The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

対話システムにおける知識獲得質問のための ラベル文字列を用いた知識グラフ補完性能の向上

Improvement of Knowledge Graph Completion Using Label Characters for Questions to Acquire Knowledge in Dialog Systems

藤岡 勇真 *1 Yuma Fujioka　林 克彦 *1 Katsuhiko Hayashi　中野 幹生 *2 Mikio Nakano　駒谷 和範 *1 Kazunori Komatani

*1大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

*2(株) ホンダ・リサーチ・インスティチュート・ジャパン
Honda Research Institute Japan Co., Ltd.

Dialogue systems cannot respond about information that is not explicitly described in their knowledge bases. Constructing a perfect knowledge base is practically impossible; that is, filling all the values in databases is quite labor-intensive. We are trying to construct a system that can acquire information that is not explicitly described in the knowledge base by inferring latent information from knowledge graphs. In particular, we complement the links in a knowledge graph by using an embedding into latent space. We use partial character sequences of labels (i.e. entity names) to improve of knowledge graph completion. We also show examples of queries generated using the latent information in our target knowledge graph.

1. はじめに

データベースを参照して応答を行うタイプの対話システムは，データベースに明示的に記述されていない情報について上手く応答できない．想定する応答に必要な情報が網羅したデータベースが構築できれば良いが，人手で構築することは不可能な場合が多い．

そこで我々は，データベースにない情報を自ら獲得出来る対話システムを構築することでこの問題の解決を試みている．そのようなシステムはユーザに質問して情報を聞き出そうとするが，この時に明らかに間違っている質問をしてしまうとユーザの対話意欲を削いでしまうという課題がある．

我々は，グラフ構造を持つデータベースである知識グラフにある潜在的な情報を利用して，できるだけ間違った内容の質問を避けることができる対話システムの構築を目指している [藤岡 18]．知識グラフはデータ間の様々な関係を表現するのに長けた知識モデルであり [Angles 08]，近年様々なシステムやサービスで用いられている．. 知識グラフに対する代表的な解析手法として，潜在空間への埋め込みがある [Kadlec 17]．埋め込み表現を用いることで知識グラフのリンクの補完が行えるが，この補完されたリンクの確信度に基づきユーザへの質問を生成する．

本稿では，知識獲得で利用する知識グラフ補完の精度を向上させるため，ラベルの文字列を用いた知識グラフ埋め込み手法を提案する．ここでラベルとは，知識グラフ上のエンティティが持つ，自身を識別する名称である．このラベルの文字列を部分文字列に分解し利用することで，類似したラベルを持つエンティティ間に関連性を持たせ，補完精度の改善を図る．この手法の効果を，実際の対話システムでの利用を目的として作成された知識グラフを用いて検証し，どのような質問が行えるかを例示する．

連絡先: 藤岡 勇真，大阪大学 産業科学研究所，
大阪府茨木市美穂ヶ丘 8-1，Tel:06-6879-8416，
E-mail:fujioka@ei.sanken.osaka-u.ac.jp

2. 知識獲得を行う対話システムの枠組み

2.1 知識グラフ埋め込みを利用した知識獲得

本研究では，対話システムが知識グラフをバックエンドデータベースとして持っていることを仮定する．一般に知識グラフはラベル付き有向グラフとして表される．有向グラフ上のエッジにはリレーションを表すラベルが付与されており，ノードはエンティティに相当する．知識グラフ上のエンティティの集合を $\mathcal{E}$，リレーションの集合を $\mathcal{R}$ とする．$e_i, e_k \in \mathcal{E}$, $w_j \in \mathcal{R}$ に対して三つ組 (i, j, k) をトリプルと呼ぶ．この時 i, j, k をそれぞれ主語，述語，目的語と呼ぶ．トリプルは 2 つのエンティティ間の関係を表現する知識グラフの基本的な要素である．知識グラフ $\mathcal{G}$ はトリプルを要素とする集合として表せる．

図 1 にユーザから知識獲得を行う対話システムの枠組みを示す．知識グラフ上にない知識，すなわちトリプルをユーザから獲得するために質問リストを生成する（図 1①）．知識グラフ上にないトリプルの存在をユーザに質問することで知識獲得を試みる．図の例では (もみじ丼，料理種，?) と目的語を穴埋めするように質問リストを生成している．本稿で生成する質問リストはこの例のように，目的語を穴埋めするような形式で生成されるものとする．この時，対話の文脈や話題に関連する質問リストを生成するようにし，急な話題転換を避けるようにする．そして既知の知識グラフから埋め込み表現を学習し（図 1②），埋め込み表現から質問リスト内のトリプルに対し存在尤度を表すスコアを計算し確信度として付与する（図 1③）．付与した確信度に基づき質問リストをソートし，最も高い確信度かつその確信度がしきい値以上のトリプルの有無をユーザに質問し知識獲得を試みる（図 1④）．確信度による順位だけでなく，確信度の絶対値もしきい値によって考慮する．これは，確信度が低い，すなわち推定結果に自信がない場合にもユーザに問いかけてしまうことを防ぎ，明らかに間違っているような質問でユーザの対話意欲を削がないようにするのが狙いである．システムの設計指針は，質問する内容をなるべく正しいものとすることである．知識グラフ埋め込みによる補完精度の向上は質問内容の正しさに直結するため，ラベル文字列を用いた知識グラフ埋め込みによる補完精度向上を 3 章で提案する．

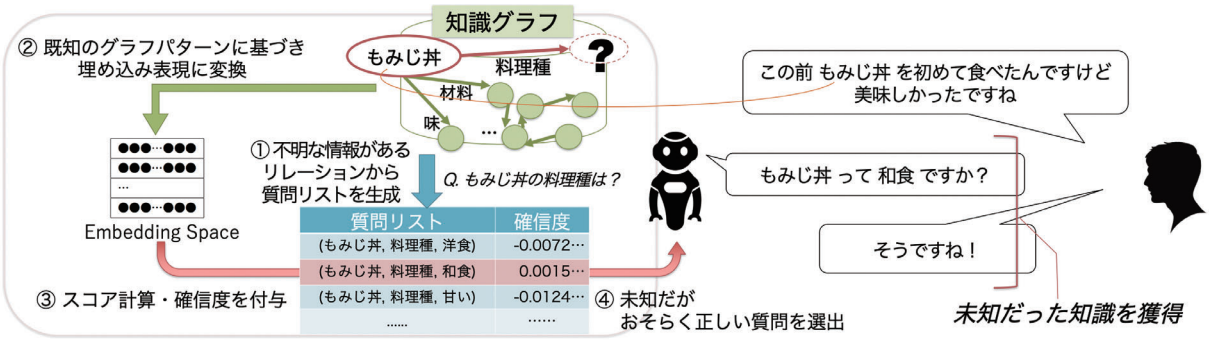

図 1: 知識獲得を行う対話システムの枠組み

2.2 知識グラフ埋め込みと ComplEx

2 章で述べた枠組みでは，トリプルの存在尤度を計算する必要がある．その計算に知識グラフの埋め込み表現を利用する．知識グラフ埋め込みは，知識グラフに対する代表的な解析手法として知られている．低次元線形空間に知識グラフを埋め込み汎化させることで，グラフ上の欠損したリンクの有無を推論し補完する．知識グラフは，$|\mathcal{E}| \times |\mathcal{R}| \times |\mathcal{E}|$ の 3 階テンソル $\mathcal{X}$ として表現することができ，$\mathcal{X}$ の (i,j,k) 要素 $x_{i,j,k}$ は以下のように表される．

$$x_{i,j,k} = \begin{cases} 1, & (i,j,k) \in \mathcal{G}; \\ -1, & otherwise. \end{cases}$$

この表現を用いて，知識グラフ埋め込みではトリプル (i,j,k) が知識グラフ上に存在する確率 $P(x_{i,j,k}=1)$ を，モデルに対応したスコア関数 ϕ を用いて以下のように表す．

$$P(x_{i,j,k}=1) = \sigma(\phi(i,j,k;\Theta))$$

$\sigma(\cdot)$ はシグモイド関数，Θ は各モデルにおけるパラメータを表す．

ComplEx[Trouillon 16] は知識グラフ埋め込みモデルの一種である．ComplEx は複素数表現とエルミート内積を利用したモデルとして知られる．ComplEx におけるスコア関数 ϕ_{Comp} は，$\mathbb{C}^n$ を複素 n-次元空間，$\boldsymbol{e}_i, \boldsymbol{e}_k \in \mathbb{C}^D$，$\boldsymbol{w}_j \in \mathbb{C}^D$ をそれぞれエンティティ・リレーションに関する D 次元の埋め込みベクトルとし，$Re(x)$ を x の実部とすると，

$$\phi_{Comp}(i,\, j,\, k;\Theta) = Re(< \boldsymbol{w}_j, \boldsymbol{e}_i, \overline{\boldsymbol{e}}_k >)$$

と書ける．ここでベクトル $\boldsymbol{v}$ の l 番目の要素を v_l とした時，$< \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} >:= \sum_k a_k b_k c_k$ と定義する．$\overline{\boldsymbol{v}}$ は $\boldsymbol{v}$ の複素共役ベクトルである．

3. ラベル文字列を用いた知識グラフ埋め込み

本章では，エンティティの持つラベル文字列を知識グラフの埋め込み学習に用いる手法を提案する．2.2 節で述べたモデルはエンティティ同士がどのようにリンクしているかを元に埋め込み表現を学習するが，すべてのエンティティが持つラベル文字列に関する情報は学習に組み込まれていない．そこで，知識グラフ上のエンティティのラベル文字列を部分文字列に分解し，先頭もしくは末尾の数文字を擬似エンティティとして作成することでラベル文字列を組み込む．似た部分文字列を持つエンティティ同士は似た性質を持つ可能性が高いという仮定のもとで，このようなエンティティ同士の関連性が強くなり補完精度の向上に結びつくことを期待する．

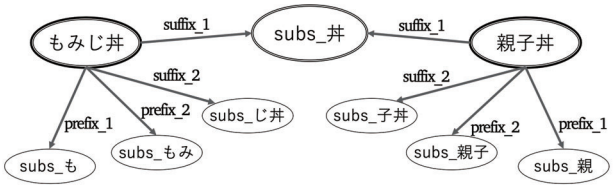

図 2: 「もみじ丼」と「親子丼」に対する $N=2$ での展開例

N 文字以下の先頭もしくは末尾の部分文字列で擬似的にエンティティを作成し新たにトリプルを構成することを，N 文字で展開すると記述する．図 2 に $N=2$，すなわち 2 文字で展開を行った例を示す．展開の対象となっている「もみじ丼」と「親子丼」の先頭・末尾 1，2 文字が抽出され，部分文字列であることを示す subs_ を先頭に付けたラベルを持つエンティティが作成される．そして展開元のエンティティから作成した疑似エンティティに向けて prefix_N もしくは suffix_N というリレーションでトリプルを構成する．図 2 の例では，2 つの料理が「subs_丼」という疑似エンティティを経由して繋がっていることがわかる．

4. ラベル文字列を用いた知識グラフ埋め込みによる補完精度

ラベル文字列を用いた知識グラフ埋め込みによる補完の精度を検証し，その効果を調査する．

4.1 使用データ

本実験で使用する知識グラフは，対話システムでの運用を目的として，料理に関する表形式のデータベースを元に作成されたものである．詳述すると，料理やその材料，料理種，味，食べられる場所などが格納されている．このデータベースは人手で作成されたものであり，部分的にしか情報がないため，現在も継続して情報が追加され続けている．このデータベースから作成される知識グラフに対し，2.1 節で述べた知識獲得の枠組みを適用して情報の拡充を行うという設定の下で議論する．

この知識グラフはエンティティ数 $|\mathcal{E}| = 7289$，リレーション数 $|\mathcal{R}| = 14$ であり，そのトリプル数は 22321 である．エンティティに付与されたラベルの長さの平均は 5.88（文字），標準偏差は 3.18（文字）である．

表 1: 各設定における精度指標及び学習データの詳細

設定	Hits@ 1	3	5	10	MRR	学習トリプル数	エンティティ数	リレーション数
Baseline	0.140	0.205	0.272	0.406	0.215	17857	6898	14
$N=1$	0.187	0.306	0.394	0.521	0.286	31681	7865	16
$N=2$	0.237	0.384	0.473	0.580	0.345	45361	12147	18
$N=3$	**0.257**	0.436	0.524	0.621	**0.377**	57893	18534	20
$N=4$	0.246	0.441	0.523	0.626	0.373	68477	25035	22
$N=5$	0.241	0.448	0.532	0.627	0.373	76837	30970	24
$N=6$	0.227	0.452	0.537	0.629	0.368	83271	35791	26
$N=7$	0.221	**0.472**	**0.550**	**0.641**	0.373	87975	39381	28

表 2: リレーション毎の Hits@K．$N=3$ とした場合を示している（括弧内の数値は Baseline での結果との差を表す）．

リレーション名	Hits@ 1		3		5		10		出現数
is_a	**0.157**	(+0.140)	0.284	(+0.216)	0.331	(+0.240)	0.389	(+0.243)	795
材料	0.056	(+0.047)	0.108	(+0.075)	0.158	(+0.107)	0.238	(+0.115)	702
popularity	**0.119**	(+0.108)	0.377	(+0.238)	0.565	(+0.308)	0.736	(+0.169)	639
別名	0.828	(▼ −0.108)	0.980	(+0.010)	0.982	(+0.008)	0.986	(+0.012)	501
meal_type	**0.578**	(+0.516)	0.878	(+0.609)	0.916	(+0.477)	0.950	(+0.235)	417
料理種	**0.130**	(+0.115)	0.367	(+0.315)	0.536	(+0.382)	0.682	(+0.279)	330
温度	**0.297**	(+0.246)	0.572	(+0.381)	0.674	(+0.347)	0.801	(+0.220)	236
restaurant	**0.385**	(+0.261)	0.578	(+0.280)	0.646	(+0.211)	0.714	(+0.180)	161
味	**0.239**	(+0.159)	0.487	(+0.336)	0.522	(+0.301)	0.611	(+0.310)	113
場所	0.000	(±0.000)	0.030	(±0.000)	0.061	(+0.030)	0.091	(+0.030)	33
typical_side	0.056	(+0.056)	0.111	(+0.111)	0.222	(+0.167)	0.278	(+0.222)	18
代表料理	0.000	(±0.000)	0.000	(±0.000)	0.063	(+0.063)	0.063	(+0.063)	16
機会	0.000	(±0.000)	0.067	(+0.067)	0.133	(+0.067)	0.133	(+0.067)	15
時間	0.182	(+0.091)	0.364	(+0.273)	0.455	(+0.273)	0.455	(+0.182)	11

4.2 実験方法

精度検証のため 4.1 節で述べた知識グラフを無作為に 5 分割し，その内 4 つをトレーニングデータ $\mathcal{G}'$，1 つをテストデータ $\mathcal{H}'$ とする 5 分割交差検証を行った．トレーニングデータを埋め込み，その埋め込み表現と $\mathcal{H}'$ から知識グラフ補完で一般に用いられる精度指標である Hits@K と MRR を算出した．Hits@K は $(i',j',k') \in \mathcal{H}'$ の内 k' を全エンティティと入れ替えてスコアを計算し，得られるランキング中で $\phi_{Comp}(i',j',k';\Theta)$ が上位 K 位に入る割合を指す．MRR は平均逆順位とも呼ばれ，前述のランキングにおける $\phi_{Comp}(i',j',k';\Theta)$ の順位の逆数の平均として表される．

ラベル文字列を用いた知識グラフ埋め込みの効果を検証するため，展開文字数 $N=1, 2, ..., 7$ としてトレーニングデータを展開した場合に加え，ベースラインとして展開を行わない場合の Hits@K，MRR を算出した．展開時のトレーニングデータに関する詳細を表 1 の右部に示す．テストデータ数は 4464 個である．全ての場合で共通して ComplEx による埋め込みを適用し，埋め込み次元は複素 200 次元としてロジスティック回帰による学習を行った．学習率調整は [Trouillon 16] に倣い Adagrad を利用し，イテレーション数は 1000，負例サンプリング数は 5 とした．指標算出時のランキングについて，展開して得られた部分文字列エンティティを含むトリプルはランキングの対象外とした．同様に，既にテストデータに含まれるトリプルに関してもランキングの対象外とした．

4.3 知識グラフ埋め込みによる補完精度

表 1 の左部に 5 分割交差検証の結果を示す．展開文字数 N に関わらず，ベースラインに対して全ての指標が上昇していることわかった．Hits@3，5，10 については，N に比例して上昇していた．一方で Hits@1 は，$N=3$ の時に最大値をとり，$N=4, 5, ...$ と増加するにつれ減少する傾向が見られた．MRR に関しても同様に $N=3$ の時が最大であった．

この分割したデータで補完された実例を示す．例えば (たらこ茶漬け， is_a， お茶漬け) といった部分文字列がそのまま結びついた単純な補完例があり，2987 位から 1 位に改善されていた．また (ホットケーキ， 味， 甘い) といったように，部分文字列で結びついたエンティティの情報から改善されたと考えられる例もあり，8 位から 1 位に改善されていた．

4.4 リレーション毎の補完精度

リレーション毎の精度を調査した．ある分割データに対し，4.2 節で述べた方法に従って算出された Hits@K をリレーション毎に分類して算出した結果を表 2 に示す．表 1 の結果（Hits@1）を考慮し，$N=3$ の場合とベースライン設定の 2 設定の比較として示す．Hits@3，5，10 の精度が悪化しているリレーションは無く，全体としてはベースライン以上の精度を保持していることがわかった．『is_a』や『料理種』といったような，階層構造やタイプ等を表しているリレーションは，ベースラインに対し精度が大きく上昇していた．またそれ以外にも,『味』や『温度』といったリレーションに関しても同等の精度上昇が確認できた．

一方で，ベースラインに対して大きな改善がされていないリレーションもあった．出現数が少ないリレーション（100 以下）では精度が改善されていないものがほとんどである．また『別名』というリレーションでは，Hits@3，5，10 はベースラインと同程度の精度であるものの，Hits@1 の値が大幅に減少するという結果になった．本質的にラベル文字列を取り扱うリレーションである『別名』にとって，部分文字列のリンク情報がノイズとなってしまった可能性が考えられる．しかし精度の絶対値自体は他のリレーションに比べ比較的良好である．これはデータの特性に起因したものであり，使用データ上で基本的に別名というリレーションがエンティティ間に双方向に張られる様に取り扱っており，補完が容易になっているためと考えている．出現数も多く重要なリレーションである『材料』の精度も大きな改善はされておらず，精度の絶対値も低いことがわかる．材料のように 1 対多の関係を表すリレーションは，1 対 1 の関係を表すリレーションに比べて，ランキング上位に正しいトリプルが来るよう学習するのが容易ではないためだと考えられる．

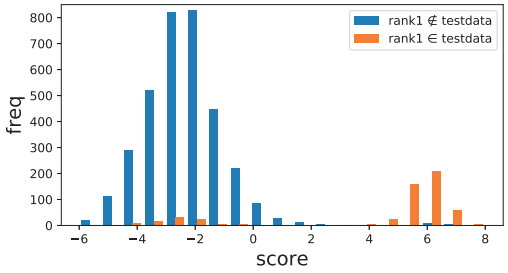

図 3: Baseline 設定でのスコア（確信度）の分布

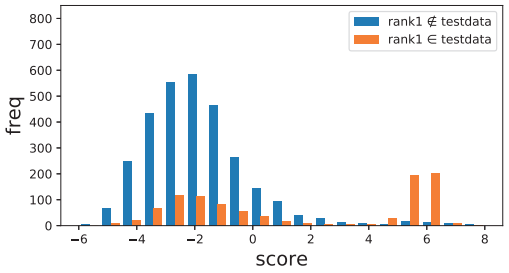

図 4: $N = 3$ としたの時のスコア（確信度）の分布

5. 質問の精度と実例

5.1 スコア分布としきい値

2.1 節で述べた知識獲得を行う対話システムの枠組みにおける，ラベル文字列を用いた知識グラフ埋め込みの効果性を検証する調査を行った．

推測がうまくいく場合，すなわちランキング上で 1 位に順位付けられるトリプルがテストデータ中に存在する場合のスコア（信頼度）の分布を調査する．つまり，テストデータ中に存在するトリプルは正例のみであるため，これが 1 位に順位付けられた場合，補完結果は正解であったとみなせる．4.4 節と同様に，知識グラフのある分割データを用いて，$\mathcal{H}'$ 中のトリプルに関して順位を算出し，1 位のトリプルがテストデータにあるかどうかで分類して各々のスコア分布を確認した．

図 3， 4 に順位が 1 位だったトリプルのスコアで作成したヒストグラムを示す．このヒストグラムは，1 位と判定されたトリプルがテストデータ中に存在するかどうかで振り分けて度数を計測している．図 3 はベースライン設定， 図 4 は $N = 3$ で展開した設定での結果である．テストデータに存在したトリプルが，部分文字列の展開によって増加していることが見てとれる．また，増加したトリプルは -2 付近を中心としたスコアを持っていることがわかる．スコアが 4〜7 付近のトリプルはそのほとんどはリレーションが『別名』である．4.4 節でも述べたが，『別名』はエンティティ間に双方向に張られているリレーションであり，補完が容易なためスコアが比較的高くなったと考えられる．

表 3 に，適合率としきい値の関係を示す．ここでの適合率とは，1 位に順位付けられたトリプルの中でテストデータ $\mathcal{H}'$ 中に存在し，しきい値以上のスコアを付与されたものの割合を表すもので，表の 3，4 列目には割合ではなくその個数を示している．全データ数は 4464 個である．この適合率は，$(i', j', k') \in \mathcal{H}'$ に対応する $(i', j', ?)$ を質問リストとして知識獲得を試みた時に，行う質問が正しい割合として捉えることができる．またしきい値以上かつテストデータ中にあるトリプル数は，2.1 節で述べた知識獲得の枠組みを通して獲得できるトリプルの数と擬似的に捉えられる．適合率が高いほど正確に，表 3 中のトリプル数が多いほどより多くのトリプル（知識）の獲得が期待ができる．同じ適合率の下でそのトリプル数を見ると，適合率が 0.8 の場合は Baseline がやや上回っており，他の適合率設定では $N = 3$ が上回っている．高い適合率の下でこのトリプル数を増加させることに対し，ラベル文字列を用いる大きな効果は見られず，出来るだけ間違った質問をせず多くの知識を獲得するという目標に対しては課題が残る．

表 3: 適合率としきい値の関係．BL は Baseline を表す．

適合率が x 以上	しきい値以上かつ $\mathcal{H}'$ 中にあるトリプル数 N=3	BL	適合率を満たすしきい値 N=3	BL
0.8	453	468	2.034	0.210
0.7	472	469	0.906	−0.333
0.6	497	472	0.160	−0.710
0.5	550	477	−0.592	−1.066
0.4	619	480	−1.280	−1.451

5.2 質問例

提案手法で生成できる質問例について述べる．学習方法は 4.2 節と同様であるが，知識グラフを分割せず，すべてを学習に用いた．ベースラインと $N = 3$ で展開した場合の 2 設定を取り上げる．

例えば（月見団子，味，甘い）という事実について，質問リストを（月見団子，味，？）として順位とそのスコアを算出した結果，ベースライン設定では 5 位，-4.00 で，$N = 3$ とした場合は 1 位，-1.68 であった．「月見団子」と「甘い」が部分文字列の展開によって結びついたわけではないが，”団子”を末尾に持つ他のエンティティの味に関する情報を元に順位とスコアが改善されたものと考えられる．また (五目炊き込みご飯，温度，あたたかい) という事実に関しても同様に順位とスコアを算出すると，ベースライン設定では 7 位，-4.13 で，$N = 3$ とした場合は 1 位，-1.66 であった．この例も，”ご飯”を末尾に持つ料理との関連性が上がったことが改善の理由だと推察出来る．

6. まとめ

本稿では，知識グラフの埋め込み表現を用いた対話システムにおける知識獲得のための質問生成の枠組みについて述べた．またラベル文字列を用いた知識グラフ埋め込みの補完精度の改善を提案し検証した．リレーション毎の補完精度，しきい値設定に関しても言及した．

今後の課題は，ランキング評価におけるルールの記述や対話内容からの質問リスト作成の枠組み設計，能動学習の導入などが挙げられる．

参考文献

[Angles 08] Angles, R. and Gutierrez, C.: Survey of Graph Database Models, *ACM Comput. Surv.*, Vol. 40, No. 1, pp. 1:1–1:39 (2008)

[Kadlec 17] Kadlec, R., Bajgar, O., and Kleindienst, J.: Knowledge Base Completion: Baselines Strike Back, in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 69–74 (2017)

[Trouillon 16] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G.: Complex Embeddings for Simple Link Prediction, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pp. 2071–2080 (2016)

[藤岡 18] 藤岡 勇真, 林 克彦, 中野 幹生, 駒谷 和範：対話システムにおける知識グラフの埋め込み表現を用いた応答生成の試み, *SIG-SLUD*, Vol. B5, No. 02, pp. 88–89 (2018)