1A3-3

DNN による RDF 上の単語間の関係の予測

Prediction of relations among RDF entities by DNN

大貫陽平 *1 Yohei Onuki　貫井駿 *1 Shun Nukui　村田剛志 *1 Tsuyoshi Murata　稲木誓哉 *2 Seiya Inagi　邱シュウレ *2 Xule Qiu　渡部雅夫 *2 Masao Watanabe
岡本洋 *2 Hiroshi Okamoto

*1東京工業大学 情報理工学院 情報工学系
Department of Computer Science, School of Computing, Tokyo Institute of Technology

*2富士ゼロックス (株) 研究技術開発本部
Research & Technology Group, Fuji Xerox Co., Ltd.

Prediction of relation among entities is important for ontology construction. TransE and TransR are the methods for such prediction. In this research, we developed RDFDNN that predicts a predicate from a subject and an object. Experimental results showed that the accuracies of predictions from subjects and objects by RDFDNN are higher than those by TransE and TransR.

1. はじめに

オントロジー学習はセマンティックウェブの実現に必要なオントロジーを構築する上で重要な要素である. 既存のオントロジーに新たな単語を追加する際, すでにオントロジーに含まれている単語との関係を高い精度で予測することができれば既存のオントロジーを自動的に拡張することにより大規模なオントロジーを構築することができる. 現在グーグルの Knowledge Graph などのセマンティックウェブ実現のための大規模データベースが存在しており, 今後大規模なオントロジーを構築する需要が高まることを考えると単語間の関係予測は重要である.

本研究では Resource Description Framework(以下 RDF) 上の単語間の関係を高い精度で予測することを目標とする. RDF はウェブ上の資源を表現するための枠組みである. トリプルは (主語, 述語, 目的語) の 3 つの要素から構成されている. 主語と目的語は単語であり, 述語は単語間の関係を示す. 例えば (日本, 首都, 東京) というトリプルによって”日本は東京という首都を持つ”という情報を表現できる. 本研究ではこのトリプルのうち”日本”と”東京”が入力として与えられた時に”首都”と予測することを目指している. 本研究では主語と目的語を入力として, それらの関係を出力とする Deep Neural Network(以下 DNN) を構築した.

本研究では FreeBase と Wordnet の 2 つのデータセットを用いた. 実験では既存手法との比較, 予測失敗例の分析, 埋め込み次元と予測精度の関係の分析, 埋め込み次元と計算時間の関係の分析を行った. これらの結果, 主語と目的語から述語を予測するタスクにおいて既存手法である TransE[Borders 13] や TransR[Yankai 15] と比較して提案手法が予測精度の面で優れていることがわかった. また,RDFDNN の埋め込み次元の適切な決定方法も判明した.

連絡先: 東京工業大学 情報理工学院 情報工学系
東京都目黒区大岡山 2 丁目 12-1 W8-59
E-mail:oonuki@net.c.titech.ac.jp

2. 提案手法

本研究では DNN を用いた RDF 上の単語間の関係の予測を行う. RDF のトリプル (h, l, t) の h と t を入力, l を出力とすることで h と t から l を予測する DNN を構築する. この DNN を以下では RDFDNN と呼ぶ.

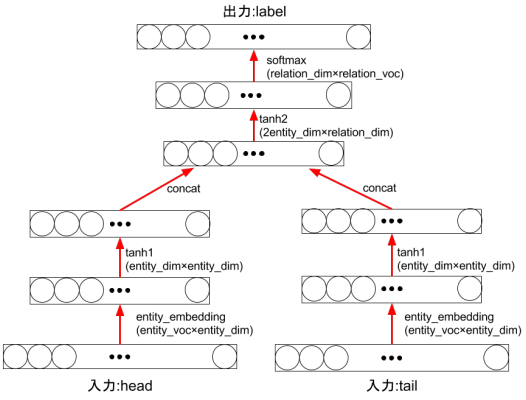

図 1: RDFDNN の構成

RDFDNN の構成を図 1 に示す. 図中の丸はノード, いくつかの丸を束ねた四角は層, 赤い矢印は重み行列による変換を示す. 入力は h と t に対応したワンホットコードであり, 出力は l のワンホットコード表現に対応した確率の分布である. ワンホットコードとは, n 個の単語が存在していて i 番目の単語を表現したい場合, 以下のような i 番目の要素が 1, それ以外の要素を 0 の n 次元のベクトルを用いて単語を表現する方法のことである.

$$\begin{pmatrix} 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix} \quad (1)$$

entity_voc と *relation_voc* は学習対象に含まれる単語数と

関係数である．$entity\_dim$ と $relation\_dim$ は変数であり重み行列の次元を表す．RDF 上の単語をベクトルによる表現に変換するのが埋め込みである．ベクトル表現に変換した際のベクトルの次元を埋め込み次元という．

RDFDNN には $entity\_embedding$, $tanh1$, $concat$, $tanh2$, $softmax$ の 5 種類の重み行列が使われている．$tanh1$ と $tanh2$ は tanh 関数による活性化の重み行列, $softmax$ は softmax 関数を用いた活性化の重み行列である．concat では h と t の埋め込み表現の合成を行うが, ここでは単純に 2 つの埋め込み表現を前後に連結した．$entity\_embedding$ は単語を埋め込み表現に変換するための重み行列である．また $entity\_embedding$, $tanh1$, $tanh2$, $softmax$ の重み行列に対して L2 正則化を行うことで過学習を抑制した．

RDFDNN の出力が l のワンホットコード表現に対応した確率の分布であるので, RDFDNN の学習の目的関数には以下のクロスエントロピーを用いた．S は学習に用いたデータセットに含まれるトリプルの集合, $P(h,t)$ は h と t を入力として RDFDNN により計算を行ったときの出力, k は $0 \leq k < relation\_voc$ を満たす整数のインデックスである．目標関数の最適化を行うオプティマイザには Adam[Kingma 15] を用いた．

$$E = -\sum_{(h,t,l)\in S}\sum_{k\in relation\_voc} l_k log P(h,t)_k \qquad (2)$$

3. 既存手法

3.1 TransE

TransE[Borders 13] は単語と関係の双方を同一のベクトル空間に埋め込むことで単語, 関係を示すベクトル同士の演算を可能にしている．この演算により h と l から t を予測したり, h と t から l を予測したりすることが可能である．TransE は以下の式が成立するように空間を生成している．

$$d(h+l,t) = 0 \qquad (3)$$

d は 2 つのベクトル間のユークリッド距離を計算する関数である．

3.2 TransR

TransR[Yankai 15] は TransE を発展させた手法であり, TransE の弱点であった 1 対 N 対応の関係の学習を行っている．1 対 N 対応の関係とは (太郎, 好物, 寿司)(太郎, 好物, ピザ) のようなある h と l の組に対して t が複数存在しうるような l のことである．TransE の場合 $d(h+l,t) = 0$ が成立するようにベクトル表現が学習されるため, 上記の例では寿司とピザが同一のベクトル表現となってしまい問題がある．TransR では h と t を l に固有の変換行列 M_l で写像変換してから演算をすることでこの問題を回避している．

TransR では以下の式が成立するように空間が生成される．M_l は関係 l に対応した変換行列である．

$$d(hM_l + l, tM_l) = 0 \qquad (4)$$

4. データセット

本実験では以下の 2 つのデータセットの一部を抜粋したものである FB15k と WN18 を用いて実験を行った．FB15k と WN18 の詳細を表 1 に示す． これらは先行研究 [Borders 13][Yankai 15] で用いられたものと同じものである．FB15k と WN18 のもととなっているデータセットは以下のとおりである．

Freebase[Bollacker 08]
Wikipedia などをもとに作成された百科事典データベース

Wordnet[Miller 95]
英単語の辞書データベース

表 1: 本実験で用いたデータセット FB15k と WN18 の詳細

	FB15k	WN18
元データ	Freebase	Wordnet
単語数 (entity_voc)	14,951	40,943
関係数 (relation_voc)	1,345	18
訓練トリプル数	483,142	141,442
テストトリプル数	59,071	5,000

5. 実験手法

5.1 評価基準

本実験では keras を用いて TensorFlow 上に RDFDNN を実装した．keras での DNN の学習は CuDNN により GPU を用いて行った．keras(https://keras.io) は python で書かれた TensorFlow または Theano 上で実行可能なニューラルネットワークのライブラリである．使用した言語は python, ライブラリは keras と tensorflow である．使用した PC の CPU は Intel Xeon CPU E5-2609, GPU は GeForce GTX 1080 である．

実験の評価は *top-k accuracy* による．訓練データで 10 サイクルの訓練を行った後に試験データに含まれるトリプルの h と t を入力, 回答を l として *top-k accuracy* で評価した．*top-k accuracy* とは与えられた入力を元にもっともらしい回答を k 個出力し, その出力の中に正しい出力が含まれていれば 1 に, そうでなければ 0 になる評価方法である．これを全テストデータに対して行い平均を求めた．今回の実験では DNN の出力である関連性の確率分布を元にもっともらしい答えを k 個選び出した．

既存手法との比較では,FB15k で $(entity\_dim, relation\_dim) = (30, 30)$ として既存手法との比較を行った．同様に WN18 でも $(entity\_dim, relation\_dim) = (30, 30)$ として既存手法との比較を行った．予測失敗例の分析では FB15k で $(entity\_dim, relation\_dim) = (30, 30)$ として RDFDNN の予測失敗の例について分析した．埋め込み次元と予測精度の関係では FB15k で RDFDNN の $entity\_dim$ と $relation\_dim$ の両方を 2,4,6,8,10 のそれぞれに変化させて *top-k accuracy* の変化を観察した．埋め込み次元と計算時間の関係では FB15k で RDFDNN の $entity\_dim$ を 60,120,180,240, $relation\_dim$ を 20,40,60,80 と変化させて計算時間の変化を観察した．

5.2 比較対象

本実験では既存研究 [Yankai 15] の実験のために実装された TransE および TransR との比較を行った．TransE では学習率を 0.01, γ を 1 として FB15k と WN18 の両方で実験を行った．埋め込み次元はそれぞれ 50,100 とした．TransR では学習率を 0.001, γ を 1 として FB15k と WN18 の両方で実験を行った．埋め込み次元はそれぞれ 50,100 とした．

6. 既存手法との比較

ここでは,FB15k で $(entity\_dim, relation\_dim) = (30, 30)$ として既存手法との比較を行った． 同様に WN18 でも

$(entity\_dim, relation\_dim) = (30, 30)$ として既存手法との比較を行った.

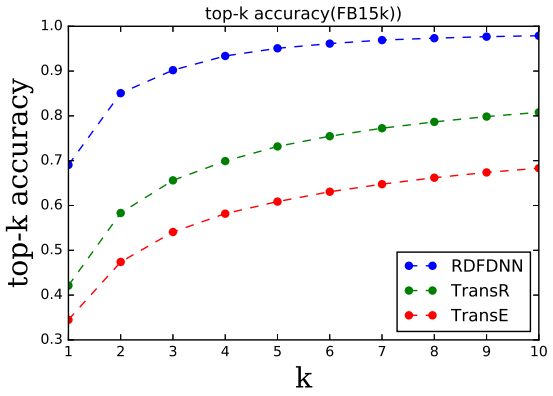

図 2: FB15k での精度比較

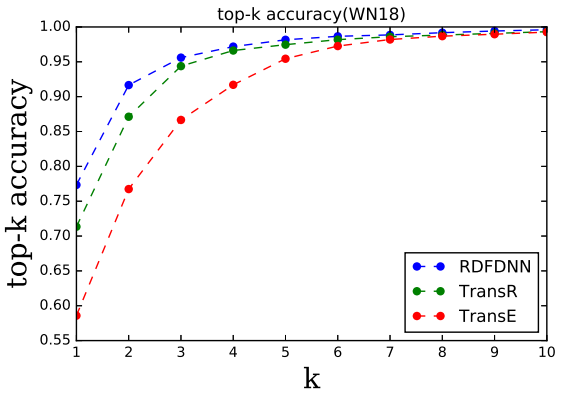

図 3: WN18 での精度比較

図 2 と図 3 は FB15k と WN18 の双方で既存手法との性能比較をした結果である. 横軸は k, 縦軸は *top-k accuracy* である. RDFDNN は最も性能の良かった $(entity\_dim, relation\_dim) = (30, 30)$ の設定とした.

いずれのデータセットにおいても RDFDNN の予測精度が TransE,TransR の予測精度より優れていた. 既存手法と RDFDNN の精度の差が特に大きく出たのは FB15k での実験であった. このため本論文では以後 FB15k での考察を中心とする. こうした結果となった理由として考えられるのは FB15k の関係数が WN18 のそれに比べてかなり大きいことである. FB15k の関係数が 1345 であるのに対して WN18 の関係数は 18 である. 関係の種類が多いために予測の難易度が高くなり, 予測精度の差が大きく反映されたと考えられる. 単語数では WN18 のほうが FB15k に比べて 2 倍ほど多いため, 関係の予測にあたっては単語数よりも関係数のほうが難易度に大きく影響すると考えられる.

7. 予測失敗例の分析

ここでは,FB15k で $(entity\_dim, relation\_dim) = (30, 30)$ として RDFDNN の予測失敗の例について分析した. RDFDNN による単語間の関係の予測失敗は以下の 4 つの種類に分類できる.

- A:出現頻度の高い関係にひきずられての間違い
- B:正解だが期待と違う
- C:構造が似ている
- D:全く無関係

表 2: RDFDNN の予測の間違いの種類別出現回数

間違いのタイプ	出現回数
A	49
B	14
C	5
D	32

$(entity\_dim, relation\_dim) = (30, 30)$ とした場合の RDFDNN において, 予測失敗を以上の 4 種類に分けてカウントを行ったところ表 2 のようになった. 訓練データの中から RDFDNN が予測に失敗したトリプルを無作為に 100 個サンプリングし人手でカウントを行った.

表 2 のとおり最も多かった予測失敗はタイプ A である. A の例としては,"Leslie Dilley"と"レイダース　失われたアーク"の関係予測において, 正しくは"アートディレクター"であるところを"出演者"と予測した例が挙げられる. この例の場合, 人間と映画の間に張られる関係の中で"出演俳優"という関係が最も多いために"出演俳優"と間違った予測をしたと考えられる. このように RDFDNN の予測失敗 100 個のうち A,B,C の 68 個の間違いには何らかの妥当性があった.

8. 埋め込み次元と予測精度の関係

ここでは FB15k で RDFDNN の *entity_dim* と *relation_dim* の両方を 2,4,6,8,10 のそれぞれに変化させて *top-k accuracy* の変化を観察した.

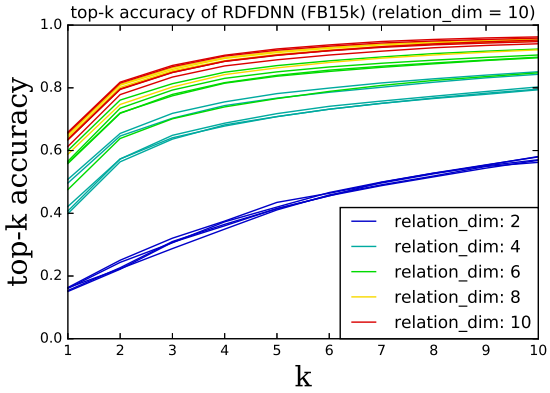

図 4: FB15k での RDFDNN の top-k accuracy (relation_dim=10)

図 4 は *entity_dim* と *relation_dim* の両方を 2,4,6,8,10 のそれぞれに変化させた場合の結果である. 横軸は k, 縦軸は *top-k accuracy* である. *relation_dim* が同じデータは同じ色でプロットされているため, 各色につき 5 本の折れ線がプロットされている.

entity_dim の操作によって生じた *top-k accuracy* の差の最大値は 0.1 程度であり, *relation_dim* によってもたらされる精

度変化に比べて極めて小さい. 図 4 では *relation_dim* が等しいデータ群 (同じ色でプロットされた実験結果) はおおよそまとまってプロットされている. これらのことから *relation_dim* のほうが *entity_dim* よりも RDFDNN の性能に与える影響が大きいことがわかる.

9. 埋め込み次元と計算時間の関係

ここでは FB15k で RDFDNN の *entity_dim* を 60,120, 180,240, *relation_dim* を 20,40,60,80 と変化させて計算時間の変化を観察した.

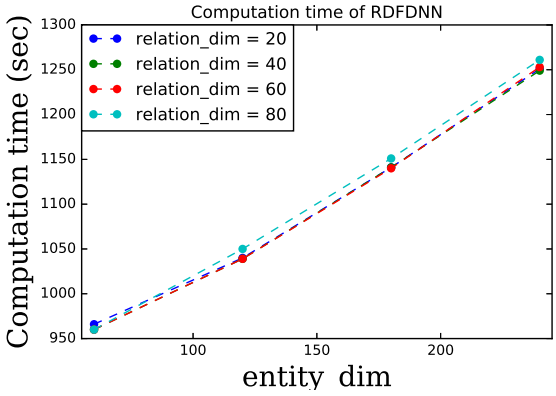

図 5: FB15k での埋め込み次元の変化に伴う RDFDNN の計算時間の変化

図 5 は RDFDNN の埋め込み次元の変化に伴う計算時間の変化を示したものである. 横軸は *entity_dim*, 縦軸は計算時間 (秒) である. グラフからわかるとおり *relation_dim* は RDFDNN の計算時間の変化に寄与しておらず, *entity_dim* のみが計算時間の変化に関連している. これは表 1 からわかるように $entity\_voc >> relation\_voc$ であることから *entity_dim* の値が RDFDNN の計算時間に大きく影響したと考えられる.

10. 考察

RDFDNN の単語間の関係予測の精度は *relation_dim* が大きいほど上昇する. また,RDFDNN の計算時間は *entity_dim* が大きくなるほど長くなる. これらのことから RDFDNN に何らかの RDF の学習を行わせたい場合,

1. 既存手法の場合と同程度の埋め込み次元の値を *entity_dim* と *relation_dim* の初期値として学習を実施
2. $entity\_dim = relation\_dim$ を保ったままで精度が低下しない範囲でこれらの値を 2 分の 1 にしていく
3. *relation_dim* を固定して精度が低下しない範囲で *entity_dim* を 2 分の 1 にしていく

とすることで可能な限り小さな埋め込み次元で高い精度を実現するような RDFDNN を構成できる. 今回の手法で埋め込み次元を減らす際に 2 分の 1 ずつ小さくしていくのは, RDFDNN では性能が低下する埋め込み次元の範囲が広いため, 埋め込み次元削減の幅を広く取ることで可能な限り少ない試行回数で望ましい設定を見つけるためである.

11. おわりに

本研究では既存手法と比較してより精度のよい単語間の関係予測の手法を提案することができた. 本研究の目標は RDFDNN による単語間の関係予測の特性を明らかにすることであった. RDFDNN による単語間の関係予測の特性に関しては以下のようなことがわかった.

- 予測の精度を高くしたい場合,*relation_dim* の値を大きくするべきである
- 計算時間を短くしたい場合,*entity_dim* の値を小さくするべきである
- 予測を失敗した場合でも, 半数以上のケースで何らかの妥当性がある

今後の課題としては RDFDNN では h と l から t を予測することが困難なことが挙げられ, この点の改良が必要である.

参考文献

[Bollacker 08] Bollacker K., Evans C., Paritosh P, Sturge T, Taylor J., "Freebase: a collaboratively created graph database for structuring human knowledge." In Proceedings of KDD, 12471250. (2008), https://developers.google.com/freebase/data, (2017/1/27 閲覧)

[Borders 13] Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O.: "Translating Embeddings for Modeling Multi-relational Data", Part of Advances in Neural Information Processing Systems 26, p. 2787-2795.(2013)

[Hinton 06] Hinton, G. E., Osindero, S., Teh, Y.: "A fast learning algorithm for deep belief nets." Neural Computation, 18, pp 1527-1554. (2006)

[Kingma 15] Kingma D., Ba J.: "Adam: A Method for Stochastic Optimization", International Conference for Learning Representations, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego(2015)

[Lassila 99] Lassila O., Swick R.: "Resource Description Framework(RDF) Model and Syntax Specification", https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ (2017/1/26 閲覧)(1999)

[Miller 95] Miller G.: "WordNet: A Lexical Database for English.", Communications of the ACM Vol. 38, No. 11: 39-41. (1995)

[Yankai 15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu: "Learning Entity and Relation Embeddings for Knowledge Graph Completion", Twenty-Ninth AAAI Conference on Artificial Intelligence,p. 2181-2187. (2015)