Vol.2021-DBS-173 No.9
Vol.2021-IFAT-144 No.9
2021/9/16

知識グラフにおける更新可能性が高いエンティティの検出

大倉真一希 [1]　　天笠俊之 [2]

[1] 筑波大学大学院 システム情報工学研究科 コンピュータサイエンス専攻
[2] 筑波大学計算科学研究センター

1　はじめに

近年，知識グラフへの注目度が高まっている．知識グラフとは，世の中のエンティティ同士の関係を主語 (subject, head entity)，述語 (predicate, relation)，目的語 (object, tail) から成るトリプルの形式によって表したグラフ構造である．知識グラフの応用例は広く，例えば自然言語処理の分野では質問応答（QA）タスクの回答データベースや機械学習タスクの教師データとして，画像認識の分野では画像中の物体同士の関係理解，推薦システムの分野ではコールドスタート問題の対策や推薦結果への情報補完などにそれぞれ用いられている．

知識グラフは，上述したような利点を持つ一方，本質的に不完全であり，継続的なメンテナンスが不可欠という性質も抱えている．メンテナンスが必要な例としては，現実世界の事実情報に追随したリレーションの追加・削除・更新，新規エンティティの追加，不備によりリレーションが欠けてしまっているトリプルの補完などが挙げられる．このような問題へ対処するために様々な研究が行われている．リレーションの更新や，新規エンティティの追加に関する研究としては，Web 文書などのパブリックかつ非構造なデータから新たなトリプルの抽出を行う OpenIE[1] や，抽出したトリプルを知識グラフ中に取り込むために，エンティティの同一性の判定を行う Entity Recognition[2] などがある．また，欠けたトリプルの補完としては，トリプルのうちの 2 要素から，欠けた 1 要素の予測を行う，知識グラフ補完に関する研究が盛んに行われている [3, 4, 5].

このように，知識グラフのメンテナンスを自動で行うことを目的として，様々な分野の研究が行われている．しかし一方で，現状の知識グラフのメンテナンスは，依然として人間の手に大きく依存しており，前述した自動化のための手法も，多くは人手と組み合わせて用いられる．

人手と自動化手法によるメンテナンスの両方に共通する問題点として，膨大なエンティティを含む知識グラフの中から，メンテナンスを行うべき箇所を判定するのが難しいという点が考えられる．人手でのメンテナンス行う場合，対象のエンティティの特定は，各人が個人で得た情報に依存する．また，自動化手法を用いて非構造文書などよりトリプルを追加する場合も，追加する情報はソースのドメインに依存する．よっていずれの手法を用いる場合においても，知識グラフ中の更新を行うべき箇所を網羅できていないと言える．もし，知識グラフ中の更新すべきエンティティを自動で検出できれば，人手によるメンテナンスにおける個人の知識差による影響を減らすことができ，また自動化手法で見落とされた箇所などの補完も可能となる．

よって本研究では，知識グラフにおいて，情報を更新すべきエンティティを自動で検出する手法の開発を目的とする．今回は，あるエンティティへの編集が，そのエンティティと周辺エンティティへの更なる編集を誘発すると考え，エンティティのリンク関係及びそれぞれへの編集情報を元にした手法の提案を行った．また，提案手法の有効性を確認するための評価実験を行い，エンティティのリンク関係を用いたベースライン手法との性能を比較検討した．

2　関連研究

2.1　知識グラフの状態遷移判定

Wijaya ら [6] は，知識グラフ中のエンティティについて，それが新たな事実情報の追加などが起きる状態にあるかを Contextual Temporal Profiles（CTPs）を用いて判定する手法を提案した．CTPs とは，各年代ごとにおけるエンティティの特性を，その年代の文書コーパスから導出したものである．Wijaya らは，CTPs の設計のためにまず，各年代のコーパスから，事象の変化の元となるようなエンティティ（大統領など）の周辺に現れる語句を抽出し，そのエンティティのドメインの特性を表現するベクトルの生成を行った．次に，年代ごとに得られたドメインの特性ベクトルの差分を考えることで，ドメインの遷移状態を表すようなベクトルを得た．最後に，ドメインの新規エンティティについて，その特性ベクトルなどを生成し，得られたドメインの特性ベクトル及び，遷移状態ベクトルと比較することで，そのエンティティに関するトリプルが発生する年代の推定を行った．この研究では，知識グラ

フのエンティティの遷移状態を大規模コーパスを元にモデル化し，変化を予測することを可能としたが，ドメインを表すような seed エンティティの選定を人手で選択する必要がある点や，モデル化のための情報源として年代ごとのコーパスが必要である点で改善の余地がある．

2.2 知識グラフにおけるリンク予測

知識グラフ中のエンティティについてリンク予測を行うタスクは知識グラフ補完（Knowledge Graph Completion）と呼ばれ，盛んに研究が行われている．知識グラフ補完の代表的な手法として，Bordes らが提案した TransE [3] がある．TransE は，知識グラフ中のトリプル (h, r, t) をもとに，エンティティに対するベクトル $\boldsymbol{e}_i$，リレーションに対するベクトル $\boldsymbol{l}_j$ を，$\boldsymbol{e}_h + \boldsymbol{l}_r = \boldsymbol{e}_t$ の関係を満たすように学習する手法である．この手法を用いて，$(h, r, ?)$ のような一部の要素が欠けたトリプルについて，ベクトル $\boldsymbol{e}_h + \boldsymbol{l}_r$ と相関が高いベクトル $\boldsymbol{e}_t$ を求めることで，tail になり得るエンティティの候補を得ることができる．

TransE は精度面，応用面で様々な拡張がされている [4, 5]．その中でも Leblay ら [7] は，時間情報を考慮した知識グラフ補完のタスクに取り組んでいる．このタスクにおいては，通常のトリプルに時間情報 τ を追加したものを扱う．時間情報も考慮した埋め込み表現を求めることで，トリプルの発生する時間の予測 (（Ethiopia, Praise or endorse, China, ?）) や，ある時間が指定された場合のリンクの予測 (（Canada, Host a visit, ?, 2014-04-20）) が可能となる．

このように知識グラフにおけるリンク予測に関して様々な研究が行われている．しかし，いずれにおいても，トリプルのうちの一部が指定されることを前提としており，予測を行うべきエンティティの検出については考えられていない．

3 提案手法

本節では提案手法について説明する．手法の方針として，まず，知識グラフにおいて，どのようなエンティティに編集が行われるかについて考える．ある時刻 t とそこから α だけ経過した時刻 $t + \alpha$ を比較した時，新たな関係の追加が行われたエンティティ及びその周辺にあるエンティティは今後においても編集が起きやすいと想定される．例として，知識グラフ中のエンティティ *Joe Binden* に新たな関係，(*Joe Binden, position hold, President of U.S*) が追加されたケースを考える．この時，新たな関係が追加された *Joe Biden* は，大統領就任に伴い周辺に様々な変化が予想されるため，知識ベース上でも今後更新がされやすいと考えられる．また，その周辺にあるエンティティにも変化が伝播していくことが予想される．更に，編集の伝播は，時間の経過とともに減衰していくと考えられる．

次に，エンティティの編集を伝播しやすいようなリレーションについて考える．ある時刻 t においてエンティティに編集が行われた時，その編集は，過去により多くの編集を伝播したリレーションによって伝播されると考えられる．例えば，知識グラフ知識グラフ中のエンティティ *President of U.S* に *office holder* に関する新たな関係，(*President of U.S, office holder, Joe Biden*) が追加された時，*Joe Biden* には過去の例に則って職や住居，活動記録などさまざまな関係が追加されていく．一方，エンティティ *Joe Binden* に *handedness* に関する新たな関係, (*Joe Binden, handedness, right handedness*) が追加されたとしても，これは *right handedness* への編集を誘発しない．このような，過去に編集を伝播したかの履歴をもとに，リレーションへの重みを決定する．

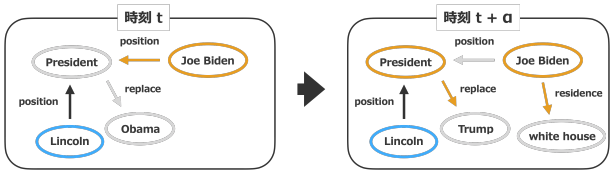

図 1: 提案手法の基本アイデア．ある時刻 t において編集が行われたエンティティは以降の時刻 $t + \alpha$ でも編集が行われやすく，そしてその編集は周辺のエンティティへも伝播すると考えられる．

これらの仮定を元に手法の設計を行う．今回は，前述した仮定を取り入れつつ，編集が起きたエンティティに類似したエンティティを見つけ出すために，Personalized PageRank（PPR）[8] を手法のベースとして用いることとした．PPR は，クエリノードを始点とするランダムウォークを行い，その際，一定確率 α でクエリノードのいずれかにその問い合わせ確率に応じてジャンプすることで，グラフ中のノード上でのランダムサーファーの存在確率を算出し，それをクエリノードとの類似度として用いる手法である．グラフのノード数を N，要素 i がノードの問合せ確率を表すクエリベクトルを $\boldsymbol{b} \in \mathbb{R}^N$，各列がノードの隣接ノードへの遷移確率を表すベクトルである隣接行列を $\boldsymbol{A} \in \mathbb{R}^{N \times N}$，最終的に得られるランダムサーファーの各ノード上での存在確率のベクトルを $\boldsymbol{v} \in \mathbb{N}$ とすると，定常状態における $\boldsymbol{v}$ は，以下の式を収束するまで再起的に計算することで得られる．

$$\boldsymbol{v} = (1 - \alpha)\boldsymbol{A}\boldsymbol{v} + \alpha\boldsymbol{b} \tag{1}$$

提案手法においては，前述した仮定を PPR に取り入れ，かつ，より推測がしづらい（定常的でない）エンティティへの編集を捉えられるよう，エンティティ

Vol.2021-DBS-173 No.9
Vol.2021-IFAT-144 No.9
2021/9/16

e_i への編集の非定常性のスコアを定義し，それをもとに隣接行列 $\boldsymbol{A}$ 及びクエリノード $\boldsymbol{b}$ にそれぞれ次のような変更，制約を加えることとする．

エンティティ編集の非定常性スコア: 知識グラフ中のエンティティへの編集の中でも，定常的に行われているものではなく，非定期（あるいは突発的）なイベント的に発生するものを捉えられるよう，エンティティ e_i に対する編集の非定常性を表現するスコア $escore(e_i)$ を定義する．今回は次の要素をもとにスコアを決定することとした．

1. **期間毎の編集数の比率**．ある時刻 t 以降に発生するエンティティへの編集を予測するとき，時刻 t の直近の任意の期間を h_{rec}，h_{rec} 以前の任意の期間を h_{prev} とする．エンティティに非定常な編集が起こっているかは，期間 h_{rec} と期間 h_{prev} それぞれにおいて発生した編集の数の比に反映されていると考えられる．一方，h_{rec} を t の直近 7 日間，h_{prev} を t の 7 日前〜14 日前のように，日時が近くかつ短い期間同士を比較した場合，同様のイベントに起因する編集が期間を跨いで含まれてしまうケースが発生し，編集の非定常性が比率にうまく反映されないことが起こり得る．よって今回は，期間 h_{rec} を t の直近 7 日間，期間 h_{prev} を t の前月とし，比率の算出を行うこととした．

2. **エンティティへのリンク数**．リンク数が多いエンティティの場合，編集の対象が多いことから，編集数もそれに応じて増加すると考えられる．よって，リンク数を用いて前述したエンティティへの編集数の比率を標準化することとした．

上記をもとに，(2) 式のようにエンティティへの編集のスコアを定義した．ここで，$D_w(e_i)$ は基準日の直近 7 日間におけるエンティティ e_i の編集回数，$D_{lm}(e_i)$ は基準日の前月におけるエンティティ e_i の総編集回数，N_{e_i} はエンティティ e_i の外向きのリンク数である．

$$escore(e_i) = \frac{D_w(e_i)}{N_{e_i} D_{lm}(e_i)} \tag{2}$$

隣接行列 $\boldsymbol{A}$: 知識グラフ中のリレーションに対して，関連するエンティティが編集されるケースの総数などから，リレーションの重みを計算する．エンティティ e_i にリンクされたリレーションの集合を R_i，編集が起きたエンティティの集合を E_{edited}，エンティティ e_i にリレーション r_j でリンクされており，かつ編集が起きたエンティティの集合を E_{e_i,r_j} とし，リレーション r_j の重み W_{e_i,r_j} を以下のように算出する．

$$W_{(e_i,r_j)} = \frac{rscore(r_j)}{\sum_{r_k \in R_{e_i}} rscore(r_k)} + \min_{rscore(r)>0} rscore(r) \tag{3}$$

$$rscore(r_j) = \frac{1}{|E_{e_h,r_j}|} \sum_{e_i \in E_{edited}} \sum_{e_k \in E_{e_i,r_j}} escore(e_k) \tag{4}$$

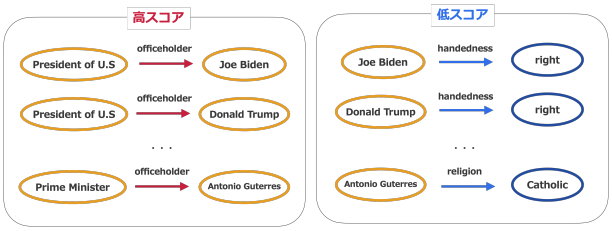

図 2: 隣接行列設計のための基本アイデア．過去により多くの編集を伝播しているリレーションで繋がっている箇所に，大きなスコアを与える．

クエリベクトル $\boldsymbol{b}$: 知識グラフ中の編集が起きたエンティティについて，行われた編集の数が多いかつ，直近で編集が行われたものほど，高い問い合わせ確率を与えるように設計を行う．全エンティティの集合を E，エンティティ e_i に対応するクエリベクトル $\boldsymbol{b}$ の要素を b_i，期間内でのエンティティ e_i の編集履歴の集合を H_{e_i}，編集 d の発生時刻（秒）を $t(d)$，基準時刻（秒）を t_{base} とすると，(5) 式より表される．

$$b_{e_i} = \frac{tscore(e_i)}{\sum_{e_j \in E} tscore(e_j)} \tag{5}$$

$$tscore(e_i) = \sum_{d \in D_{e_i}} \frac{1}{t(d) - t_{base}} \tag{6}$$

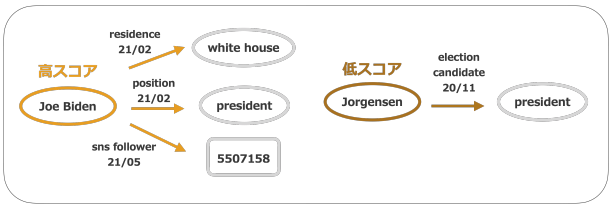

図 3: クエリベクトル設計のための基本アイデア．より直近に，多くの編集が行われたエンティティに高い値を与える．

4 評価実験

本節では，提案手法の有用性を確認するために行った，単純なアイデアに基づいた手法との比較実験について示す．

4.1 比較手法

提案手法との比較に用いる比較手法について説明する．今回は，リンクの数が多いならば編集される可能性が高い，直近で編集が行われたものは次の期間でも編集が行われる，という単純なアイデアに基づき，1. 各エンティティの持つリンクの数をそのエンティティの重要度として扱う手法，2. 知識グラフを有向グラフと見なし PageRank を適用する手法，3. 直近 7 日間で編集が行われたものを編集が起こると予測する手法，の 3 つをベースライン手法として採用した．1,2 の手法においては，得られる各エンティティの重要度スコアが，そのエンティティの編集されやさと比例すると想定する．ある重要度以上のエンティティは編集が起きるという閾値を求めることで，編集されるエンティティの予測を行う．

エンティティのリンク数を重要度とする手法: この手法においては，知識グラフ中の各エンティティについて，外向きにリンクされているリレーションの総数をそのエンティティの重要度として扱う．例として，知識グラフ内に，エンティティ *White house* に関する 2 つのトリプル *(White house, owned by, U.S.)* と *(Donald Trump, residence, White house)* がある場合について考える．この場合，エンティティ *White house* は 1 つの外向きのリレーションとリンクされているので，重要度は 1 となる．

PageRank: PargeRank[9] はグラフの中のノードの重要度を決定するアルゴリズムである．PageRank は，多くのノードにリンクされているノードからリンクされているノードほど重要度が高いというアイデアを基としてもつ．グラフ中のノード A に対するページランク値 $PR(A)$ は，ノード A にリンクされているノードを $T_i(1 \sim n)$，ノード Ti にリンクされているノードの総数を $C(T_i)$，ハイパラーパラメータを d とすると，以下の式より表される．

$$PR(A) = (1-d) + d\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)} \tag{7}$$

直近の編集有無を利用する手法: この手法においては，例として 2021/05/25〜2021/05/31 における編集の予測を行うとすると，2021/05/25 の直近 7 日間である 2021/05/17〜2021/05/24 において編集が行われたものを，同様に編集が起きると予測する．

4.2 データセット

実験用データセットとしては，公的に公開されている共同編集型大規模知識グラフの一つである wikidata[10] より次の手順にて構築したものを用いた．1. wikidata より，20210524，20210531，それぞれの dump を取得する．2. 20210524 中のグラフより，データ中の先頭 100 個の head エンティティを取り出す．3. 得られた 100 個のエンティティの 4 ホップ先までに出てきたエンティティと，それを head として持つトリプルをデータセットとして利用する．4. 最終的に得られた全てのエンティティの編集履歴を 50 件ずつ wikidata の web ページよりスクレイピングし取得する．

この手順の結果として得られたデータセットの内訳は表 1 の通りである．

表 1: Wikidata より構築したデータセットの内訳

	2021/05/24	2021/05/31
トリプル数	8,487,493	8,495,281
編集が起きたエンティティ数	62751 (5/25〜5/31)	48379 (6/1〜6/7)
エンティティ総数	995,012	
リレーション総数	5,782	

4.3 評価指標

評価指標としては以下を用いる．

- 編集されると予測したエンティティ e_i の $escore(e_i)$ の平均値．
- 手法によって算出したエンティティの重要度が上位 100 件のうち，実際に編集が行われたものの割合．
- 手法によって算出した重要度が閾値を超えたエンティティを編集が起きると予測する．
  - 予測結果において，test データ中で新たなリンクが追加されたエンティティのうち何割を検出できているかを Recall として評価する．
  - 予測結果において新たなリンクが追加されると判定されたエンティティのうち，何割が実際に test データ中でリンクが追加されたかを Precision として評価する．

4.4 実験手順

実験は次の手順で行った．

1. 2021/05/24 のグラフに対して，2021/05/25 ～ 2021/05/31 の間で編集が起きるかを予測し，それをもとに $escore$ の平均値，Recall，Precision それぞれが最大に成るように，ハイパーパラメータ最適化フレームワークである Optuna[11] の多目的最適化機能を用いて，閾値のパレート解を得る（提案手法では，クエリベクトル及び隣接行列の設計には 2021/05/17 ～ 2021/05/24 の編集履歴を用いる）．
2. 2021/05/31 のグラフに対して，得られた閾値のうち，平均 score を元に，2021/06/01 ～ 2021/06/07 の間で編集が行われるエンティティを予測し，性能を評価する（提案手法では，クエリベクトル及び隣接行列の設計には 2021/05/24 ～ 2021/05/31 の編集履歴を用いる）．

4.5 実験結果

上記の設定で行った実験の結果を表 2，3 に示す．

表 2: 実験結果（Recall が最大となる閾値の場合）

	Top 100	Precision	Recall	平均 escore*
リンク数	19 個	54.2%	71.6%	0.00217
PageRank	9 個	95.9%	0.529%	0.00798
編集有無	46 個	89.9%	16.5%	0.00173
提案手法	13 個	95.8%	0.679%	0.00874

* 全編集を正確に予測した際の平均 escore は 0.0925 となる．

表 3: 実験結果（平均 escore が最大となる閾値の場合）

	Top 100	Precision	Recall	平均 escore*
リンク数	19 個	88.3%	29.2%	0.00445
PageRank	9 個	96.0%	0.0438%	0.0196
編集有無	46 個	89.9%	16.5%	0.00173
提案手法	13 個	96.0%	0.821%	0.0318

* 全編集を正確に予測した際の平均 escore は 0.0925 となる．

表 2 の結果を見ると，リンク数を用いる手法の Recall が最も高くなっていることから，現時点で，ある程度の精度で多くの編集が起きるエンティティを予測できているという観点では，リンク数を利用するのが妥当だと考えられる．一方，表 3 をもとに，平均 escore に注目して比較すると， PageRank 及び提案手法が，リンク数を用いる手法に大きく優っていることがわかる．

次に，どれだけ非定常的な編集を捉えつつ，多くのエンティティを検出できているかを見るため，Recall に対する平均 escore の変化を見た．リンク数を用いる手法の結果を図 4 に，PageRank 及び提案手法の結果を図 5 に示す．

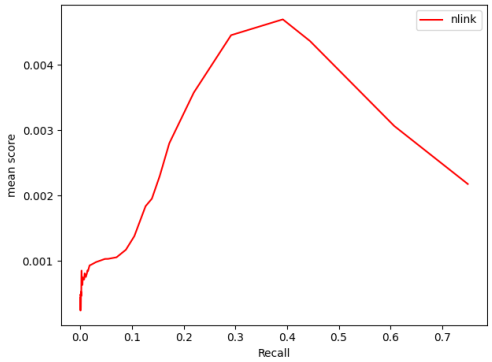

図 4: リンク数を用いる手法における Recall と平均 escore の関係

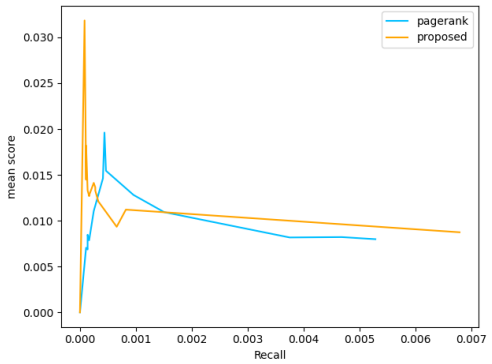

図 5: PageRank 及び提案手法における Recall と平均 escore の関係

図 4，5 のグラフにおいてまず，リンク数を用いる手法及び PageRank，提案手法を比較すると．リンク数を用いる手法の場合は，一貫して Recall が高いものの，平均 escore の観点では PageRank 及び提案手法を用いる方が高いことがわかる．よって，より多くの編集が起きるエンティティを捉えたい場合にはリンク数を用いる手法を，より非定常的な編集を捉えたい場合には PageRank もしくは提案手法を用いるなどの使い分けができると言える．

次に，図 5 より，PageRank 及び提案手法を比較する．最大平均 escore の観点では前述したのと同様，提案手法が大きく優っているものの，全体に注目すると大きな差は見られない．また，Recall が 0.001 未満の非常に小さい段階で平均 escore が大きくなっていることから，より非定常的な編集が起きるエンティティに，依然として小さいスコア付けがされてしまっていることがわかる．よって，よりエンティティの編集に関連

するような性質を特定し，スコアに反映できるよう手法の改善を行う必要がある．

5 まとめ

本研究では，知識グラフにおける新たなリンクの追加や属性値の修正などの編集が行われるエンティティを検出することを目的として，手法の提案及び，単純なアイデアを基にしたベースライン手法との比較実験を行った．実験の結果，編集の起きるエンティティの検出数という観点ではリンク数を用いる手法の，より非定常的な編集を検知できるかという観点では PageRank 及び提案手法が優っており，また， PageRank 及びそれを拡張した提案手法は全体的に大きな差は見られないという結果となった．このような結果となった主な原因としては，編集の発生に大きく関わる要素を未だ手法に取り込めていないことが考えられる．今回は，あるエンティティの編集がリンク先のエンティティの編集を誘発するという仮定のもと，手法の設計を行ったが，より編集の発生に関わる要素の特定が必要である．今後は，提案手法について，それらの問題点への対策及び性能向上に取り組むとともに．実際の検出結果などを見ることで，手法が想定通りに伝播を捉えるような動きをしているかなどの解析も行っていく予定である．

参考文献

[1] C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh, “A survey on open information extraction,” in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 3866–3878, Association for Computational Linguistics, Aug. 2018.

[2] V. Yadav and S. Bethard, “A survey on recent advances in named entity recognition from deep learning models,” in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 2145–2158, Association for Computational Linguistics, Aug. 2018.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Advances in Neural Information Processing Systems* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), vol. 26, pp. 2787–2795, Curran Associates, Inc., 2013.

[4] B. Shi and T. Weninger, “Proje: Embedding projection for knowledge graph completion,” *CoRR*, vol. abs/1611.05425, 2016.

[5] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation learning of knowledge graphs with entity descriptions,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI’16, p. 2659–2665, AAAI Press, 2016.

[6] D. T. Wijaya, N. Nakashole, and T. M. Mitchell, “CTPs: Contextual temporal profiles for time scoping facts using state change detection,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1930–1936, Association for Computational Linguistics, Oct. 2014.

[7] J. Leblay and M. W. Chekol, “Deriving validity time in knowledge graph,” in *Companion Proceedings of the The Web Conference 2018*, WWW ’18, (Republic and Canton of Geneva, CHE), p. 1771–1776, International World Wide Web Conferences Steering Committee, 2018.

[8] G. Jeh and J. Widom, “Scaling personalized web search,” in *Proceedings of the 12th International Conference on World Wide Web*, WWW ’03, (New York, NY, USA), p. 271–279, Association for Computing Machinery, 2003.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web,” in *Proceedings of the 7th International World Wide Web Conference*, (Brisbane, Australia), pp. 161–172, 1998.

[10] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” *CoRR*, vol. abs/1907.10902, 2019.