

卒業研究報告書

題 目

Masked Language Modeling を用いた
Knowledge Graph 補完手法の提案

研究グループ 第1研究グループ

指導教員 森 直樹 教授

令和5年（2023年）度卒業

(No. 1201201120) 堀本 隆誠

大阪府立大学工学域電気電子系学類情報工学課程

Masked Language Modeling を用いた Knowledge Graph 補完手法の提案

第 1 グループ 堀本 隆誠

1. はじめに

近年, 人工知能技術は急速な発展を遂げている. その中で人間の知識をグラフ構造で表現する Knowledge Graph [1] が注目を集めており, 人工知能の基盤技術としてさまざまな分野で活用されている. しかし, Knowledge Graph 内の知識とそれらの関係を人手ですべて網羅するには多大なコストがかかる. この問題を解決するために Knowledge Graph 内の関係を基に含まれていない関係を自動的に補完する手法が求められている. 本研究では, 言語モデルである BERT の Masked Language Modeling を用いて Knowledge Graph を補完する手法を提案して, その有効性を検証した.

2. 要素技術

2.1. Knowledge Graph

Knowledge Graph [1] とは, さまざまな知識を体系的に連結し, その関係をグラフ構造で表した知識ネットワークのことである. Knowledge Graph は, head, tail を要素に持つ entity 集合と, その entity 間の関係を表現する relation を要素に持つ relation 集合によって構成されており, (head, relation, tail) という 3 つ組構造 (triple) の集合で表すことができる.

2.2. BERT

Bidirectional Encoder Representations from Transformers (BERT) [2] は, Transformer をベースとして双方向エンコーダで構成された言語モデルである. BERT は, マスクされた単語を予測する Masked Language Modeling (MLM) と 2 つの文が連続するかを分類する Next Sentence Prediction (NSP) という 2 つの手法で学習している.

Knowledge Graph BERT (KG-BERT) [3] は, Yao らによって提案された BERT を用いた Knowledge Graph 補完手法の 1 つである. KG-BERT では, triple を表した文 “[CLS] head [SEP] relation [SEP] tail [SEP]” を BERT の入力として, その triple が存在するか否かを “[CLS]” トークンのみを利用した 2 値分類で判定している.

2.3. WN18RR

WN18RR [4] は, 英語の大規模な語彙データベースである WordNet から triple を自動抽出して得られたデータセット WN18 から作られており, WN18 から head と tail を入れ替えて得られる逆関係の triple を除去したデータセットとなっている. WN18RR では, entity の head, もしくは tail が “見出し語, その説明文” の形で表される.

3. 提案手法

Knowledge Graph の triple における head, relation, tail をそれぞれ h, r, t とすると, Knowledge Graph 補完では triple (h, r, ?) に対して ? に入る tail を回答することで entity 間の関係性を予測する. 本研究では, BERT の MLM を fine-tuning したモデルに triple (head, relation, [MASK]) を 1 つのシーケンスとして入力して tail を予測することによる Knowledge Graph 補完手法を提案する.

4. 数値実験

BERT の MLM を fine-tuning したモデルに対して, 2 種類の入力文を用いて実験した. 実験 1 では triple “[CLS] head [SEP] relation [SEP] tail [SEP]” に対して, tail の見出し語を “[MASK]” トークンに置き換えた文 “[CLS] head [SEP] relation [SEP] [MASK], tail の説明文 [SEP]” を入力文とし

表 1: KG-BERT と実験 1, 2 の MRR, Hits@k の値

	MRR	Hits@1	Hits@3	Hits@10
KG-BERT	0.25	12.41	29.44	51.85
実験 1	0.546	52.55	56.16	57.79
実験 2	0.168	10.94	19.37	27.95

て評価した. 実験 2 では実験 1 の入力文に対して, tail の説明文を除去した文 “[CLS] head [SEP] relation [SEP] [MASK] [SEP]” を入力文として評価した.

データを訓練: 検証: テスト = 8 : 1 : 1 に分割し, 評価には Mean Reciprocal Rank (MRR) と Hits@k を使用した. 予測結果の r 番目に正解があるとき, その順位 r のことをランクと呼び, $|T|$ を triple 数, r_i を triple_i における正解 tail のランクとすると, MRR は (1) 式で表される.

$$\text{MRR} = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{1}{r_i} \quad (1)$$

Hits@k は, tail 予測において上位 k 個以内に正解の要素が出力されている割合を表す. 本研究では $k = 1, 3, 10$ で評価した. MRR, Hits@k はともに値が大きいときに推定精度が良いと判断される.

表 1 に KG-BERT および, 実験 1, 2 の MRR と Hits@k の値を示す. 実験 1 ではすべての評価指標において KG-BERT を上回る結果が得られた. 入力に tail の説明文の情報が含まれていることにより tail の見出し語を予測できたと考えられる. 一方, 実験 2 では Hits@1 において KG-BERT と同程度の結果が得られたが, 他の指標においては下回る結果となった. 本実験では tail 予測の出力候補が BERT に登録されている単語すべてであるため, 出力候補を entity に限定することで精度の向上が見込まれる.

5. まとめと今後の課題

本研究では, MLM を用いた 2 種類の入力文に対する tail の見出し語予測について KG-BERT と比較し, 実験 1 において KG-BERT を上回る結果が得られた. これにより MLM を用いた Knowledge Graph 補完手法の有効性が確認できた. 今後の課題として, MLM の出力候補を entity に限定したモデルの作成が挙げられる.

参考文献

- [1] 川村隆浩, 江上周作, 田村光太郎, 外園康智, 鶴飼孝典, 小柳佑介, 西野文人, 岡嶋成司, 村上勝彦, 高松邦彦, 杉浦あおい, 白松俊, 張翔宇, 古崎晃司. 第 1 回ナレッジグラフ推論チャレンジ 2018 開催報告 — 説明性のある人工知能システムを目指して —. 人工知能 34 巻 3 号, 2019.
- [2] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. pp. 4171–4186, June 2019.
- [3] L. Yao, C. Mao, and Y. Luo. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193, 2019.
- [4] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2d knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

目次

1	はじめに	1
2	要素技術	3
2.1	Knowledge Graph	3
2.1.1	概要	3
2.1.2	データ構造	3
2.1.3	グラフ表現	4
2.1.4	Triple	5
2.1.5	埋め込み手法	5
2.1.6	活用例	8
2.2	Transformer	8
2.3	Bidirectional Encoder Representations from Transformers (BERT)	9
2.4	Knowledge Graph BERT (KG-BERT)	11
2.5	WN18RR	14
3	提案手法	17
4	数値実験	20
4.1	データセット	20
4.2	実験 1	20
4.3	実験 2	21
4.4	評価指標	22
4.5	実験結果	23
4.6	考察	25
5	まとめと今後の課題	28
	謝辞	29
	参考文献	30

図目次

2.1	Knowledge Graph の例	4
2.2	Transformer の概略図	10
2.3	BERT の事前学習と転移学習の概略図	11
2.4	KG-BERT	13
3.1	提案手法のモデル概略図	18
3.2	BERT の MLM による fine-tuning の概略図	19
4.1	実験 1 のモデル概略図	22
4.2	実験 2 のモデル概略図	23

表 目 次

2.1	Knowledge Graph の Triple 表現	5
2.2	WN18 から逆関係の Triple を除去する例	15
2.3	WN18RR における Entity の例	15
2.4	WN18RR における Triple の例	16
4.1	データセットをトークン列とした例	21
4.2	WN18RR の内訳	21
4.3	KG-BERT と実験 1, 2 のパラメータ	24
4.4	Filtered Rank の例	24
4.5	KG-BERT と実験 1, 2 の結果	25

1 はじめに

近年, 人工知能技術は急速な発展を遂げている. その中で人間の知識をグラフ構造で表現する Knowledge Graph ^[1] が注目を集めており, 人工知能の基盤技術としてさまざまな分野で活用されている. Knowledge Graph はさまざまな知識とそのつながりをグラフ構造を用いて表現するデータ構造である. 多種多様な情報とそのつながりを体系的に表現できるという利点や, 数値や有限の属性値に限らず自然言語文や音声データといった非構造データを情報として扱えるという利点がある. これらの利点は, 1 つの概念に対する情報を得たいときにそれにつながるさまざまな知識を得られるという点で有用である. また, グラフ構造を体系的にまとめることで効率よく検索ができるという点でも有用である. これらの特性から, Knowledge Graph は Google の Google Knowledge Graph ^[2] などで使用されている.

Knowledge Graph への注目に伴い, 機械学習を用いて Knowledge Graph を表現することの気運が高まっている. 機械学習を利用する利点として, Knowledge Graph の特徴を活かすことで蓄積されたさまざまな種類のデータとそのつながりから従来の単一的な情報では得られなかった新しい知識が得られる点がある. しかし, Knowledge Graph 内の知識とそれらの関係を人手ですべて網羅するには多大なコストがかかる. この問題を解決するために Knowledge Graph 内の関係を基に含まれていない関係を自動的に補完する Knowledge Graph 補完が用いられる. Knowledge Graph 補完は, Knowledge Graph 内の知識を k_1 , k_2 , それらの関係を r としたとき, (k_1, r, k_2) に対して k_1, r の情報から k_2 を回答し, 知識間の関係性を予測する. 従来の Knowledge Graph 補完手法として Knowledge Graph Embedding 手法がある. この手法は知識と関係をそれぞれ実数値ベクトルとして Triple (3 組) で表し, これらのベクトルを用いて Triple の妥当性を評価する. しかし, ほとんどの Knowledge Graph Embedding モデルは Triple の構造情報しか使用しないため, 知識自体の意味情報を効果的に捉えていない.

ELMo ^[3], BERT ^[4], および XLNet ^[5] などの事前学習済み言語モデルは自然言語処理で大きな成功を収めている. これらのモデルは大量の自由なテキストデータで文脈化された単語埋め込みを学習し, 多くの言語処理タスクで最先

端の性能を示している。特に BERT は, Masked Language Modeling (MLM) と Next Sentence Prediction (NSP) を通じて双方向 Transformer エンコーダを事前学習することで豊かな言語情報を扱うことができるという点で注目されている。

本研究では, 知識自体の意味情報を効果的に捉えるために事前学習済みの言語モデルである BERT の MLM を使用して Knowledge Graph を補完する手法を提案する。知識, 関係, および Triple をテキストシーケンスとして BERT の MLM の入力文とすることで, 知識間の関係性を予測する。提案手法について, BERT を用いた Knowledge Graph 補完手法である KG-BERT^[6] との比較実験により有効性を示す。

本論文では, 2 章で関連する要素技術について紹介する。3 章で本研究の提案手法について述べる。4 章で数値実験により提案手法の有効性を確認する。5 章でまとめと今後の課題を示す。

2 要素技術

本章では, 本研究に関連する要素技術についての説明をする.

2.1 Knowledge Graph

本節では, Knowledge Graph というデータ構造について説明する. なお, Knowledge Graph に一意の定義は存在しないため, 本研究に用いた Knowledge Graph の定義についてのみ詳説する.

2.1.1 概要

本研究で扱う Knowledge Graph は, さまざまな情報やデータから抽出した知識をグラフ構造上に整理した知識ベースの一種である. Google により実世界のオブジェクトの検索を可能にするものとして紹介され, さまざまな研究や産業分野に急速に普及された [2].

図 2.1 に Knowledge Graph の例を示す. Knowledge Graph は, それぞれ意味をもつ Entity (実体) 集合と, その Entity 同士の関係性を表現する Relation (関係) 集合によって構成され, それぞれが図 2.1 のグラフにおけるノードとエッジの役目を担っている.

2.1.2 データ構造

Knowledge Graph はその要素として Entity 集合と Relation 集合をもつ. Entity 集合は, 人物, 場所, 単語, 文といった, 何らかの属性をもつ “Entity” の集合である. 各 Entity がもつ属性の種類は, 有限の属性だけでなく, 実数値や自然言語文などの非有限のものもある. なお, 全く同じ属性をもつ異なる Entity は存在しない.

続いて, Relation 集合は 2 つの Entity に対して, 一方の Entity から見た Entity との関係性を規定する “Relation” の集合である. Relation によって個々の Entity に繋がりを与えて, それにより全体として有用な意味をもつネットワークができる. 各 Relation は Entity と同様に属性をもち, それにより

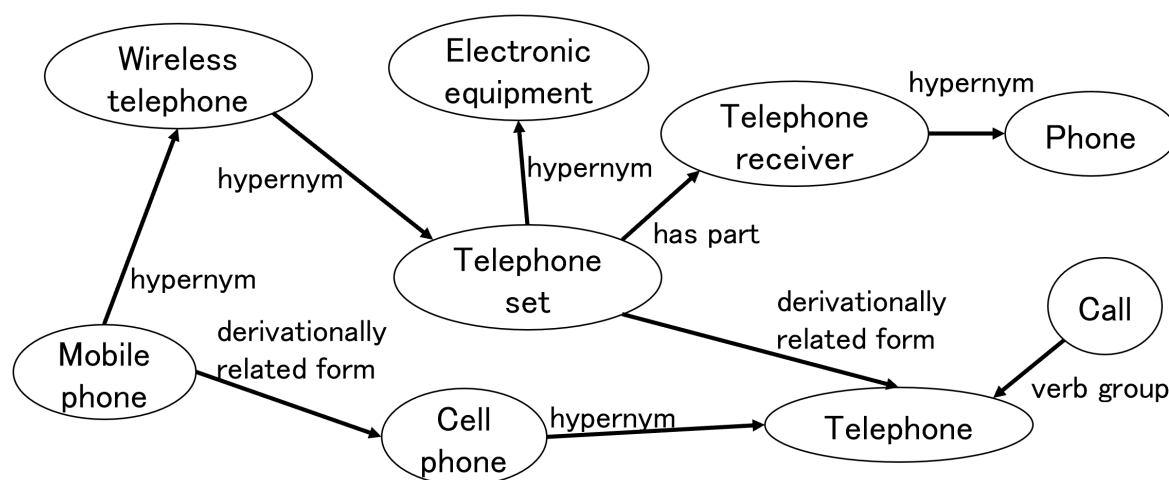


図 2.1: Knowledge Graph の例

Entity 同士の関係を細かく規定する. なお, Relation のもつ属性は有限の属性に限定される.

2.1.3 グラフ表現

Knowledge Graph はグラフとして表現できる. 図 2.1 の Knowledge Graph のグラフを例として説明する. 図 2.1 の Knowledge Graph は, WordNet の一部を Knowledge Graph として表現した一例である. Knowledge Graph をグラフとして表現する際には一般的に, Entity をグラフにおけるノード, Relation をグラフにおけるエッジとする. このようにして構成されるため, このグラフはノードおよびエッジの属性をもつ. また, 2 つの Entity の間に複数の関係がある場合は, 2 つの Entity の間に複数の Relation が与えられる. そのため, Knowledge Graph をグラフとして表現したものは重み付き多重グラフであるといえる.

図 2.1 に示すグラフにおけるノードが Entity であり, エッジが Relation となる. すなわち, この Knowledge Graph は “Wireless telephone”, “Telephone set”, “Telephone”, “Call” などの Entity をもち, “hypernym”, “has part”, “verb group”, “derivationally related form” の 4 種類の Relation をもつ. この Knowledge Graph は WordNet ^[7] を Knowledge Graph として表現した

一例であるため、各 Entity は属性として WordNet の各見出し語の単語をもつ。そして、Relation によってそれらの単語の関係が規定されている。例として、“hypernym” という Relation は単語間の上位下位の関係を規定するものである。そして、図 2.1 では、この属性をもつ Relation によって “Wireless telephone” から “Telephone set” へと繋がれている。この 2 つの Entity と 1 つの Relation によって、「“Wireless telephone” の上位語は “Telephone set”」という関係が定まる。

2.1.4 Triple

グラフとは異なる表現として、Relation で繋がれた 2 つの Entity に着目し、2 つの Entity とそれを結ぶ Relation をひとまとめにした Triple の集合として表現する事もできる。表 2.1 に Knowledge Graph を Triple として表現した例を示す。表 2.1 は、図 2.1 の一部の情報を Triple で表現している。

それぞれの Triple は、Relation の出発元の Entity, Relation, 向かう先の Entity をもつ。それぞれを Head, Relation, Tail と呼ぶ。本研究では、主に Triple 表現を利用する。

表 2.1: Knowledge Graph の Triple 表現

Head	Relation	Tail
Wireless telephone	hypernym	Telephone set
Telephone set	derivationally related form	Telephone
Telephone set	has part	Telephone receiver
Call	verb group	Telephone
Mobile phone	hypernym	Wireless telephone
⋮	⋮	⋮

2.1.5 埋め込み手法

Knowledge Graph はその要素として Entity 集合, Relation 集合をもち、それらは実数値や自然言語文などの属性をもつ。しかし Knowledge Graph を計算

機で扱う上で, さまざまな種類のデータによって構成されたそれらをそのまま用いることは容易ではない. そこで, 自然言語処理と同様に, Knowledge Graph の各要素を埋め込み表現にするさまざまな手法が考案されてきた. それらのうち代表的なものをいくつか紹介する. Knowledge Graph 埋め込みの手法を大きく分類すると, 以下の 3 つの種類がある [8].

行列変換に基づく手法

行列変換に基づく手法は, Triple の隣接行列を分解することで, Entity や Relation を表す埋め込み表現を得る. すなわち, $n = |\text{Entity}|$ 個の Entity と $m = |\text{Relation}|$ 個の Relation によって構成される Knowledge Graph の隣接行列 $V \in \mathbb{R}^{n \times m}$, $v_{x,y} \in \{0, 1, \dots, m-1\}$ を分解して, 各 Entity, Relation を埋め込み表現にする手法である.

これに分類される埋め込み表現獲得手法としては, RESCAL [9], DistMult [10], HolE [11], 複素数空間を用いる ComplEx [12] などがある.

ベクトルの移動に基づく手法

ベクトルの移動に基づく手法は, Entity や Relation の埋め込み表現を, 各 Triple, あるいは存在しない偽物の Triple に含まれる Head, Relation, Tail の埋め込み表現のベクトル演算によって score を算出して, その score が条件を満たすように最適化することでもっともらしい埋め込み表現を得る.

これに分類される埋め込み表現獲得手法の 1 つとして, TransE [13] があり, これを例に説明する. TransE は, 各 Entity および各 Relation をそれぞれ d 次元の埋め込み表現にする. TransE は Triple を構成する Head, Relation, Tail のそれぞれの埋め込み表現 $\mathbf{v}_{\text{Head}}, \mathbf{v}_{\text{Relation}}, \mathbf{v}_{\text{Tail}}$ に対して, 各 Triple が $\mathbf{v}_{\text{Head}} + \mathbf{v}_{\text{Relation}} = \mathbf{v}_{\text{Tail}}$ を満たすように埋め込み表現を学習する手法である. このとき, score は $\mathbf{v}_{\text{Head}} + \mathbf{v}_{\text{Relation}}$ と \mathbf{v}_{Tail} の乖離であり, 式で表すと

$$\text{score} = - \|\mathbf{v}_{\text{Head}} + \mathbf{v}_{\text{Relation}} - \mathbf{v}_{\text{Tail}}\| \quad (2.1)$$

と定義される. この score の値が正例の Triple に対して最大化して負例

の Triple に対して最小化するように、誤差逆伝播法を用いて埋め込み表現を学習する。

このように、埋め込み表現に対する演算によって score を算出して、より良い score が得られるように埋め込み表現を学習していく。TransE 以外の手法としては、TransE を改良した ITransE^[14], TransH^[15], TransR^[16], PTransE^[17], 複素数ベクトルを用いてその回転を利用する手法である RotatE^[18] がある。さらに、階層構造や木構造をうまく扱うことを期待してユークリッド空間ではなくポアンカレ空間^[19] に埋め込む Poincare Embedding^[20] を用いる手法の RotH^[21], AttH^[21] がある。

ニューラルネットワークに基づく手法

ニューラルネットワークに基づく手法は、Entity や Relation の埋め込み表現をニューラルネットワークによって最適化することで埋め込み表現を得る。

これに分類される埋め込み表現獲得手法としては、畳み込み器^[22] を用いて Head と Relation から Tail の埋め込み表現を推定し、訓練データにより埋め込み表現を最適化する ConvE^[23], 階層型 Transformer によって Triple の関係と各 Entity のもつすべての Triple の情報をモデルに入力とすることで、複雑な関係が含まれた Knowledge Graph に対応する HittER^[24] などがある。

また、Knowledge Graph に限定せず人間関係、Web ページのリンク、道路交通網、化学結合といったさまざまなグラフを埋め込むグラフニューラルネットワーク (Graph Neural Networks, GNN)^[25] についてもさまざまなものが提案されている。代表的なものとしては、グラフ構造を畳み込む Relational Graph Convolutional Networks (R-GCN)^[26], GNN に Attention 機構を導入した Graph Attention Networks (GAN)^[27] などがある。また、R-GCN を Knowledge Graph に応用し、購買推薦システムに用いる KG-CNN^[28] も提案されている。

2.1.6 活用例

Knowledge Graph はさまざまな種類のデータを統合的にグラフとして扱う。これにより、一度に多角的な情報を得ることができる。Knowledge Graph が利用されているものとして最も有名なものが Google Knowledge Graph である。これは Google が保有する各情報を Entity として、それぞれが Relation によって紐付けられた Knowledge Graph である。これを用いることでユーザーはある情報について検索した際にそれに付随する人や場所や物事等に関する多角的な情報を得ることができる。

また近年, Knowledge Graph を機械学習に用いることによってさまざまなデータ間の複雑なつながりを学習する事への期待が高まっている [29]。例えば過去の売買履歴や商品データによる購買推薦システムを作る際に Knowledge Graph を用いることで、顧客の購入履歴や品物の種類といったデータを総合的に扱い、商品と顧客のつながりを学習することができる。この際に、推論の根拠となる Entity や Relation の明確化により、解釈可能な推論が期待できる [28] [30]。

2.2 Transformer

Transformer [31] は従来用いられてきた Recurrent Neural Network (RNN) [32] を用いず Attention 機構 [33] のみを基本構造とするエンコーダデコーダモデルである。RNN は時系列データに対して有効な手法であるものの、単語の位置に従い計算をするため計算の並列化が難しい。そのため、多くの計算時間がかかるという欠点があった。一方で、Attention 機構のみを用いた本モデルは、行列計算の組み合わせのみで表現できるため、並列化が容易となった。図 2.2 にその概略図を示す。図 2.2 の左側がエンコーダ、右側がデコーダの役割をもつ。なお、入力を並列に扱うため、Position Encoding によって入力に位置情報を付与している。Transformer のエンコーダおよびデコーダはそれぞれ Self-Attention (自己注意機構) を基本構造にもつ。Self-Attention とは自己注意機構と呼ばれており、Attention 機構において特に Query, Key, Value が同一の物を指す。異なるデータ間の対応関係を獲得するのではなく入力データ内

の単語同士での類似度や重要度を獲得できる。これによって文章内での単語の依存関係を獲得することを想定している。そして、Attention によって得られるベクトルを正規化し、FeedForward 層に入れて各シーケンスに対するベクトルを得て、それを更に次の Attention 層へと入力する処理をエンコーダ、デコーダ共に $N \in \mathbb{N}$ 回繰り返す処理がなされている。

2.3 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) ^[4] は、2018 年に Google が発表した複数の双方向 Transformer に基づく汎用言語モデルである。BERT は、入力された単語列全体と、含まれる各単語に対応する分散表現を出力する。BERT は膨大なデータで教師なし学習をしたモデルである。これを各タスクに fine-tuning することでさまざまなタスクに柔軟に対応させることができる。

事前学習では、膨大なデータを用いて Masked Language Modeling (MLM) と Next Sentence Prediction (NSP) の 2 つの教師なし学習が並行してなされる。図 2.3 に BERT の事前学習と転移学習の概略図を示す。

MLM は、入力データの一部の単語を “[MASK]” トークンに置き換えて BERT の入力として、得られた各単語の分散表現から “[MASK]” トークン部分の単語の元の単語を予測する学習方法である。“[MASK]” がどの単語なのかを、周辺の文脈から推定するタスクを解くことで、BERT は文脈を考慮した単語埋め込みを生成する能力を獲得する。しかし、“[MASK]” というトークンは事前学習には登場するものの、fine-tuning の際にはタスクによっては登場せず、事前学習と fine-tuning に乖離が生じてしまう。その問題を緩和するため、“[MASK]” トークンに置き換えた単語に対して、一定の確率で元の単語に戻し、また一定の確率でランダムな単語に置き換える。BERT が提案された論文においては、全体の単語のうち 15% を “[MASK]” とし、置き換えられた単語のうち 10% を元の単語に戻し、10% をランダムな単語に置き換えている。

NSP は、本来ならば文脈的に連続する 2 文の入力に対して、一定の確率で不

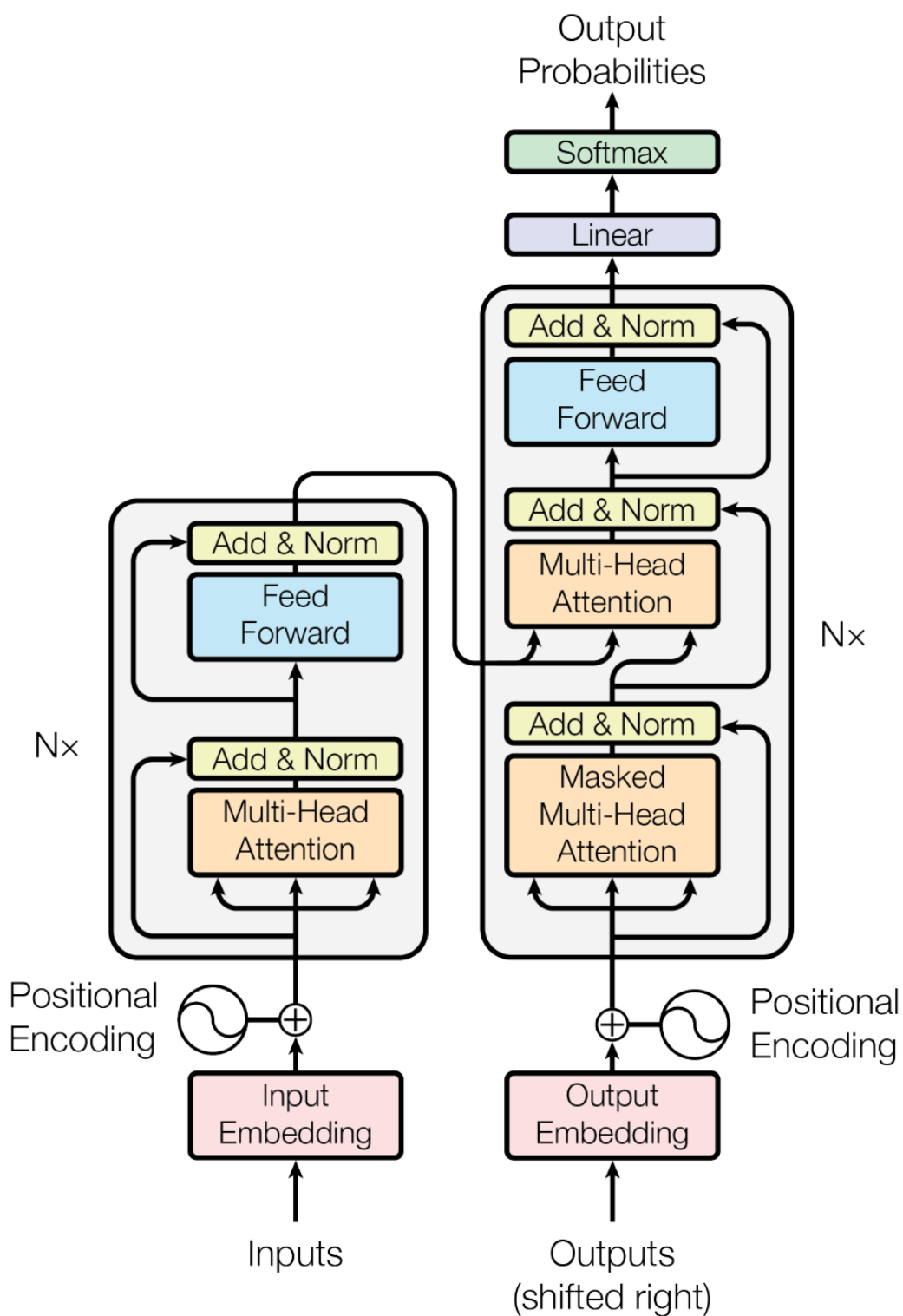


図 2.2: Transformer の概略図 (文献 [31] 図 1 参照)

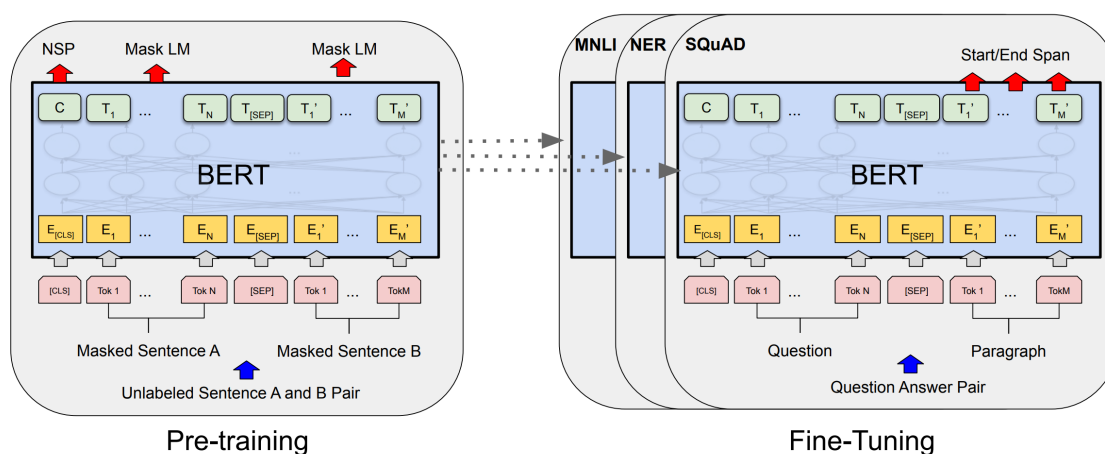


図 2.3: BERT の事前学習と転移学習の概略図 (文献^[4] 図 1 参照)

連続な 2 文に置き換えて、得られた単語列全体の分散表現からその 2 文が連続しているかどうかを推定する学習方法である。2 文の間の文脈を考慮してこのタスクを解くことで、BERT は、文の交互関係が必要となるタスクを解く能力を獲得する。

これらの事前学習によって得られた BERT を fine-tuning することで、BERT は複数のタスクにおいて当時での最高峰精度を達成した。なお、BERT の登場以降、BERT はモデル構造に見合った性能を発揮できていないとして、BERT の訓練データを変え、事前学習から NSP を除き、動的マスキングをするなどの調整により、より良い精度を達成した RoBERTa^[34] や、BERT のモデルは肥大だとして、推論能力をほとんど保持したままモデルサイズを大幅に減らした DistilBERT^[35] といった、BERT に改良を加えたモデルが複数提案されている。

2.4 Knowledge Graph BERT (KG-BERT)

Knowledge Graph BERT (KG-BERT)^[6] は、Yao らによって提案された BERT を fine-tuning することで Knowledge Graph を補完する手法である。図 2.4 に KG-BERT の概略図を示す。Entity と Relation はその名前や説明を単語のシーケンスとして表現し、それを入力文として BERT モデルを fine-tuning

する。文は実際の言語的な文ではなく、任意の連続したテキストまたは単語のシーケンスである。Triple (Head, Relation, Tail) の文章を 1 つのシーケンスとしてまとめ、BERT への入力トークンシーケンスとして扱う。

図 2.4 において、Tok はトークンを、 \mathbf{E} , \mathbf{T} は埋め込みベクトルとモデルの出力のベクトルをそれぞれ示す。 a, b, c は Head, Relation, Tail のトークン数を、 h, r, t は Head, Relation, Tail をそれぞれ表す。入力シーケンスの最初のトークンは常に特別なトークン “[CLS]” である。Head Entity はトークン $\text{Tok}_1^h, \dots, \text{Tok}_a^h$ を含む文で表現され、Relation はトークン $\text{Tok}_1^r, \dots, \text{Tok}_b^r$ を含む文で表現され、Tail Entity はトークン $\text{Tok}_1^t, \dots, \text{Tok}_c^t$ を含む文で表現される。Entity と Relation の文は特別なトークン “[SEP]” で区切られる。同じ位置の異なるトークンは同じ位置埋め込みを共有し、“[SEP]” で区切られた異なる要素は異なるセグメント埋め込みをもつ。各入力トークンの入力表現は、対応するトークン、セグメント、および位置埋め込みを合計して構築され、異なる “[SEP]” で区切られた要素は異なるセグメント埋め込みをもつ。同じ位置 $i \in \{1, 2, 3, \dots, 512\}$ の異なるトークンは同じ位置埋め込みを共有する。入力トークンの表現は、元の BERT の実装に基づく多層双方向 Transformer エンコーダである BERT モデルアーキテクチャに供給される。特別な “[CLS]” トークンに対応する最終的な隠れベクトルと i 番目の入力トークンの最終的な隠れベクトルは、それぞれ $\mathbf{C} \in \mathbb{R}^H$ および $\mathbf{T}_i \in \mathbb{R}^H$ で表される。ここで、 H は BERT における隠れベクトルのサイズである。“[CLS]” に対応する最終的な隠れベクトル \mathbf{C} は、Triple スコアを計算するための集約シーケンス表現として使用される。Triple $\tau = (h, r, t)$ のスコア関数は

$$\mathbf{s}_\tau = f(h, r, t) = \text{sigmoid}(\mathbf{C}\mathbf{W}^T) \quad (2.2)$$

であり、 \mathbf{s}_τ は $s_{\tau 0}, s_{\tau 1} \in [0, 1]$ および $s_{\tau 0} + s_{\tau 1} = 1$ となる 2 次元の実ベクトルである。

正例の Triple セット \mathbb{D}^+ およびそれに応じて構築された負例の Triple セット \mathbb{D}^- が与えられた場合、Triple のラベルに対するクロスエントロピー損失が計算される。

$$L = - \sum_{\tau \in \mathbb{D}^+ \cup \mathbb{D}^-} (y_\tau \log s_{\tau 0} + (1 - y_\tau) \log s_{\tau 1}) \quad (2.3)$$

ラベル $y_\tau \in \{0, 1\}$ はその Triple のラベル (負または正) である. 負例の Triple セット \mathbb{D}^- は, 正例の Triple $(h, r, t) \in \mathbb{D}^+$ の Head Entity h または Tail Entity t をランダムな Entity h' または t' で置き換えることで生成される.

$$\begin{aligned} \mathbb{D}^- = & \{(h', r, t) | h' \in \mathbb{E} \wedge h' \neq h \wedge (h', r, t) \notin \mathbb{D}^+\} \\ & \cup \{(h, r, t') | t \in \mathbb{E} \wedge t' \neq t \wedge (h, r, t') \notin \mathbb{D}^+\} \end{aligned} \quad (2.4)$$

ここで, \mathbb{E} は Entity の集合である. 負例として扱うための条件は, その Triple が既に正例セット \mathbb{D}^+ に存在していない場合である. また, 事前に学習されたパラメータの重みと新しい重み $W \in \mathbb{R}^{2 \times H}$ は勾配降下法を使用して更新される.

Triple Label $y \in \{0, 1\}$

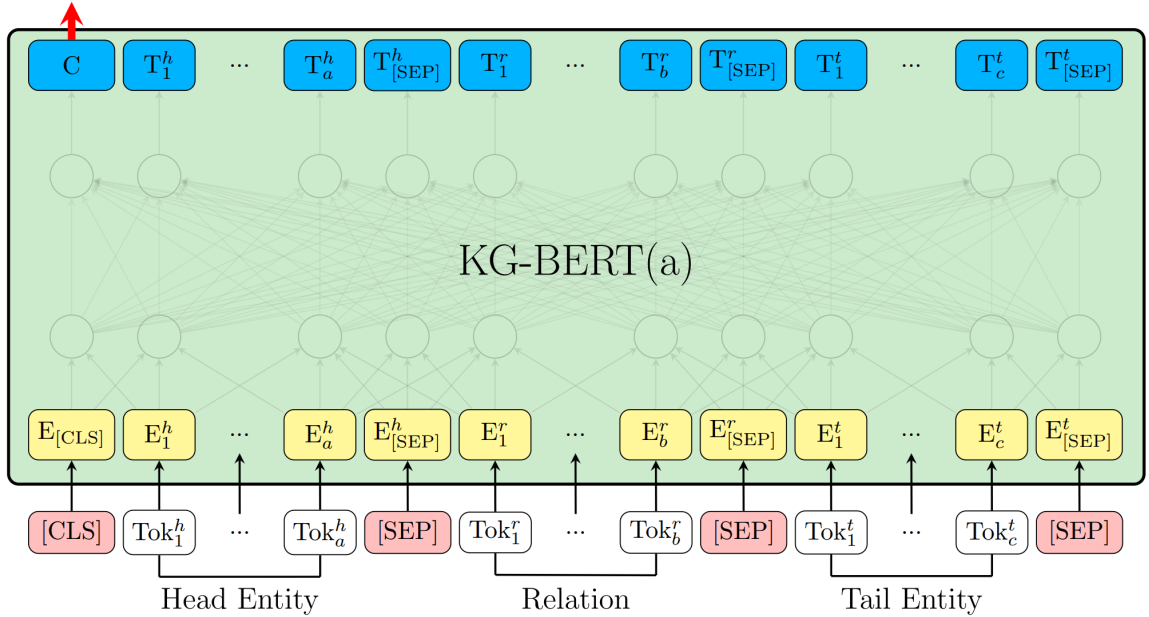


図 2.4: KG-BERT (文献 [6] 図 1 参照)

2.5 WN18RR

WN18RR^[23] は, Dettmers らによって WordNet^[7] を基に作成された Triple のデータセットである.

WordNet は, 多言語で構成される大規模な語彙データベースであり, 単語同士の関係が品詞別に階層構造の形で格納されている. WordNet では Entity は単語の意味に対応し, Relation は Entity 間の語彙的な関係を表している. WordNet に含まれる関係の例として以下のものが挙げられる.

- Synonymy (同義語)
- Hypernymy (上位語)
- Hyponymy (下位語)
- Holonymy (全体語)
- Meronymy (部分語)

WN18^[13] は WordNet のサブセットであり, 18 種類の Relation と 40,943 種類の Entity で構成されている. WN18 には WordNet の Relation と Entity から作成された 151,442 個の Triple が存在し, その中に hyponym (下位語) の関係をもつ Triple と hypernym (上位語) の関係をもつ Triple が含まれている. このように, 階層構造となっている WN18 には逆関係の Triple が複数含まれている. そのため, WN18 では逆関係の Triple によるテスト漏洩の問題が生じる場合がある. 例えば, ある Relation r をもつ Triple (e_1, r, e_2) とその逆関係の Triple (e_2, r', e_1) が存在してそれぞれが訓練データ, テストデータに割り振られている場合, テストデータには訓練データに含まれる Triple と似ている Triple が存在することになる.

そこで, 逆関係のテスト漏洩がないデータセットとして WN18RR が作成された. WN18RR は WN18 から逆関係の Triple を除去して片方の Triple に統一したデータセットである. 表 2.2 に WN18 から逆関係の Triple を除去する例を示す. ある Relation r をもつ Triple (e_1, r, e_2) とその逆関係の Triple (e_2, r', e_1) の場合, Triple (e_1, r, e_2) を残してその逆関係の Triple (e_2, r', e_1) を除

表 2.2: WN18 から逆関係の Triple を除去する例

	Head	Relation	Tail
	poodle dog	hypernym	domestic dog
除去	domestic dog	hyponym	poodle dog
	telephone	hypernym	telecommunicate
除去	telecommunicate	hyponym	telephone
	home movie	hypernym	picture show
除去	picture show	hyponym	home movie
	⋮	⋮	⋮

表 2.3: WN18RR における Entity の例

caption	description
poodle dog	an intelligent dog with a heavy curly solid-colored ...
domestic dog	a member of the genus Canis (probably descended ...
telephone	get or try to get into communication (with ...
wireless telephone	a telephone that communicates by radio waves ...
telephone set	electronic equipment that converts sound into ...
	⋮

去する. これによって作成された WN18RR は 40,943 種類の Entity と 11 種類の Relation, 93,003 個の Triple をもつ. Entity はそれぞれ“(見出し語), (説明文)”の形で表される. 表 2.3, 2.4 に WN18RR における Entity と Triple の例をそれぞれ示す.

表 2.4: WN18RR における Triple の例

Head	Relation	Tail
telephone	hypernym	telecommunicate
telephone system	has part	telephone set
telephone system	has part	telephone exchange
telephone system	hypernym	communication system
telephone set	derivationally related form	telephone
call	verb group	telephone
cell phone	hypernym	telephone
⋮	⋮	⋮

3 提案手法

本章では, Knowledge Graph 補完および, 本研究における提案手法についての説明をする. 本研究における提案手法はモデルの提案とモデルの学習方法の提案に分けられる. これらの提案手法により, 言語情報を含む Knowledge Graph を扱う従来研究の問題点の解決を目指す.

Knowledge Graph 補完について説明する. Knowledge Graph のある Triple における Head, Relation, Tail をそれぞれ h, r, t とすると, Knowledge Graph 補完では Triple (h, r, t) に対して “ t ” に入る Tail を回答することで Entity 間の関係性を予測する.

従来の Knowledge Graph 補完手法として Knowledge Graph Embedding 手法である TransE^[13] や ComplEx^[12] などがある. これらの手法は Entity と Relation をそれぞれ実数値ベクトルとして表し, それらのベクトルの関係式を用いて Triple の妥当性を評価する.

例えば TransE では, まず Entity と Relation をそれぞれ d 次元の埋め込み表現にする. このとき, 与えられる埋め込み表現はランダムに初期化されたベクトルとする. 次に Triple を構成する Head, Relation, Tail の埋め込み表現 $\mathbf{v}_h, \mathbf{v}_r, \mathbf{v}_t$ に対して, すべての Triple が “ $\mathbf{v}_h + \mathbf{v}_r = \mathbf{v}_t$ ” を満たすように埋め込み表現を学習する. これにより Head と Relation から Tail を予測することができる.

このように, 既存の Knowledge Graph 補完手法は Triple の構造情報を重視した手法となっており, Entity 自体の意味情報を重視して補完する形ではないという特徴が挙げられる.

本研究では, Entity 自体の意味情報を効果的に捉えるために BERT を MLM で fine-tuning したモデルを用いて, 1 つのシーケンス (Head, Relation, [MASK]) を入力文として Tail を予測することによる Knowledge Graph 補完手法を提案する. 図 3.1 に提案手法のモデル概略図を示す. 図 3.1 の Tok はトークンを, \mathbf{E}, \mathbf{T} は埋め込みベクトルとモデルの出力のベクトルをそれぞれ示す. l, m, n は Head, Relation, Tail のトークン数を, h, r, t は Head, Relation, Tail をそれぞれ表す. 入力シーケンスの最初のトークンは常に特別なトークン “[CLS]” である. Head はトークン $\text{Tok}_1^h, \dots, \text{Tok}_l^h$ を含む文, Relation はトークン Tok_1^r, \dots

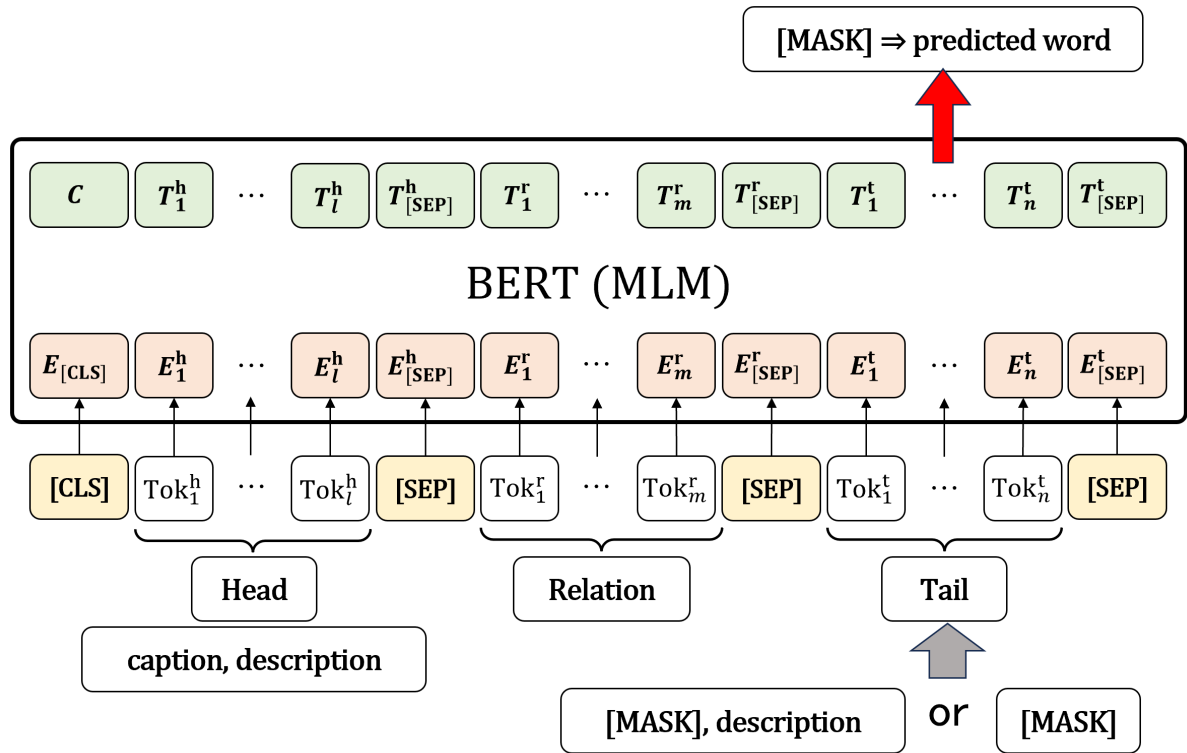
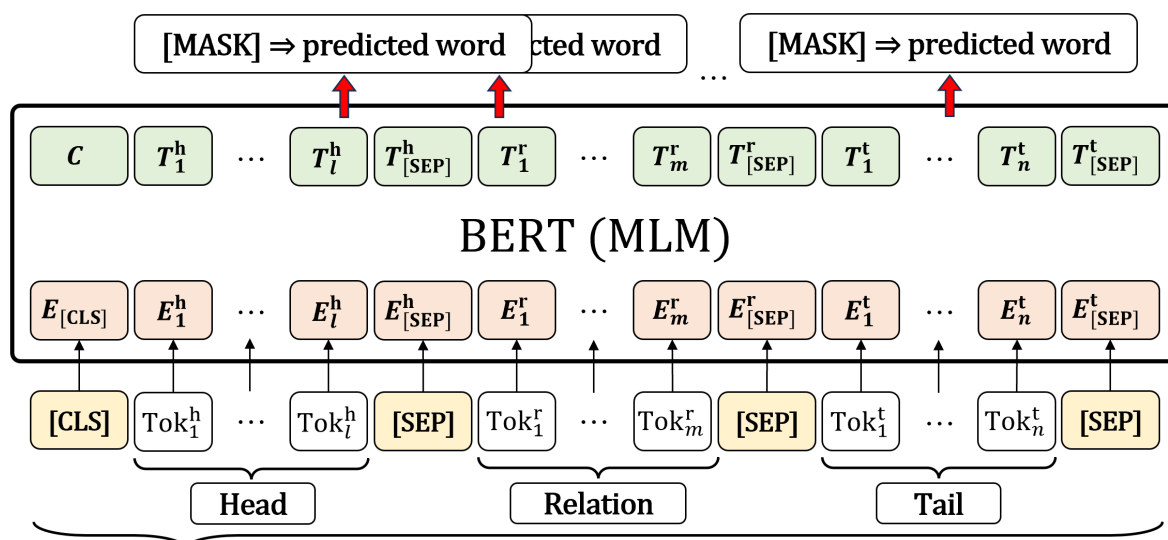


図 3.1: 提案手法のモデル概略図

, Tok_m^r を含む文, Tail はトークン Tok_1^t, \dots, Tok_n^t を含む文で表現される. Head, Relation, Tail の文は特別なトークン “[SEP]” で区切られる. このトークン列のうち, Tail の見出し語のみ, もしくは Tail 全体を “[MASK]” トークンに置き換えたシーケンスを BERT への入力とする. これにより, “[MASK]” トークンとした箇所の予測単語が出力される.

図 3.2 に BERT の MLM による fine-tuning の概略図を示す. 図 3.2 の記号は図 3.1 の記号と同様である. 入力シーケンスの最初のトークンは常に特別なトークン “[CLS]” である. Head はトークン Tok_1^h, \dots, Tok_l^h を含む文, Relation はトークン Tok_1^r, \dots, Tok_m^r を含む文, Tail はトークン Tok_1^t, \dots, Tok_n^t を含む文で表現される. Head, Relation, Tail の文は特別なトークン “[SEP]” で区切られる. さらに, BERT が提案された論文^[4]と同様に入力トークンの 15% のうち, 80% を “[MASK]” トークンに置き換え, 10% をランダムなトークンに置き換え, 10% をそのままのトークンにする. そして, 入力トークンの 15% に Tail の見出し語を必ず含むように設定する. このシーケンスを BERT への入力と



Out of the 15%,
80% is masked, 10% is random, and the remaining 10% is left unchanged.

└─ Within the 15%, the caption of the tail is always included.

図 3.2: BERT の MLM による fine-tuning の概略図

し, “[MASK]” トークンとした箇所, もしくはランダムなトークンとした箇所の予測単語が元の単語となるように学習する.

4 数値実験

本章では, 提案手法の有効性を確認するために実施した 2 種類の実験について説明する. 提案手法の有効性は 2 種類の実験の結果を 2.4 節で説明した KG-BERT を用いた結果と比較することで確認した.

4.1 データセット

本研究では, Knowledge Graph として 2.5 節で説明した WN18RR を用いた. このデータセットは Head, Relation, Tail を Triple として保存している. WN18RR の Triple を比較手法の KG-BERT モデルの入力文と対応させるため, 先頭を “[CLS]” トークンとして Head, Relation, Tail を “[SEP]” トークンで区切ったトークン列を入力文とした. 表 4.1 に例を示す.

以下の 2 種類の実験では, WN18RR の Triple を訓練データ, 検証データ, テストデータとしてそれぞれ 8 : 1 : 1 に分割して用いた. 表 4.2 に WN18RR の内訳を示す.

4.2 実験 1

実験 1 では, 提案するモデルおよび学習方法の有効性を確認するため, 訓練データを用いて BERT を MLM で fine-tuning し, このモデルを用いてテストデータの Tail を予測させることでモデルの精度を確認した.

図 4.1 に実験 1 のモデル概略図を示す. BERT を MLM で fine-tuning したモデルに対して, Triple “[CLS] Head [SEP] Relation [SEP] Tail [SEP]” における Tail の見出し語を “[MASK]” トークンに置き換えた文 “[CLS] Head [SEP] Relation [SEP] [MASK], Tail の説明文 [SEP]” を 1 つのシーケンスとして入力する. 図 3.1 において “Tail” を “[MASK], description” とした場合である. これにより, “[MASK]” トークンとなった Tail の見出し語を予測する. なお, 予測する Tail の見出し語の説明文をそのまま入力しているため, 通常想定される Tail 予測タスクより Tail の予測が容易である.

表 4.3 に比較手法の KG-BERT と実験 1 のパラメータを示す. なお, KG-BERT の eval batch size と max seq length はメモリ容量の都合上文献値とは

表 4.1: データセットをトークン列とした例

[CLS]Head[SEP]Relation[SEP]Tail[SEP]
[CLS]telephone set, electronic equipment that converts sound into electrical signals that ...[SEP]has part[SEP]telephone receiver, earphone that converts electrical signals into sounds[SEP]
[CLS]wireless telephone, a telephone that communicates by radio waves rather than along cables[SEP]hypernym[SEP]telephone set, electronic equipment that converts sound into electrical signals that ...[SEP]
[CLS]telephone receiver, earphone that converts electrical signals into sounds[SEP]hypernym[SEP]phone, electro-acoustic transducer for converting electric signals into sounds; ...[SEP]
⋮

表 4.2: WN18RR の内訳

Entity	Relation	Triple	訓練	検証	テスト
40,943	11	93,003	86,835	3,034	3,134

異なっている。また、表中の「- (ハイフン)」はそのモデルには必要のないパラメータを表している。

4.3 実験 2

実験 2 では、提案するモデルおよび学習方法の有効性を確認するため、訓練データを用いて BERT を MLM で fine-tuning し、このモデルを用いてテストデータの Tail を予測させることでモデルの精度を確認した。なお、実験 1 では Tail 予測のヒントとして Tail の説明文も入力していたのに対し、実験 2 では Tail の説明文を入力しないものとする。

図 4.2 に実験 2 のモデル概略図を示す。実験 1 の入力文 “[CLS] Head [SEP] Relation [SEP] [MASK], Tail の説明文 [SEP]” における Tail の説明文を除去した文 “[CLS] Head [SEP] Relation [SEP] [MASK] [SEP]” を 1 つのシーケン

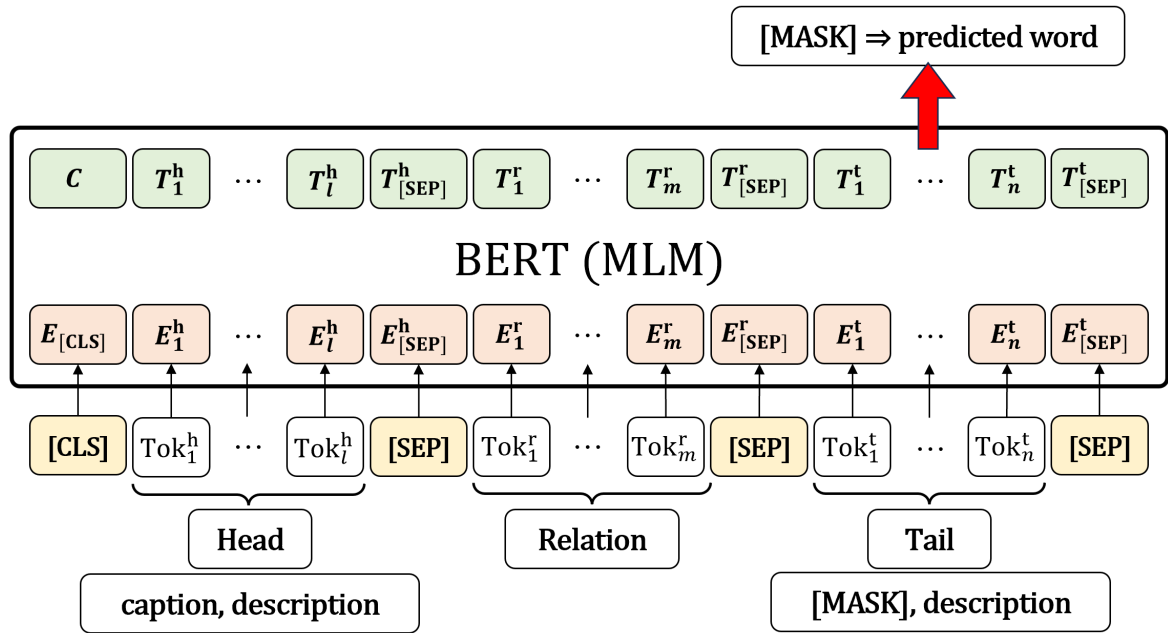


図 4.1: 実験 1 のモデル概略図

スとして入力する. この場合, 図 3.1 における “Tail” は “[MASK]” のみとなる. これにより, “[MASK]” トークンとなった Tail の見出し語を予測する. なお, 予測する Tail の見出し語の説明文は入力していないため, Head と Relation の情報のみを用いて Tail を予測する形となる.

実験 2 のパラメータは実験 1 と同様であり, 表 4.3 に示している.

4.4 評価指標

評価指標として Mean Reciprocal Rank (MRR), Hits@ k , Filtered MRR, Filtered Hits@ k を用いる. 予測結果の r 番目に正解があるとき, その順位 r のことを Rank と呼ぶ. $|T|$ を Triple 数, r_i を $Triple_i$ における正解 Tail の Rank とすると, MRR は (4.1) 式で表される.

$$MRR = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{1}{r_i} \quad (4.1)$$

Hits@ k は, Tail 予測において上位 k 個以内に正解の要素が出力されている割合を表す. このとき, あるテスト Triple の正解 Tail より予測結果の上位にその

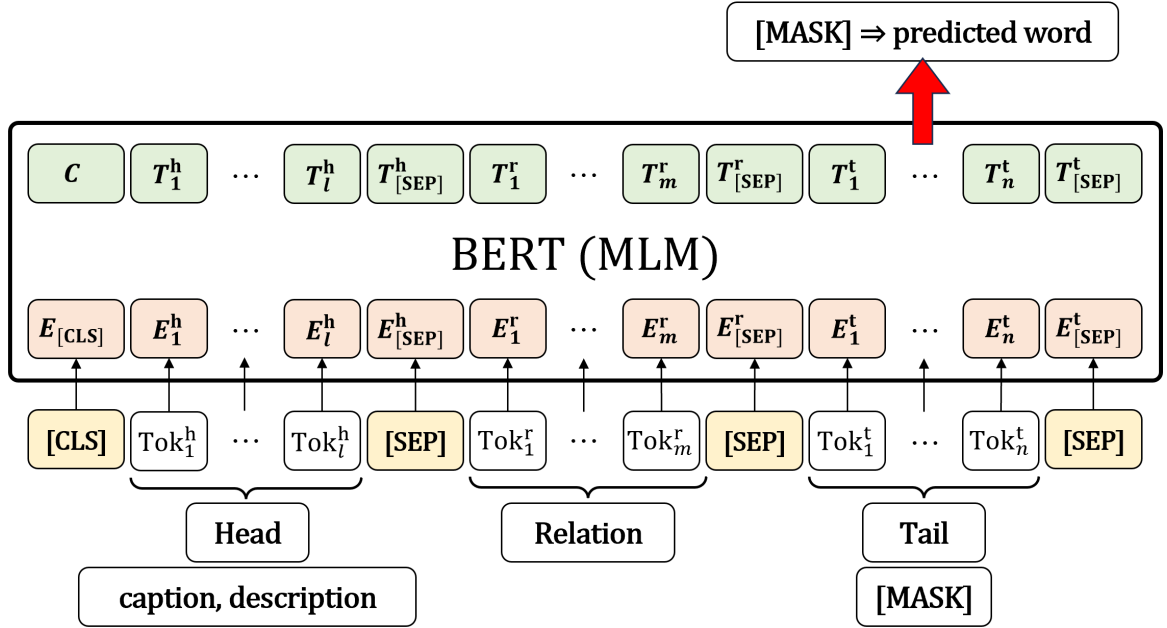


図 4.2: 実験 2 のモデル概略図

テスト Triple と同じ Head と Relation をもつ Triple の Tail が存在する場合, その分低くランク付けされる. そこで, テスト Triple と同じ Head と Relation をもつ Triple の Tail を予測結果から除去してランク付けをする. この Rank を Filtered Rank とする. 表 4.4 に Filtered Rank の例を示す. 表 4.4 において, テスト Triple が (Head, Relation, Word r) で, データとして Triple (Head, Relation, Word 2), (Head, Relation, Word 4) の 2 つが存在するとき, Rank を 2 つ繰り上げることで Filtered Rank を得る. Filtered Rank によって計算される MRR, Hits@ k をそれぞれ Filtered MRR, Filtered Hits@ k とする. 本研究では $k = 1, 3, 10$ で評価した. MMR, Hits@ k , Filtered MRR, Filtered Hits@ k はすべて値が大きいと推定精度が良いと判断される.

4.5 実験結果

表 4.5 に KG-BERT と実験 1, 2 の結果を示す. KG-BERT は本実験と Tail 予測方法が異なるため単純な比較はできないが, 参考として文献値と再現実験の 2 つの結果を示している. 表中の「- (ハイフン)」は文献 [6] に記載されてい

表 4.3: KG-BERT と実験 1, 2 のパラメータ

パラメータ	KG-BERT (文献値)	実験 1, 2
leaning rate	5.0×10^{-5}	5.0×10^{-5}
mask probability	-	0.15
batch size	32	32
eval batch size	128 (5000)	-
max seq length	32 (50)	128
epoch	5	20

表 4.4: Filtered Rank の例

Triple	Rank	Tail 予測結果	Filtered Rank
	1	Word 1	1
存在	2	Word 2 除去	-
	3	Word 3	2
存在	4	Word 4 除去	-
	\vdots	\vdots	\vdots
正解	r	Word r	$r - 2$

なかったことを表している。

ここで、本実験と KG-BERT の Rank の所得方法について説明する。本実験では、“[MASK]” トークンに置き換えた箇所の単語を上位 300 位まで予測し、正解 Tail がある順位を Rank とする。予測できなかった Tail の Rank は Entity 数である 40,943 とする。KG-BERT では、負例の Triple がある正例の Triple の Tail を他の Entity に置き換えて作成され、それらの Triple をそれぞれ BERT に入力する。これにより、入力した Triple が存在するか否かを 2 値分類によって判定する。このときに得られる分類スコアを正例の Triple とそれによって作成される複数の負例の Triple に対して降順に並べて順位付けをし、得られた正例の Triple の順位を Rank とする。

表 4.5 より、実験 1 ではすべての評価指標において KG-BERT の結果を上

表 4.5: KG-BERT と実験 1, 2 の結果

モデル	WN18RR			
	MRR	Hits@1	Hits@3	Hits@10
KG-BERT (文献値)	-	-	-	52.4
KG-BERT (再現実験)	0.25	12.41	29.44	51.85
実験 1	0.546	52.55	56.16	57.79
実験 1 (Filtered)	0.550	52.81	56.38	57.82
実験 2	0.168	10.94	19.37	27.95
実験 2 (Filtered)	0.169	11.04	19.66	28.05

回る結果が得られた。これにより, Tail 予測に BERT の MLM を利用することの有効性が確認できた。しかし, 通常想定される Tail 予測タスクより Tail の予測が容易であるため, KG-BERT と単純な比較はできない。また, Filtered MRR, Filtered Hits@ k について MRR, Hits@ k と比較するとその値の変化は微細な範囲にとどまった。

実験 2 では Hits@1 において KG-BERT と同程度の結果が得られたが, 他の評価指標においては下回る結果となった。また, Filtered MRR, Filtered Hits@ k については実験 1 と同様に MRR, Hits@ k より少し向上したがほとんど変化していなかった。

4.6 考察

本節では Triple を (“Head の見出し語, その説明文”, “Relation”, “Tail の見出し語, その説明文”) とし, Tail の見出し語を太字で表す。

Tail 予測の結果について本実験と KG-BERT の再現実験を比較する。

実験 1 でテスト Triple (“position, the particular portion of space occupied …”, “hypernym”, “**point**, the precise location of something; a spatially limited location; “she walked to a point where she could survey the whole street””) の見出し語 “point” を “[MASK]” に置き換えて Tail を予測すると, 提案手法では “point (地点)” を予測して正解したのに対し, KG-BERT では “situation

(状況)”を予測していた。これは Tail の見出し語の説明文に “point” が含まれていることが予測に影響を与えたと考えられる。そのため、Tail の説明文の情報が結果に大きく影響することがわかった。

実験 2 でテスト Triple (“evidence, an indication that makes something evident; …”, “hypernym”, “**indication**, something that serves to indicate or suggest; …”) の見出し語 “indication” とその説明文を “[MASK]” に置き換えて Tail を予測すると、提案手法では “indication (指示)” を予測して正解したのに対し、KG-BERT では “avermment (喚起)” を予測していた。これは Head の見出し語の説明文に “indication” が含まれていることが予測に影響を与えたと考えられる。このような Head の見出し語の説明文に Tail の情報を含む Triple に対して提案手法は有効であることがわかった。

本実験の Tail 予測結果について考察する。

実験 1 でテスト Triple (“signer, someone who can use sign language to communicate”, “hypernym”, “**communicator**, a person who communicates with others”) の見出し語 “communicator” を “[MASK]” に置き換えて Tail を予測すると、“communicate” を予測していた。“communicator (コミュニケーションをとる人)” と “communicate (コミュニケーションをとる)” は派生語であり意味が近い単語として捉えることができる。また、実験 2 でテスト Triple (“existence, the state or fact of existing; “a point of view gradually coming into being”; “laws in existence for centuries””, “derivationally related form”, “**existent**, presently existing in fact and not merely potential or possible; …”) の見出し語 “existent” を “[MASK]” に置き換えて Tail を予測すると、“existence” を予測していた。同様に “existent (現存する)” と “existence (存在)” は派生語であり意味が近い単語として捉えることができる。このように、実験 1, 2 において正解 Tail の派生語であり意味が近い単語を予測できる可能性があることがわかる。しかし、本実験ではこのような予測結果に対して間違いであると評価している場合がある。そこで、予測結果が正解 Tail の派生語であり意味が近い単語であることを考慮する評価方法を追加することを今後検討する。

実験 1, 2 の Filtered MRR, Filtered Hits@ k においてどちらも MRR, Hits@ k

と比較するとその値の変化は微細な範囲にとどまった. Filtered MRR, Filtered Hits@ k について考察する.

実験 1 では, 予測する Tail の説明文を入力していることにより Head と Relation が同じ他の Triple の Tail を予測する可能性が少ないと考えられる. 実験 2 では, テスト Triple (“vanish, become invisible or unnoticeable; “The effect vanished when day broke””, “derivationally related form”, “**disappearance**, the event of passing out of sight”) の見出し語 “disappearance” とその説明文を “[MASK]” に置き換えて Tail を予測すると, 予測結果 2 番目に “fade (次第に消える)” があり, “disappearance (消滅)” を予測できていない例があった. なお, 訓練データ内に テスト Triple の Head と Relation が同じで Tail が “fade” の Triple (“vanish, become invisible or unnoticeable; “The effect vanished when day broke””, “derivationally related form”, “**fade**, gradually ceasing to be visible”) が存在している. このように, 訓練データ内にある Triple の影響で Tail を予測できないことが MRR, Hits@ k と Filtered MRR, Filtered Hits@ k の値に変化がない原因であると考えられる.

本研究では, MLM の出力は BERT に登録されている単語すべてが候補となっているため, WN18RR データセット内に Entity として存在しない単語も出力される可能性がある. そこで, MLM の出力候補を WN18RR データセット内の Entity のみに限定したモデルの作成を検討し, Tail の予測精度の向上を目指す.

5 まとめと今後の課題

本研究では、言語情報を含む Knowledge Graph に対して BERT の MLM を用いて Knowledge Graph 補完として Tail を予測するモデルおよびその学習方法を提案した。

Tail の説明文も入力する実験 1 において比較手法の KG-BERT を上回る結果が得られた。これにより、BERT の MLM を用いた Knowledge Graph 補完手法の有効性が確認できた。Head と Relation の情報のみから Tail を予測する実験 2 において Hits@1 は KG-BERT と同程度の結果が得られたが、その他の評価指標は KG-BERT を下回る結果となった。また、提案手法で予測でき、KG-BERT で予測できなかった Triple があることを確認した。実験 1 では Tail の説明文に含まれる単語を重視して Tail を予測し、実験 2 では Head の説明文に含まれる単語を重視して Tail を予測することで KG-BERT では予測できなかった Tail を予測できたと考えられる。

今後の課題として、予測精度の向上のためにパラメータの調整や MLM の出力候補を BERT に登録されている単語ではなく WN18RR の Entity に存在する単語のみに限定したモデルの作成が挙げられる。さらに、本研究では計測できていない予測結果の Rank の平均を評価指標として追加することや、提案手法の実用的な使用法について Knowledge Graph 補完だけでなく他の使用法を検討することなどがある。例えば、既存の Knowledge Graph において新たな関係性の発見の手助けなどが挙げられる。

謝辞

本研究を進めるにあたり御指導, 御鞭撻を賜りました森直樹教授に深く感謝申し上げます。直接御指導頂きました岡田真助教には, 研究のアイデアや方針だけでなく, 論文の書き方や発表の方法に関する事など, 多岐にわたって多大な御助言を頂きました。心より御礼申し上げます。そして, 日頃から研究に関して建設的な意見をしてくださった諸先輩方, 専門的観点から意見をくださった同じ研究グループの皆さん, とともに支え合いながら研究に取り組んできた同期の皆さんに感謝致します。

2024年2月22日

参考文献

- [1] 川村隆浩, 江上周作, 田村光太郎, 外園康智, 鵜飼孝典, 小柳佑介, 西野文人, 岡嶋成司, 村上勝彦, 高松邦彦, 杉浦あおい, 白松俊, 張翔宇, 古崎晃司. 第 1 回ナレッジグラフ推論チャレンジ 2018 開催報告 —説明性のある人工知能システムを目指して—. 人工知能 34 巻 3 号, 2019.
- [2] Amit Singhal. Introducing the Knowledge Graph: things, not strings. *Official google blog*, 2012. 2020-11-13.
- [3] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, June 2018.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, June 2019.
- [5] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *CoRR*, Vol. abs/1906.08237, , 2019.
- [6] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: BERT for Knowledge Graph Completion. *CoRR*, Vol. abs/1909.03193, , 2019.
- [7] Christiane Fellbaum. In Theory and applications of ontology: computer applications. *Springer*, pp. 231–243, 2010.

- [8] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29, No. 12, pp. 2724–2743, 2017.
- [9] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. *International Conference on Machine Learning*, 2011.
- [10] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *Proceedings of the International Conference on Learning Representations*, 2015.
- [11] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic Embeddings of Knowledge Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, No. 1, 2015.
- [12] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. *PMLR*, Vol. 48, pp. 2071–2080, 20–22 Jun 2016.
- [13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. *Advances in Neural Information Processing Systems*, Vol. 26, , 2013.
- [14] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative Entity Alignment via Joint Knowledge Embeddings. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 4258–4264, 2017.
- [15] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28, No. 1, Jun 2014.

- [16] Hailun Lin, Yong Liu, Weiping Wang, Yinliang Yue, and Zheng Lin. Learning Entity and Relation Embeddings for Knowledge Resolution. *Procedia Computer Science*, Vol. 108, pp. 345–354, 2017. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.
- [17] Yimei Li, Donghong Qin, and Xiaodong Yang. Path Modeling based on Entity-Connectivity for Knowledge Base Completion. *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 984–989, 2020.
- [18] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *CoRR*, Vol. abs/1902.10197, , 2019.
- [19] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, Vol. 82, No. 3, sep 2010.
- [20] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. *CoRR*, Vol. abs/1705.08039, , 2017.
- [21] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6901–6914, July 2020.
- [22] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, pp. 640–651, 2017.
- [23] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. *AAAI Conference on Artificial Intelligence*, 2017.

- [24] Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. HittER: Hierarchical Transformers for Knowledge Graph Embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10395–10407, November 2021.
- [25] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, Vol. 1, pp. 57–81, 2020.
- [26] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. *The Semantic Web*, pp. 593–607, 2018. 15th International Conference on Extended Semantic Web Conference, ESWC 2018 ; Conference date: 03-06-2018 Through 07-06-2018.
- [27] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio. Graph Attention Networks. *ArXiv*, Vol. abs/1710.10903, , 2017.
- [28] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pp. 3307–3313, May 2019. 2019 World Wide Web Conference, WWW 2019 ; Conference date: 13-05-2019 Through 17-05-2019.
- [29] 松島ひろむ, 森澤竣, 石山琢己, 山名早人. 推薦システムにおける推薦理由の説明可能性に関するサーベイ. 情報科学技術フォーラム講演論文集, Vol. 20th, , 2021.
- [30] Huanyu Zhang, Xiaoxuan Shen, Baolin Yi, Wei Wang, and Yong Feng. KGAN: Knowledge Grouping Aggregation Network for course recommendation in MOOCs. *Expert Systems with Applications*, Vol. 211, p. 118344, 2023.

- [31] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *Neural Information Processing Systems*, 2017.
- [32] Alex Sherstinsky. Deriving the Recurrent Neural Network Definition and RNN Unrolling Using Signal Processing. *32nd Conference on Neural Information Processing Systems*, 12 2018.
- [33] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, Vol. abs/1409.0473, , 2014.
- [34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, Vol. abs/1907.11692, , 2019.
- [35] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, Vol. abs/1910.01108, , 2019.