}_r^{\text{head}} \mathbf{v}_h)) \quad (5)$$

$$T_{\text{tail}}(\mathbf{v}_t; e, r, t) = \text{ReLU}(\text{BN}(\mathbf{A}_r^{\text{tail}} \mathbf{v}_t)) \quad (6)$$

ただし BN は batch normalization [9] を表す.

3.2.2 Pooling 関数

Pooling 関数 P はベクトルの集合からベクトルへの写像を提供する関数であり, 数式としては $P: 2^{\mathbb{R}^d} \rightarrow \mathbb{R}^d$ となる. Pooling 関数の目的はベクトルの集合から共通している性質を抽出することである. ベクトルの集合を $S = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ とした時, Pooling 関数の例として次の関数を挙げるができる.

$$P(S) = \sum_{i=1}^N \mathbf{x}_i, \quad (\text{sum pooling})$$

$$P(S) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i, \quad (\text{average pooling})$$

$$P(S) = \max(\{\mathbf{x}_i\}_{i=1}^N), \quad (\text{max pooling})$$

ここで \max は要素ごとの \max を取る関数である. Sum pooling は式 (1) で書いた通り, 先行研究 [13], [18] で用いられているものである. 他にも重み付き和や attention 機構を用いることも可能である.

3.3 出力モデル: スコア関数と目的関数

我々は出力モデルに TransE [2] に基づいた目的関数を用いる. TransE は埋め込みに基づいた知識ベース補完手法の 1 つであり, 単純性と訓練のしやすさの利点を持つ.

この小節では, まず我々は TransE で用いられるスコア関数と目的関数に関して説明を行い, 次にどう変更したかを説明する.

3.3.1 スコア関数

スコア関数は 3 つ組の不自然さ (implausibility) を評価するものであり, 小さいスコアであればその 3 つ組がより自然らしいことを意味する. TransE におけるスコア関数は $f(h, r, t) = \|\mathbf{v}_h + \mathbf{v}_r - \mathbf{v}_t\|$ で定義される. ただし \mathbf{v}_h

と $\mathbf{v}_r, \mathbf{v}_t$ はそれぞれ head · entity, 関係, tail · entity の埋め込みベクトルである. このスコア関数は, 正しい 3 つ組 (h, r, t) の場合, head entity と関係のベクトルの和 $\mathbf{v}_h + \mathbf{v}_r$ が tail entity のベクトル \mathbf{v}_t の近くに来て欲しいことを意味する. このスコア関数の拡張方法に関しては多くの研究がなされている [20], [22].

3.3.2 目的関数: margin-based

目的関数は正しい 3 つ組のスコアとそうでない 3 つ組のスコアを組み合わせて定義される.

次の目的関数は広く使われており [2], [22], 簡便化のため本論文では margin-based の目的関数と呼ぶ.

$$\mathcal{L} = \sum_{i=1}^N [\tau + f(h_i, r_i, t_i) - f(h'_i, r_i, t'_i)]_+ \quad (7)$$

ただし $[x]_+$ はヒンジ関数 $[x]_+ = \max(0, x)$ であり, スカラ $\tau \in \mathbb{R}_+$ は 0 以上の閾値である. また (h_i, r_i, t_i) は正しい 3 つ組を, (h'_i, r_i, t'_i) はそうでない 3 つ組を表す. この目的関数は, (h'_i, r_i, t'_i) のスコアが (h_i, r_i, t_i) のスコアより τ 以上大きいことを要求する. これは最適化において, 差分が $-\tau$ 以下の場合にはパラメータの更新が発生し, 逆にその差分が $-\tau$ 以下であれば損失は 0 になるのでパラメータの更新は行われなためである. 例えば TransE の場合, 距離を用いた定義からスコアは 0 以上であるため, 基本的に正しい 3 つ組のスコアは 0 に近づけられる. しかし 3 つ組 (h'_i, r_i, t'_i) のスコアが十分大きいものであった場合, 正しい 3 つ組 (h_i, r_i, t_i) のスコアはある程度大きくても目的関数には影響を及ぼさない.

3.3.3 目的関数: region-based

我々は前述の margin-based の目的関数の代わりに次の目的関数を用いる.

$$\mathcal{L} = \sum_{i=1}^N f(h_i, r_i, t_i) + [\tau - f(h'_i, r_i, t'_i)]_+ \quad (8)$$

簡便化のためにこの関数を本論文では region-based の目的関数と呼ぶ. この目的関数は 3 つ組の差に注目するのではなく, 正しい 3 つ組とそうでない 3 つ組のスコア, それぞれに注目する. この変更の結果として, 正しい 3 つ組のスコアは, そうでない 3 つ組がなんであれ, なるべく 0 になるように学習され, また正しくない 3 つ組も正しい 3 つ組がなんであれ, 最低 τ は値を持つように学習される.

3.4 実装とハイパーパラメータについて

我々は提案手法を neural network ライブラリの 1 つである Chainer^{*1} を用いて実装した. ネットワーク全体の訓練には, 誤差逆伝播法を用い, 確率的勾配法の一つである Adam [12] を用いた. Adam の step size は各 epoch に対し, $\frac{\alpha_1}{\alpha_2 * epoch + 1.0}$ で計算した. ただし $\alpha_1 = 0.01, \alpha_2 = 0.0001$

^{*1} <http://chainer.org/>

表 1: 3 つ組分類で用いられるデータセットの計量的な性質．検証データとテストデータの半分は間違いの 3 つ組で構成される．

	WN11	FB13
関係の種類	11	13
entity の種類	38,696	75,043
訓練データ数	112,581	316,232
検証データ数	5,218	11,816
テストデータ数	21,088	47,466

表 2: 標準的な知識ベース補完の実験の結果 (未知 entity を含まない)．それぞれの数値は精度を表す．提案手法を除いた他の数値はそれぞれの論文から写したものである．太字体と下線の数値は，それぞれ各データセットでの最も良いスコアと次に良いスコアを表している．

method	WN11	FB13
NTN [19]	70.4	87.1
TransE [2]	75.9	81.5
TransH [20]	78.8	83.3
TransR [14]	85.9	82.5
TransD [10]	86.4	89.1
TransE-COMP [5]	80.3	87.6
TranSparse [11]	86.8	88.2
ManifoldE [21]	<u>87.5</u>	87.3
TransG [22]	87.4	87.3
lppTransD [23]	86.2	<u>88.6</u>
TransE-NMM [16]	86.8	<u>88.6</u>
proposed method	87.8	81.6

である．ミニバッチサイズは 5,000，訓練 epoch は 300，を全ての実験で用いた．モデルの次元は，標準的な 3 つ組分類では 200 次元を用い，それ以外では 100 次元を用いた．

予備実験において，我々は幾つかの活性化関数と pooling 関数を試し，計算コストとパフォーマンスの観点から次のハイパーパラメータ設定を見出した．我々は ReLU 関数を活性化関数にし，max 関数を pooling 関数に用いて標準的な 3 つ組分類を行った．これら予備実験の結果は margin-based の目的関数ではなく region-based の目的関数を選んだことを反映している．これは region-based の目的関数が margin-based の目的関数と比べより速い収束を示したためである．また式 (8) における閾値を $\tau = 300$ とし設定した．我々は検証データを用い，スコアから正しい 3 つ組とそうでない 3 つ組を分類するための基準を決めた．

第 2 章で説明したように，知識ベースに含まれていない 3 つ組が，単純に知識ベースの不完全性によるものか，或いはその 3 つ組が間違っ知識であるため知識ベースに含まれていないかは明らかではない．そのため，正しくない 3 つ組として，確実に間違っている 3 つ組を用いるのは難しい．この問題を緩和するため，正しい 3 つ組の一部をランダムに置き換えることによって粗悪化された 3 つ組を製作し，

間違いである 3 つ組の代わりに用いる技法が一般に用いられている．我々は特にサンプル時に Bernoulli trick という技法を用いた．これは高頻出の entity は低いサンプル率を，反対に低頻出の entity には高いサンプル率を割り当てる技法であり，最初 [20] で紹介され，後の研究 [10], [11], [14] で用いられているものである．

GPU メモリなどの利用可能な計算リソースの制限のため，我々は近傍の数が多すぎる場合はそれらの中から一定の個数をサンプルして用いた．実際，*male*，*female* などの幾つかの entity は非常に多くの 3 つ組に含まれ，結果としてその entity は多くの近傍を持つ．近傍の数が 64 を超える場合は我々は 64 個をランダムに選んで用いた．

4. 実験

4.1 標準的な 3 つ組分類

4.1.1 データセット

本研究では，我々は先行研究である [19] によって紹介された WordNet11 (WN11) と Freebase13 (FB13) を評価に用いた．データのファイルはサイト <http://ml.knu.ac.kr/lppKE/> からダウンロードした．このデータセットは一般的な知識グラフである WordNet [15] と Freebase [1] から構成される．またそれぞれの計量的な性質は表 1 にまとめられている．両方のデータセットとも，訓練データと検証データ，テストデータを含み，検証データとテストデータは正しい 3 つ組と間違いの 3 つ組を含む．訓練データは正しくない 3 つ組を含まないため粗悪化した 3 つ組を作る必要があり，そのために我々は Bernoulli trick を用いて粗悪化した 3 つ組を生成した．これは第 3 章の装束とハイパーパラメータで説明した技法である．

4.1.2 結果

本実験では提案手法と既存の知識ベース補完の手法とを比較する．特にこの実験は未知 entity を伴わない標準的な 3 つ組分類の設定である．結果は表 2 に記載した．我々の手法は WordNet11 のデータセットにおいて最も良い精度を示した．しかし Freebase では十分良い精度を示してはいない．我々はこのパフォーマンスの違いは，それぞれのデータセットにおける知識の転移の難しさに起因すると思われる．WordNet11 は *similar-to* のような推移的な推測が成り立つ関係を比較的多く含む．反対に Freebase13 は *cause-of-death* のような知識の転移が難しい関係性を含んでいる．

4.2 未知 entity 実験

4.2.1 データセット

我々は未知 entity の実験のために，WordNet11 を用いて 9 つのデータセットを新たに作成した．データセットの作成手続きは，以下で説明されるような未知 entity の選択と，データの分割の 2 つの手続きから構成される．

表 3: 未知 entity のデータセットに含まれる entity と 3 つ組の数 . 3 つ組の数は間違っ た 3 つ組を含む .

	Head			Tail			Both		
	1000	3000	5000	1000	3000	5000	1000	3000	5000
訓練データ数	108,197	99,963	92,309	96,968	78,763	67,774	93,364	71,097	57,601
検証データ数	4,613	4,184	3,845	3,999	3,122	2,601	3,799	2,759	2,166
テストデータの未知 entity 数	348	1,034	1,744	942	2,627	4,011	1,238	3,319	4,963
テストデータの 3 つ組数	994	2,969	4,919	986	2,880	4,603	960	2,708	4,196
補助データの entity 数	2,474	6,791	10,784	8,191	16,193	20,345	9,899	19,218	23,792
補助データの 3 つ組数	4,352	12,376	19,625	15,277	31,770	40,584	18,638	38,285	48,425

表 4: 未知 entity に関する実験の結果 . 単純なベースラインと提案手法の精度を載せている . また太字体と下線の数値は , それぞれ各データセットでの最も良いスコアと次に良いスコアを表している .

method	pooling	Head			Tail			Both		
		1000	3000	5000	1000	3000	5000	1000	3000	5000
baseline	sum	54.6	52.5	52.0	53.7	53.0	52.8	54.0	52.7	53.2
	max	58.1	56.3	56.4	55.2	54.2	55.3	56.8	56.8	56.4
	avg	63.0	60.2	61.1	63.8	<u>63.9</u>	<u>63.0</u>	65.3	<u>63.9</u>	<u>64.8</u>
proposed	sum	70.2	62.6	59.6	64.6	56.5	55.0	59.5	55.2	54.2
	max	<u>80.3</u>	<u>75.4</u>	<u>72.7</u>	<u>74.8</u>	63.1	58.7	<u>68.0</u>	59.5	56.5
	avg	87.3	84.3	83.3	84.0	75.2	69.2	83.0	73.3	68.2

(1) 未知 entity の選択

未知 entity として扱う entity を選ぶにあたって , 我々は WordNet11 のテストファイルから $N = 1,000, 3,000, 5,000$ 個の 3 つ組を選びだした . また選ばれたそれぞれの 3 つの集合に対し , 我々は更に 3 つの違う設定で {Head, Tail, Both} のデータセットを作成した .

Head のデータセットは , 選ばれた 3 つ組の head entity を未知 entity の候補として扱うデータセットである . Tail のデータセットも Head のデータセットに似ており , tail entity が未知 entity として扱う . Both のデータセットでは , 選ばれた 3 つ組の head , tail entity の両方を未知 entity の候補として扱う . それぞれの設定で選ばれた未知 entity の候補の集合は C として表記される . 最終的に未知 entity として用いられる entity は WordNet11 の訓練セットに (e, r, e') か (e', r, e) として含まれているものである . ただし $e \in C$ は未知 entity の候補であり , $e' \notin C$ は既知の entity とする .

(2) データの分割

選ばれた未知 entity に基づいて , オリジナルの訓練データは訓練データ , 補助データ , 利用しないデータに分割される . 訓練データは未知 entity を含まない 3 つ組であり通常の知識ベースの役割を果たす . 補助データは 1 つの未知 entity と 1 つの既知 entity を含む 3 つ組から構成され , このデータが既知の知識ベースから未知 entity への知識の転移を手助けする . 最後に利用しないデータは未知 entity のみから構成される 3 つ組で

あり , 今回の実験では排除される .

検証データには , 単純に WordNet11 の検証データから未知 entity を含むものを排除したものをを用いた . テストデータの 3 つ組には我々は step-1 で選ばれた 3 つ組を用いる . ただし未知 entity を含まないものは排除した .

作成された未知 entity のデータセットの計量的な性質は表 3 に記載されている . 我々は生成されたそれぞれのデータセットを , {Head, Tail, Both}-{1000, 3000, 5000} , とそれぞれ表記した . ただし最初の箇所は未知 entity が 3 つ組のどの位置なのかを表し , 次の箇所はどれだけ未知 entity として用いられたかを表す .

4.2.2 結果

上記の手続きによって制作された 9 つのデータセットを用い , 我々は提案手法の有効性を確認した .

我々は次のような単純な手法を本実験のベースラインとした . 与えられた未知 entity u に対し , 我々はまず TransE で学習された近傍の埋め込みベクトルを獲得し , そしてそれらのベクトルに sum や max , 平均といった操作を適用することで未知 entity u の表現ベクトルを獲得する . 未知 entity u の近傍は全て既知の知識ベースに含まれるため , それら近傍のベクトルは全て通常の知識ベース補完の手法で学習することが可能である . TransE の訓練やハイパーパラメータは原著 [2] の設定を用いた .

実験結果は表 4 に記載されている . タイトルが 'pooling' である列はどの pooling 関数が用いられたかを示す . 表が

示すように、我々のモデルはベースラインより良い精度を示している。特に、平均 pooling を用いた Graph-NN は全てのデータセットで他のモデルより良い精度を示している。max pooling 関数を用いた Graph-NN も良い精度を示しているが、しかし例えば Tail-3000 などの幾つかの設定において、平均 pooling を用いたベースラインの方が良い精度を示している。

5. 結論

本論文では我々は訓練時に観測されない entity を扱う新しい知識ベース補完のタスクを提案した。またそのような未知の entity を扱うため、Graph neural network を用いた知識ベース補完の手法を提案した。提案手法は未知 entity 問題においてベースラインより優れた精度を示した。また標準的な知識ベース補完の設定においても、WordNet11 のデータセットの場合に先行研究より良い精度を示した。

参考文献

- [1] Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J.: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge, *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pp. 1247–1250 (online), DOI: 10.1145/1376616.1376746 (2008).
- [2] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. and Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data, *Advances in Neural Information Processing Systems 26* (Burgess, C. J. C., Bottou, L., Welling, M., Ghahramani, Z. and Weinberger, K. Q., eds.), pp. 2787–2795 (online), available from <http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf> (2013).
- [3] Cao, S., Lu, W. and Xu, Q.: Deep Neural Networks for Learning Graph Representations (2016).
- [4] Defferrard, M., Bresson, X. and Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, *Advances in Neural Information Processing Systems 29* (Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I. and Garnett, R., eds.), pp. 3844–3852 (online), available from <http://papers.nips.cc/paper/6081-convolutional-neural-networks-on-graphs-with-fast-localized-spectral-filtering.pdf> (2016).
- [5] Guu, K., Miller, J. and Liang, P.: Traversing Knowledge Graphs in Vector Space, *Empirical Methods in Natural Language Processing (EMNLP)* (2015).
- [6] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (online), DOI: 10.1109/CVPR.2016.90 (2016).
- [7] Hoffart, J., Altun, Y. and Weikum, G.: Discovering Emerging Entities with Ambiguous Names, *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pp. 385–396 (online), DOI: 10.1145/2566486.2568003 (2014).
- [8] Hoffart, J., Milchevski, D., Weikum, G., Anand, A. and Singh, J.: The Knowledge Awakens: Keeping Knowledge Bases Fresh with Emerging Entities, *Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion*, pp. 203–206 (online), DOI: 10.1145/2872518.2890537 (2016).
- [9] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456 (online), available from <http://jmlr.org/proceedings/papers/v37/loffe15.html> (2015).
- [10] Ji, G., He, S., Xu, L., Liu, K. and Zhao, J.: Knowledge Graph Embedding via Dynamic Mapping Matrix, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 687–696 (online), available from <http://www.aclweb.org/anthology/P15-1067> (2015).
- [11] Ji, G., Liu, K., He, S. and Zhao, J.: Knowledge Graph Completion with Adaptive Sparse Transfer Matrix (2016).
- [12] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *CoRR*, Vol. abs/1412.6980 (online), available from <http://arxiv.org/abs/1412.6980> (2014).
- [13] Li, Y., Tarlow, D., Brockschmidt, M. and Zemel, R. S.: Gated Graph Sequence Neural Networks, *CoRR*, Vol. abs/1511.05493 (online), available from <http://arxiv.org/abs/1511.05493> (2015).
- [14] Lin, Y., Liu, Z., Sun, M., Liu, Y. and Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion (2015).
- [15] Miller, G. A.: WordNet: A Lexical Database for English, *Commun. ACM*, Vol. 38, No. 11, pp. 39–41 (online), DOI: 10.1145/219717.219748 (1995).
- [16] Nguyen, D. Q., Sirts, K., Qu, L. and Johnson, M.: Neighborhood Mixture Model for Knowledge Base Completion, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 40–50 (online), available from <http://aclweb.org/anthology/K/K16/K16-1005.pdf> (2016).
- [17] Nickel, M., Murphy, K., Tresp, V. and Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graphs, *Proceedings of the IEEE*, Vol. 104, No. 1, pp. 11–33 (online), DOI: 10.1109/JPROC.2015.2483592 (2016).
- [18] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. and Monfardini, G.: The Graph Neural Network Model, *Trans. Neur. Netw.*, Vol. 20, No. 1, pp. 61–80 (online), DOI: 10.1109/TNN.2008.2005605 (2009).
- [19] Socher, R., Chen, D., Manning, C. D. and Ng, A.: Reasoning With Neural Tensor Networks for Knowledge Base Completion, *Advances in Neural Information Processing Systems 26* (Burgess, C. J. C., Bottou, L., Welling, M., Ghahramani, Z. and Weinberger, K. Q., eds.), pp. 926–934 (online), available from <http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf> (2013).
- [20] Wang, Z., Zhang, J., Feng, J. and Chen, Z.: Knowledge Graph Embedding by Translating on Hyperplanes (2014).
- [21] Xiao, H., Huang, M. and Zhu, X.: From One Point to a Manifold: Knowledge Graph Embedding for Precise Link Prediction, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1315–1321 (online), available from <http://www.ijcai.org/Abstract/16/190> (2016).
- [22] Xiao, H., Huang, M. and Zhu, X.: TransG : A Generative Model for Knowledge Graph Embedding, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2316–2325 (online), available from <http://www.aclweb.org/anthology/P16-1219> (2016).
- [23] Yoon, H.-G., Song, H.-J., Park, S.-B. and Park, S.-Y.:

A Translation-Based Knowledge Graph Embedding Preserving Logical Property of Relations, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 907–916 (online), available from <http://www.aclweb.org/anthology/N16-1105> (2016).