

第2回ナレッジグラフ推論チャレンジ2019 受賞作品

ナレッジグラフの埋め込み手法

2020年9月9日
株式会社KDDI総合研究所
黒川 茂莉

■ ナレッジグラフに含まれる知識を使った**帰納的学習（機械学習）**を行うことにより、ナレッジグラフにない知識を予測する方法を解説します

- 悪魔の足では、mortimer を *kill* したのは誰か？はナレッジグラフにはない
- まだらの紐では、julia を *kill* したのは誰か？はナレッジグラフにはない

■ 特に、ナレッジグラフの要素を連続空間に**埋め込み**、埋め込み空間上での演算により推論を行う方法を紹介します

- mortimer や julia や holmes や watson や *kill* を埋め込む

■ ナレッジグラフ埋め込みについて

■ ナレッジグラフ推論チャレンジ2019でのアプローチ

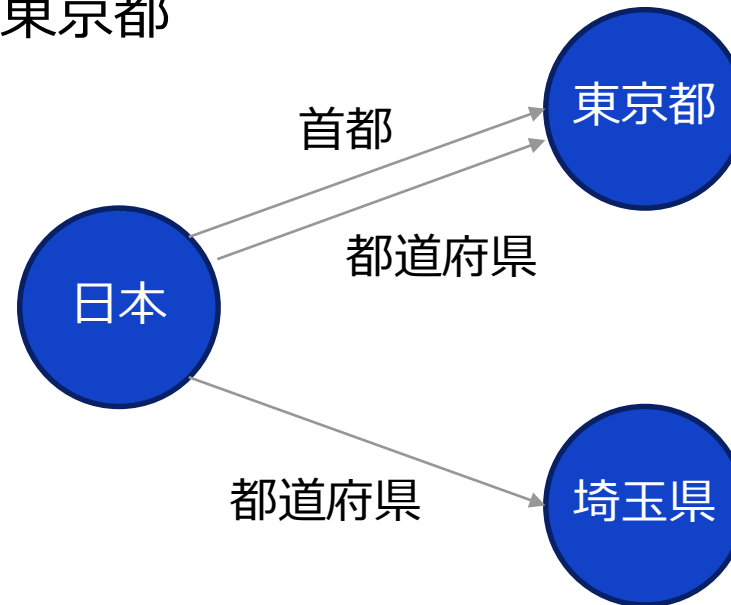
■ ナレッジグラフ埋め込みについて

■ ナレッジグラフ推論チャレンジ2019でのアプローチ

ナレッジグラフとは

■ ナレッジグラフの定義 $KG=\{E,R,T\}$

- E =エンティティ(実体)の集合 例：{日本, 東京都, 埼玉県}
- R =リレーション(関係)の集合 例：{首都, 都道府県}
- T =三つ組の集合 例：{(日本,首都,東京都), (日本,都道府県,埼玉県), ... }
 - ・ 三つ組は(ヘッド,関係,テイル)から構成される。(日本,首都,東京都)の場合、ヘッド=日本,関係=首都,テイル=東京都



※ナレッジグラフ埋め込みの領域でよくある定義です。
(知識工学では違う定義があるかもしれません。)

■ ナレッジグラフは人間の英知と労力の結集

- 多数のオープンなナレッジグラフ

例：ナレッジグラフ推論チャレンジのナレッジグラフ、DBPedia、FreeBase、YAGO、他多数

■ しかしながら、完全なものを作るのは困難

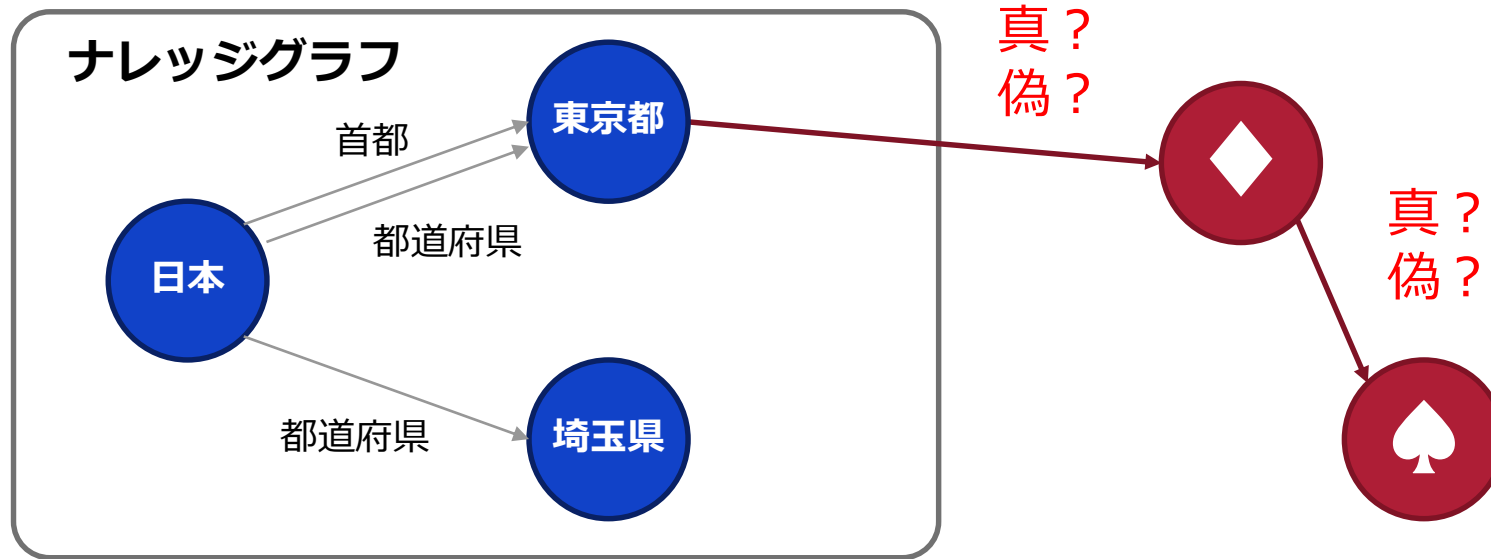
- 欠損している情報があるかもしれない（忘れられていたり、知られていなかったり、暗黙知のため表現されていなかったり）
- また、人間なので間違いもあるかもしれない

■ ナレッジグラフを完全なものに近づけるため、それまでに表現された三つ組を使って、ナレッジグラフに欠損している三つ組を補完（＝ナレッジグラフ補完）するのが重要

ナレッジグラフ補完の前提

ナレッジグラフ外にも真の知識があるか？

- 閉世界仮説（CWA）：ナレッジグラフ外の知識は偽
 - 開世界仮説（OWA）：ナレッジグラフ外にも真の知識がある
- ➡ナレッジグラフが不完全であることを想定すると、開世界仮説のほうが自然？



※補足：ナレッジグラフの一部知識は間違いかもしれないという前提はあまり考えられていないが、今後は重要かもしれない

ナレッジグラフ補完における正例と負例

ナレッジグラフ内の知識は正例のみ→負例をどこからか持ってくる必要がある

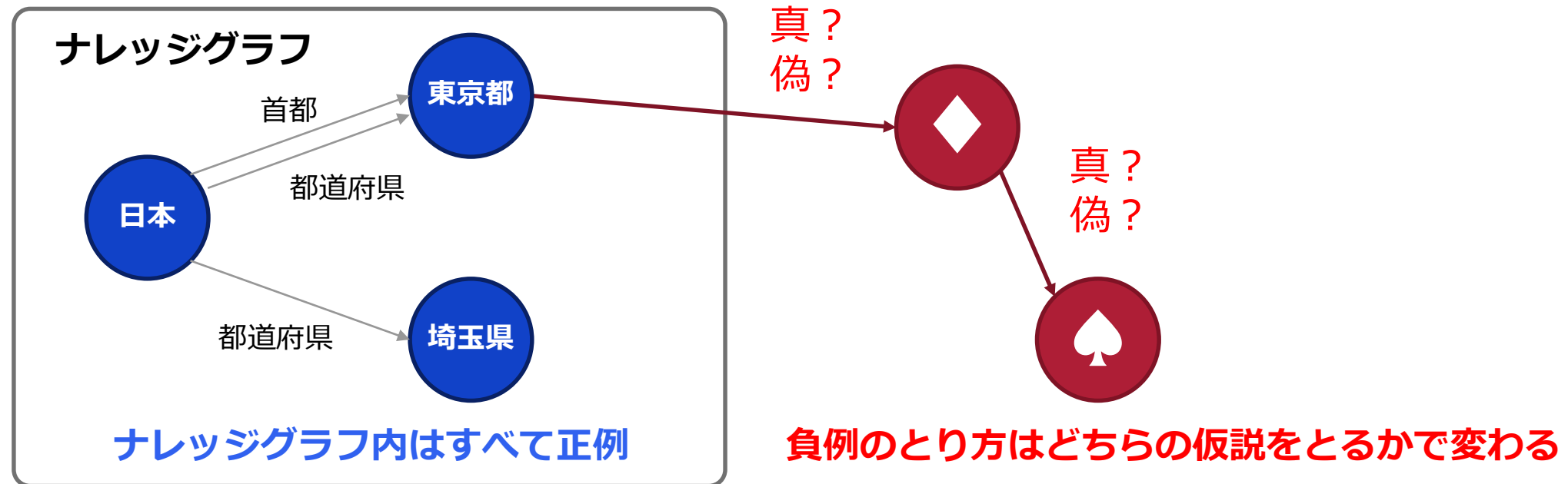
■ 閉世界仮説（CWA）：ナレッジグラフ外の知識は偽

→ナレッジグラフ外の知識は何でも負例

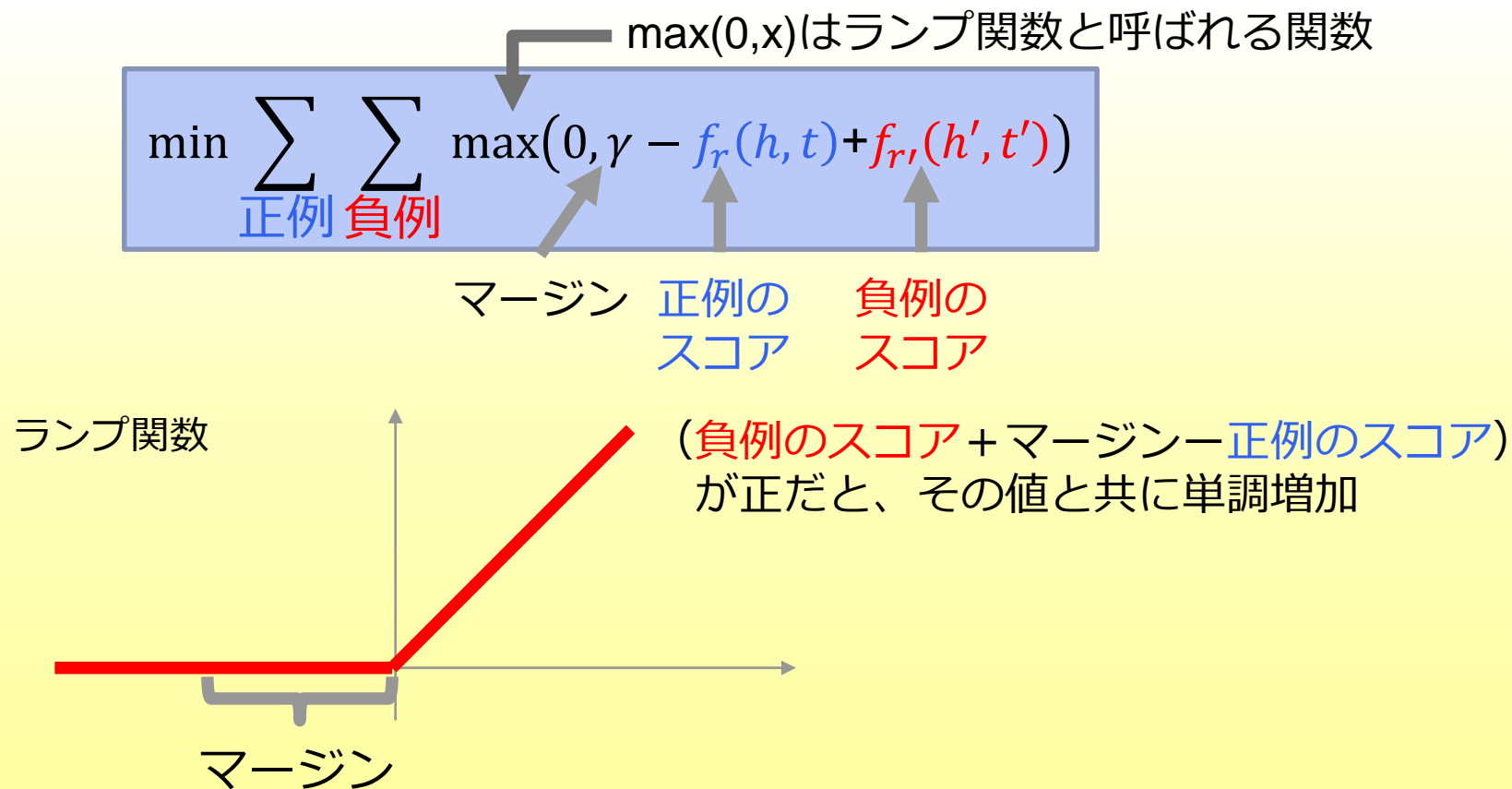
■ 開世界仮説（OWA）：ナレッジグラフ外にも真の知識がある

→ヒューリスティクスが必要。例えば、**正例の一部を変えて負例を作成**

（正例の周辺はきちんと表現されていて、正例を少し変えたものは負例であるという仮定）



■ 開世界仮説でのよくあるロス関数 = Pairwise ranking loss (Margin-based)



要は、(負例のスコア + マージン > 正例のスコア) となったらペナルティ

他には、Log-softmax lossもよくつかわれる

ナレッジグラフ埋め込みによる補完

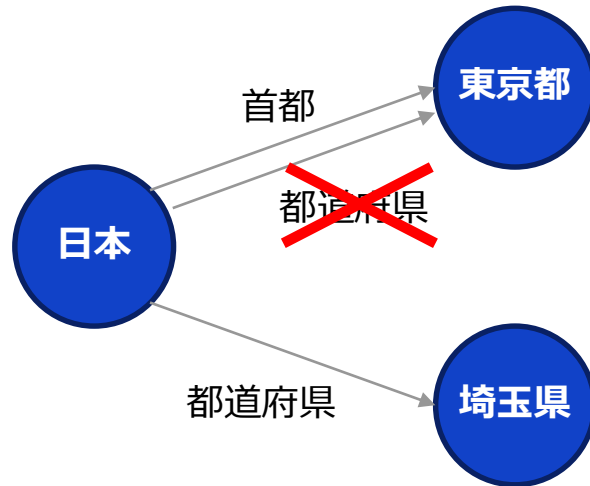
- 連続空間 (i.e. 埋め込み空間) に「何か」を満たすように実体、関係を射影し、その空間の上でベクトル演算により欠損しているリンクを予測

仮に(日本,都道府県,東京都)という三つ組が欠損していたとする

リンク予測: (日本,?,東京都)

$\vec{\text{日本}} - \vec{\text{東京都}}$ の近傍のベクトルは? \rightarrow $\vec{\text{都道府県}}$

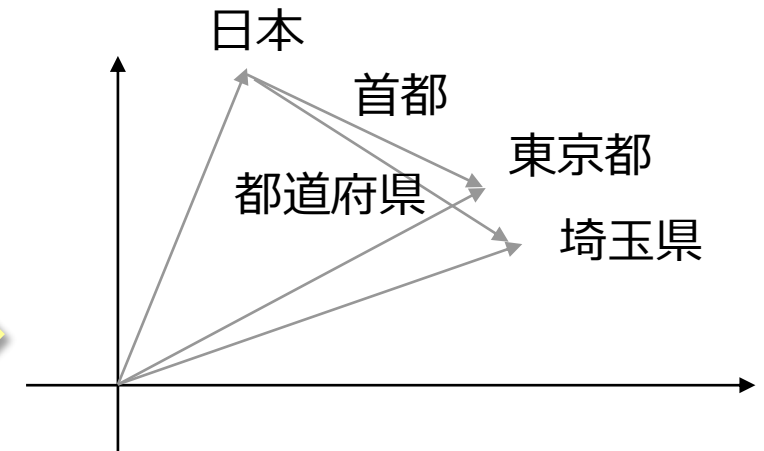
ナレッジグラフ



TransE[1]の場合

$\vec{\text{ヘッド}} + \vec{\text{関係}} \sim \vec{\text{テイル}}$

(例: $\vec{\text{日本}} + \vec{\text{首都}} \sim \vec{\text{東京都}}$)
が成り立つように埋め込み

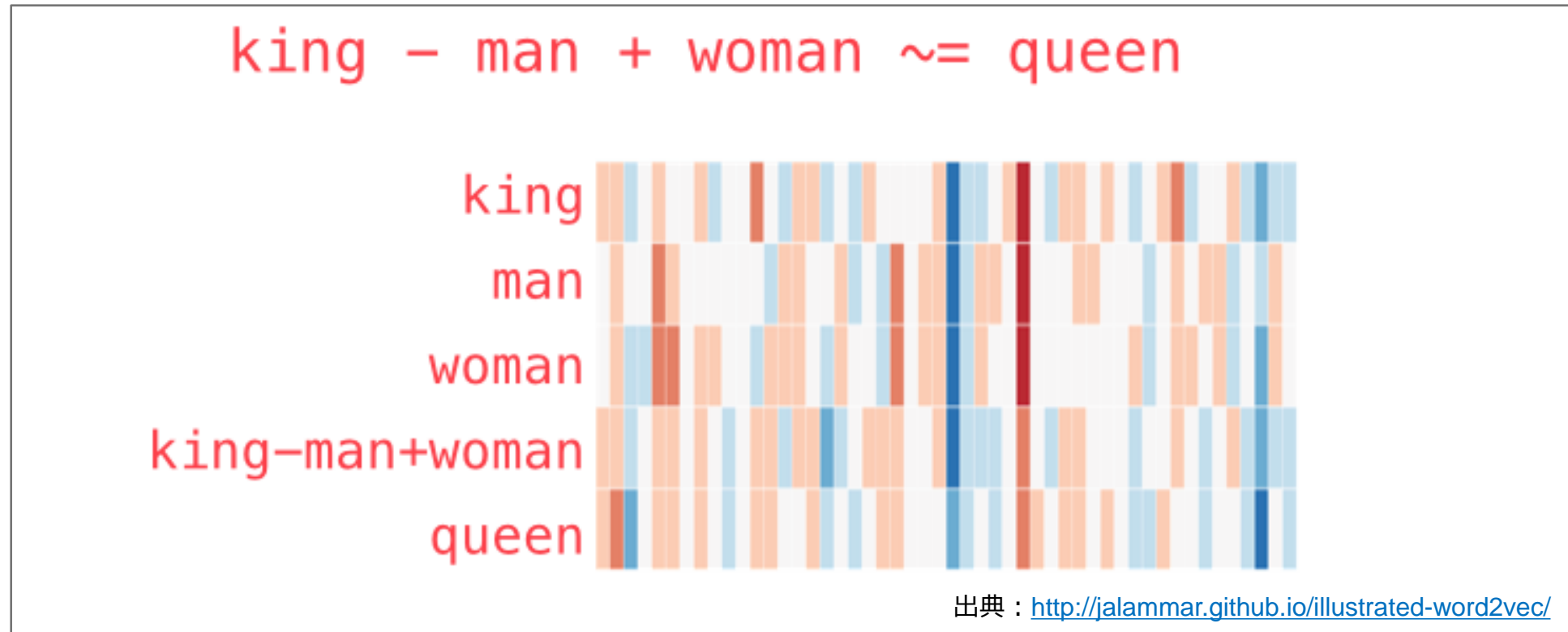


補足: $=$ ではなく \sim : 厳密に $=$ にしてしまうとどこかで破綻してしまう

[1] Bordes, A. et al., NeurIPS2013.

■ 語義の獲得（文脈の考慮の効果）

- 文脈を考慮することで、共起しやすい語との関連付けを学習できる



Word2Vecは、文脈を考慮することで結果的にこのような関係を学習したが、TransEは、直接的に $\overrightarrow{\text{ヘッド}} + \overrightarrow{\text{関係}} \sim \overrightarrow{\text{テイル}}$ のような関係を学習させる

- スコアリング関数...三つ組の関係をスコアリングする関数（沢山提案がある）
- 制約...無制約にすると原点につぶれたり発散したりしやすい⇒単位超球面に制約することが多い

表は[1]より引用

TABLE 1
Summary of Translational Distance Models (See Section 3.1 for Details)

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h, t)$	Constraints/Regularization
TransE [14]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\ \mathbf{h} + \mathbf{r} - \mathbf{t}\ _{1/2}$	$\ \mathbf{h}\ _2 = 1, \ \mathbf{t}\ _2 = 1$
TransH [15]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{w}_r \in \mathbb{R}^d$	$-\ (\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r) + \mathbf{r} - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r)\ _2^2$	$\ \mathbf{h}\ _2 \leq 1, \ \mathbf{t}\ _2 \leq 1$
TransR [16]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k \times d}$	$-\ \mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t}\ _2^2$	$ \mathbf{w}_r^\top \mathbf{r} / \ \mathbf{r}\ _2 \leq \epsilon, \ \mathbf{w}_r\ _2 = 1$ $\ \mathbf{h}\ _2 \leq 1, \ \mathbf{t}\ _2 \leq 1, \ \mathbf{r}\ _2 \leq 1$

スコアリング関数

制約



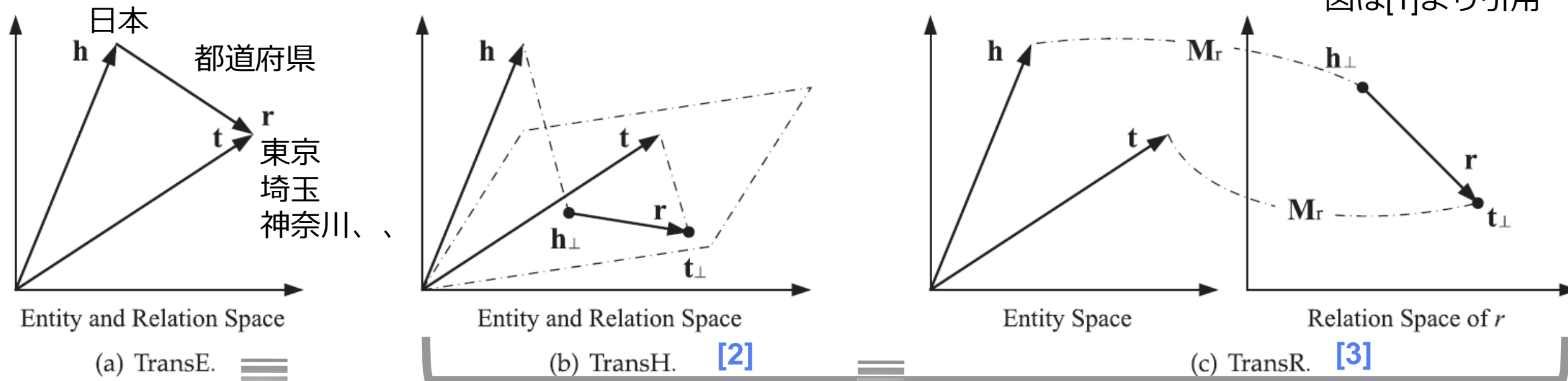
具体的なスコアリング関数の考え方は次頁

手法は山のようにあるので、考え方を知る & いろいろ試すのが重要

[1] Wang, Q. et al., IEEE Trans. KDE2017.

- 関係ごとに部分空間を分ける (TransH) や関係ごとの空間に射影する (TransR) により、関係ごとの違いを表現できるようにする

図は[1]より引用



☹️ 一对多がうまく表せないよ

(日本の都道府県は東京、埼玉、神奈川、他にもいっぱいあるが1点につぶれやすくなってしまう)

😊 一对多がうまく表せそう

(都道府県の空間では東京、埼玉、神奈川、他は1点につぶれてもいいけど、他の関係の空間ではつぶれないようにできる)

[1] Wang, Q. et al., IEEE Trans. KDE2017. [2] Wang, Z. et al., AAAI2014. [3] Lin, Y. et al., AAAI2015.

単一ナレッジグラフ埋め込みのバリエーション

- 😞 1ホップの関係しか考慮できない

➡➡➡ 😊 関係のパスを考慮

- PTransE[1]

- 😞 簡単な負例だと学習の効率が悪い

➡➡➡ 😊 GANの仕組みを使って負例を改善

- KBGAN[2]

- 😞 制約が強過ぎると関係の整合性が崩れる

➡➡➡ 😊 制約をトラスに緩和

- TorusE[3]

- 以上のような着眼点の中に、ナレッジグラフ埋め込みを使う手法選択のヒントがあるかもしれない

[1] Lin, Y. et al., EMNLP2015. [2] Cai, L. and Wang W.Y., NAACL-HLT 2018. [3] Ebisu, T. and Ichise, R., AAAI2018.

複数ナレッジグラフの埋め込み

- 複数ナレッジグラフ間で共通項（実体ペア）がある前提で埋め込み
 - 共通項の例：日本=Japan（日本語と英語のナレッジグラフの場合）
- タスク：欠損しているリンクを予測する、または、欠損している共通項を予測する

- よくあるテクニック
 - パラメータ変換...共通項の埋め込み間で線形変換を仮定
 - パラメータシェア...共通項の埋め込みを共通化
 - パラメータスワップ...三つ組のうち共通項の一方を他方で置換
 - ソフトアラインメント...共通項らしいとモデルが判断した実体ペアを擬似的に共通項とする

有名な手法	主な特徴	三つ組ロス関数	負例サンプリング	アラインメントロス関数	ソフトアラインメント
MTransE[1]	パラメータ変換	TransE(負例なし)と同じ	なし	変換ロス(ユークリッド)	無し
I(P)TransE[2]	パラメータシェア +パラメータスワップ	(P)TransEと同じ	ランダム	0（パラメータシェアなのでロスを測れない）	有り
BootEA[3]	パラメータスワップ	Limit-based loss	独自（ ϵ -truncated）	半教師あり学習の尤度	有り

[1] Chen, M. et al., IJCAI2017. [2] Zhu, H. et al., IJCAI2017. [3] Sun, Z. et al., IJCAI2018.

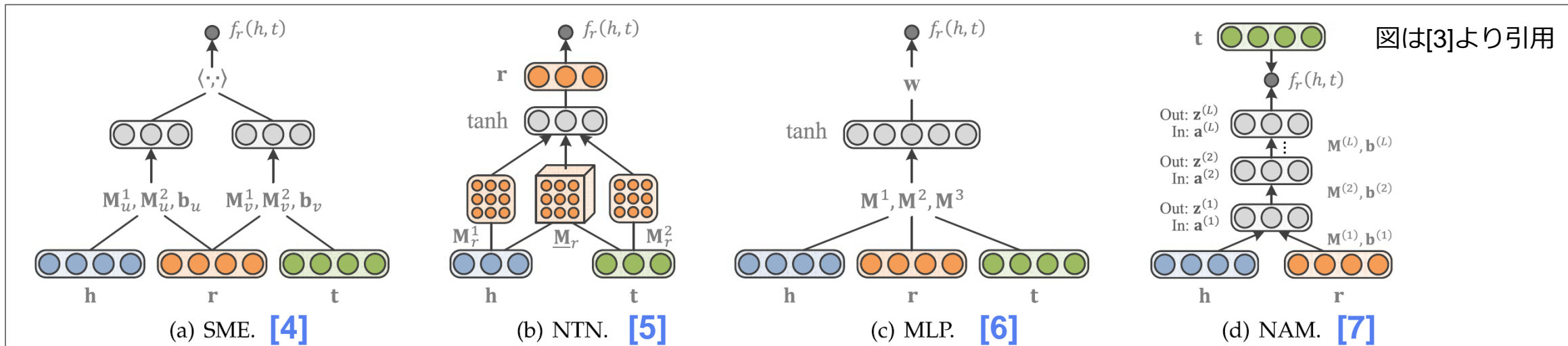
■ [単一ナレッジグラフ] 双線形モデル (RESCAL[1], DistMult[2] 等)

- ヘッドとテイルの埋め込みの間に関係ごとのマッチング行列を仮定

$$f_r(h, t) = \vec{e}_h^t W_r \vec{e}_t \quad \text{線形モデルにしてはパラメータ数が多く、過学習してしまう場合がある[3]} \\ \rightarrow \text{DistMult[2]等の簡略化}$$

- c.f. TranE等は平行移動 (Translation) モデルと呼ばれる

■ [単一ナレッジグラフ] ニューラルネットワーク化



■ [複数ナレッジグラフ] グラフ深層学習の応用 (Attention-based GCN[8], Highway GCN[9]等)

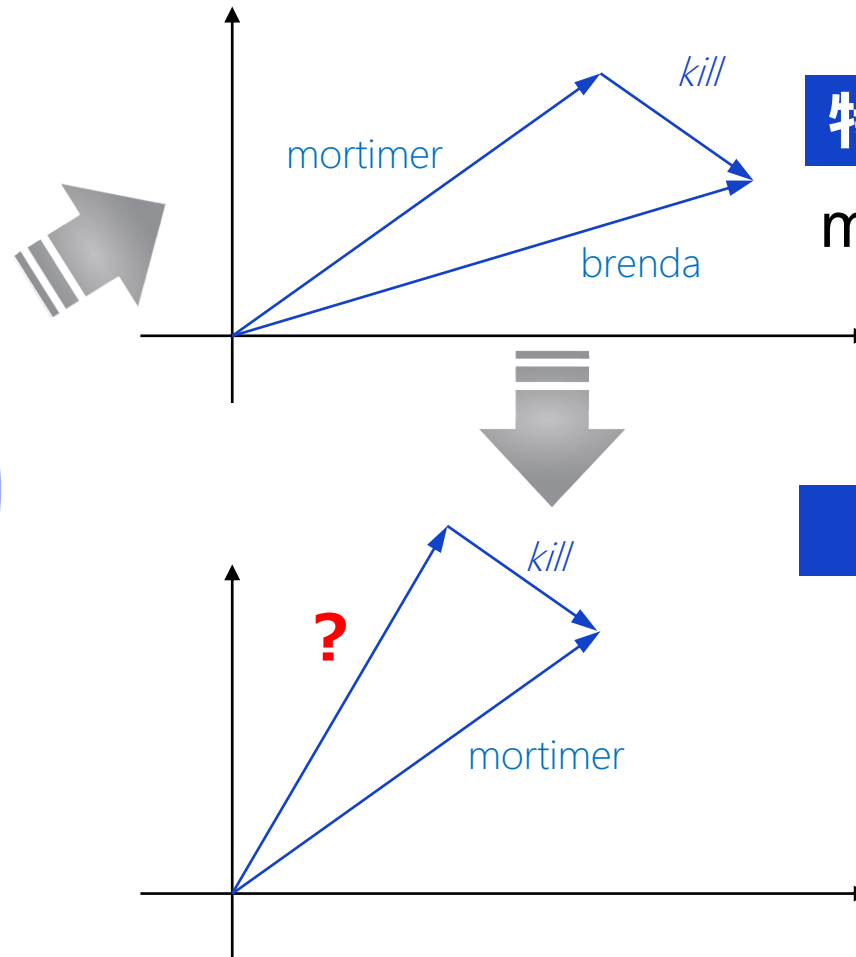
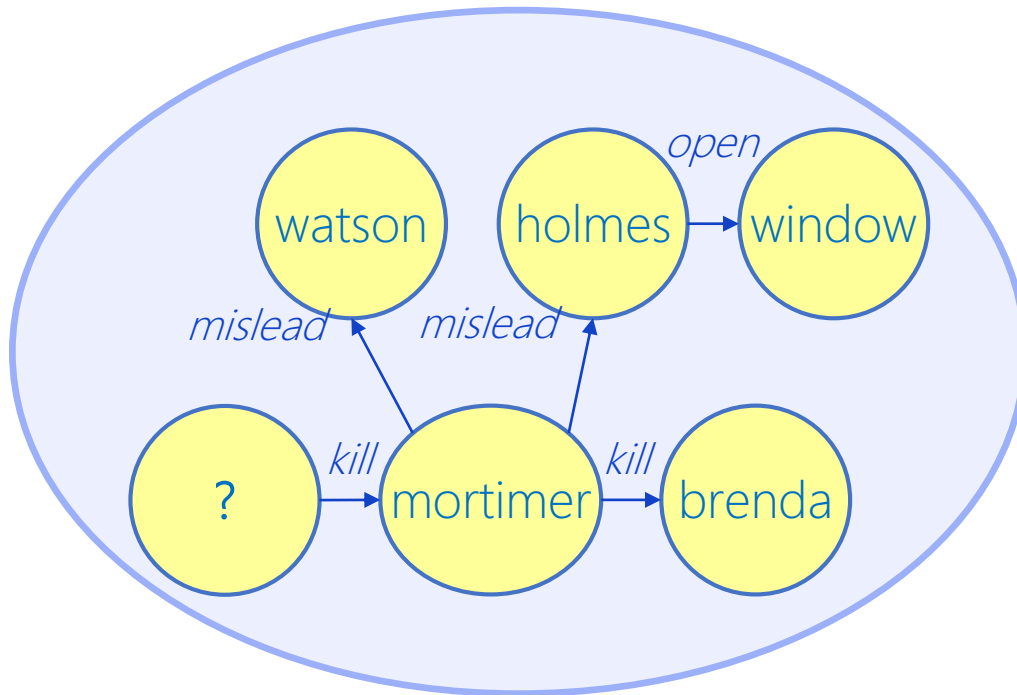
[1] Nickel, M. et al., ICML2011. [2] Yang, B. et al., ICLR2015. [3] Wang, Q. et al., IEEE Trans. KDE2017. [4] Bordes, A. et al., Machine Learning 2014. [5] Socher, R. et al., NeurIPS2013. [6] Dong X.L., et al., SIGKDD2014. [7] Liu, Q. et al., arXiv:1603.07704. [8] Xu, K. et al., ACL2019. [9] Wu, Y. et al., EMNLP2019.

■ ナレッジグラフ埋め込みについて

■ ナレッジグラフ推論チャレンジ2019でのアプローチ

- ナレッジグラフの要素（ホームズ、ワトソン、被害者、凶器、犯行場所等）を特徴空間に埋め込み、特徴空間上のベクトル演算に基づいて犯人を推論する

▼ 悪魔の足ナレッジグラフ



特徴空間に埋め込み

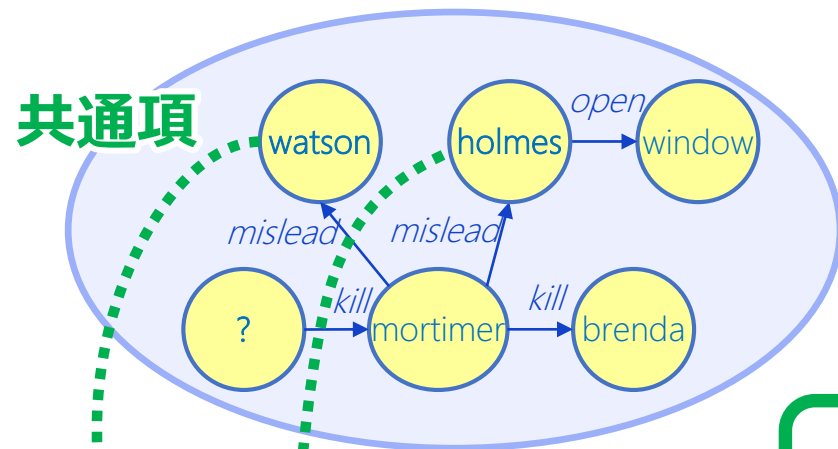
$\text{mortimer} + \text{kill} \sim \text{brenda}$

犯人の推論

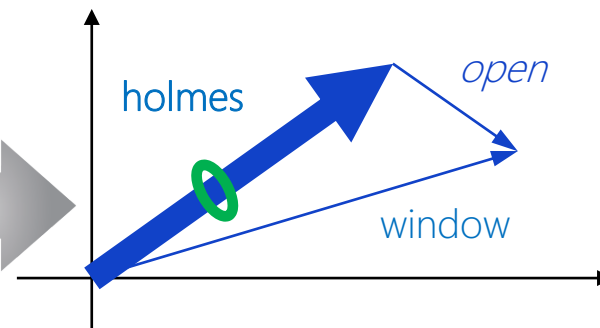
$? + \text{kill} \sim \text{mortimer}$

■ 複数ナレッジグラフに拡張

▼ 悪魔の足ナレッジグラフ



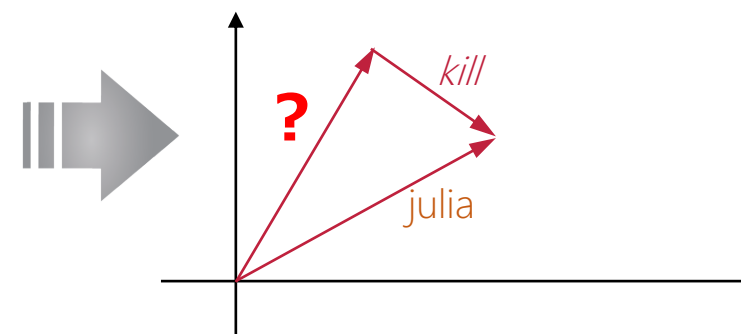
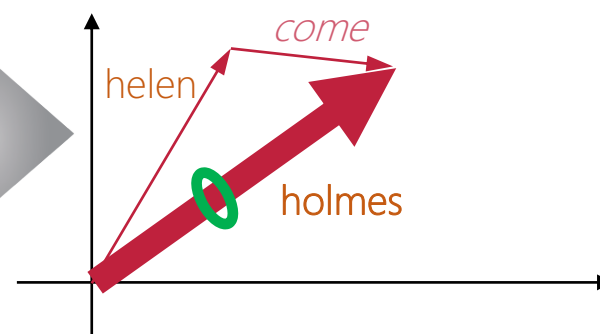
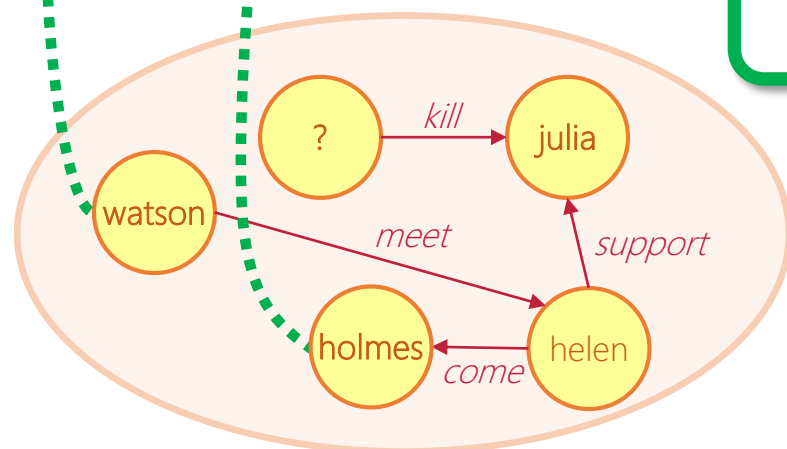
特徴空間に埋め込み



共通項(holmes, watson)について
パラメータシェア

犯人の推論

?+kill~julia



▲ まだらの紐ナレッジグラフ

ポイント1: 被害者の近傍探索（ナレッジグラフの要素を埋め込んだ特徴空間上で）

- ナレッジグラフの要素を特徴空間に埋め込むことで、特徴空間上の被害者の近傍探索を行うことにより関係する人物を特徴空間から探することが可能になる

ポイント2: *kill* 関係を考慮した被害者の近傍探索

- 悪魔の足ナレッジグラフに「mortimer *kill* brenda」や「brenda *kill*(ed) case_1」の関係が存在するため、特徴空間上で*kill*を表現でき、**?**+*kill*~mortimerという推論（=mortimer-*kill*の近傍探索）が可能になる

ポイント3: 複数ナレッジグラフの援用

- 複数ナレッジグラフの共通項であるholmes, watsonを用いて特徴空間を関連付けることで、他のナレッジグラフから特徴空間で知識が流入させ、推論の精度を向上させる

■ 処理フロー

① ナレッジグラフからの三つ組抽出と整形



② 三つ組の各要素の特徴空間への埋め込み



③ 特徴ベクトル演算に基づく犯人の推論

- 各ナレッジグラフにSPARQLで問い合わせを行い「ヘッド, リレーション, テイル」の三つ組を抽出
 - 「第1回ナレッジグラフ推論チャレンジ」における野村総合研究所チームの手法と同様＊次ページ
- ヘッド, リレーション, テイルの名称を全て小文字に変換（例：Holmes→holmes）
- ヘッドとテイルに含まれるエンティティの名称の先頭に小説タイトルを追記（例：standale→DevilsFoot_standale）
 - エンティティは、同じ名称でも小説毎に異なる意味を持つ想定
ただし、holmes、watsonは小説間で共有
- リレーションの名称はレンマ化をして動詞の語形変化を除去（例：guessed→guess）
 - レンマ化にはNatural Language Toolkt (<https://www.nltk.org/>)のWordNetLemmatizerを利用
 - リレーションは、同じ名称であれば小説が異なっても同じ意味を持つと想定し、小説間で共有
- 「brenda killed case_1」は「case_1 kill brenda」に変換（応募版との差分）

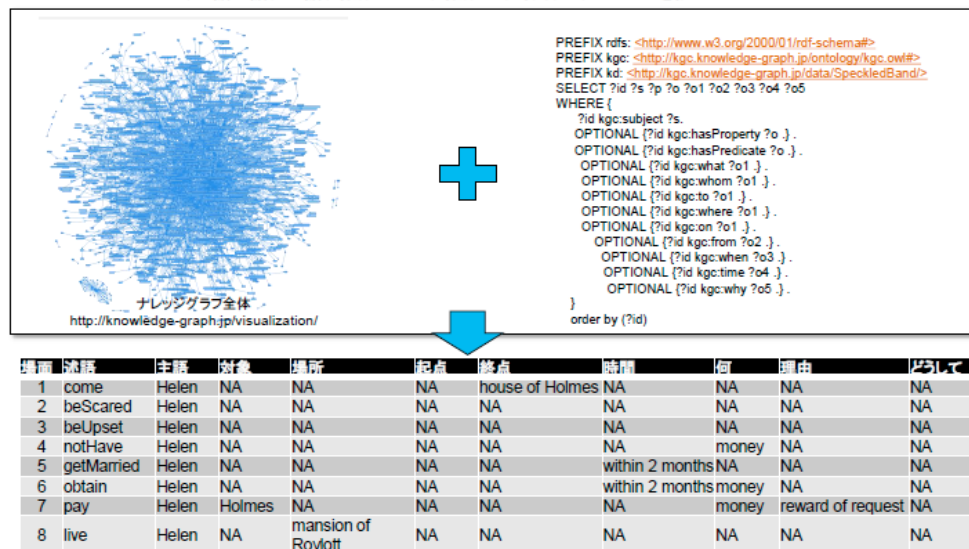
①オントロジーデータから、SPARQLで、主語述語目的語のRDFデータを抽出

■手順の目的

- 関係性解析をテンソル形式で行うため、ナレッジグラフを変換する。

■手順の概要

- 以下のSPARQLファイルで、主語述語目的語、場所FROM、場所TO、時間、理由 データを抽出



NRI Copyright (C) Nomura Research Institute, Ltd. All rights reserved.

4

引用元 : https://challenge.knowledge-graph.jp/submissions/2018/tamura/submission_tamura.pdf

■ 後処理

- ヘッド(h)=主語、リレーション(r)=述語は上表より抽出
- 「対象」～「どうして」まで全てをテイル(t)とみなす
- $hrt_1t_2t_3...$ は、 hrt_1 、 hrt_2 、 $hrt_3...$ に分解

推論過程の説明～②三つ組の各要素の特徴空間へ埋め込み

■ 単一ナレッジグラフの場合

- TransE[1] : $\vec{h} + \vec{r} \sim \vec{t}$ の関係ができる限り成り立つように埋め込みベクトルを学習
- PTransE[2] : 関係のパス $\vec{p} = \vec{r}_1 \circ \vec{r}_2 \circ \dots$ も考慮

■ 複数ナレッジグラフの場合

- ITransE[3] : TransE+パラメータシェア
- IPTransE[3] : PTransE+パラメータシェア
- ここではソフトアラインメントは用いず、パラメータシェアのみ

■ 手法の選択理由

- 共通項 (holmes、watson) が少ないため、線形変換 (MTransE) は学習が難しいと予想
- 共通項が少ないため、ソフトアラインメントの予測モデルも学習が難しいと予想
(ソフトアラインメントに依存するBootEAは採用しにくい、また、ITransEのソフトアラインメントも採用しにくい)

➡ 以上の選択が正しい保証はないので、いろいろ試して頂きたい

[1] Bordes, A. et al., NeurIPS2013. [2] Lin, Y. et al., EMNLP2015. [3] Zhu, H. et al., IJCAI2017.

(再掲) 複数ナレッジグラフの埋め込み

- 複数ナレッジグラフ間で共通項（実体ペア）がある前提で埋め込み
 - 共通項の例：日本=Japan（日本語と英語のナレッジグラフの場合）
- タスク：欠損しているリンクを予測する、または、欠損している共通項を予測する

■ よくあるテクニック

- パラメータ変換...共通項の埋め込み間で線形変換を仮定
- **パラメータシェア...共通項の埋め込みを共通化**
- パラメータスワップ...三つ組のうち共通項の一方を他方で置換
- ソフトアラインメント...共通項らしいとモデルが判断した実体ペアを擬似的に共通項とする

有名な手法	主な特徴	三つ組ロス関数	負例サンプリング	アラインメントロス関数	ソフトアラインメント
MTransE[1]	パラメータ変換	TransE(負例なし)と同じ	なし	変換ロス(ユークリッド)	無し
I(P)TransE[2]	パラメータシェア +パラメータスワップ	(P)TransEと同じ	ランダム	0（パラメータシェアなのでロスを測れない）	有り
BootEA[3]	パラメータスワップ	Limit-based loss	独自（ ϵ -truncated）	半教師あり学習の尤度	有り

[1] Chen, M. et al., IJCAI2017. [2] Zhu, H. et al., IJCAI2017. [3] Sun, Z. et al., IJCAI2018.

■悪魔の足 (case2)



犯人当て
(?, kill, mortimer)

➤➤ $\overrightarrow{mortimer} - \overrightarrow{kill}$ の近傍の
エンティティをランキング*
正解: standaleを当てられるか?

■まだらの紐



犯人当て
(?, kill, julia)

➤➤ $\overrightarrow{julia} - \overrightarrow{kill}$ の近傍の
エンティティをランキング*
正解: roylottを当てられるか?

■事後フィルタ

- ランキングで人間でない要素が出てきた場合は、除外する
- ランキングでholmes、watsonが出てきた場合は、除外する

■ データの統計値

5KG...5つのナレッシググラフ全て使用の場合

2KG...複数ナレッシググラフ(悪魔の足+まだらの紐)使用の場合

1KG...単一ナレッシググラフ(悪魔の足)使用の場合

	悪魔の足	まだらの紐	同一事件	背中の曲 がった男	踊る人形
使用ID (フル)	1-489	1-401	1-580	1-124	1-231
使用ID (-10%)	1-440	1-360	1-522	1-111	1-207
使用ID (-25%)	1-366	1-300	1-435	1-93	1-173
エンティティ数	579	266	648	121	226
リレーション数	208	110	163	61	71

297

469

■ リレーションが*kill*である三つ組

- mortimer, *kill*, brenda (悪魔の足)
- case_1, *kill*, brenda (悪魔の足)
- mortimer, *kill*, for_money (悪魔の足)
- abe_slaney, *kill*, qubit (踊る人形)

■ 人物の一覧 (Holmes、Watsonを除く)

● 悪魔の足

- standale, mortimer, brenda, roundhay, owen, george, doctor_richard, porter, tin_miner, servant, former_wife

● まだらの紐

- julia, helen, roylott, mother_of_helen, sister, mother_of_sister, roma, percy_armitage

- 「②三つ組の各要素の特徴空間へ埋め込み」 ステップで用いるパラメータ
 - マージン $\gamma=1$
 - 負例は正例1個に対し1個
 - 勾配降下法で学習（学習率=0.001）

🔍 犯人当て

- 利用するナレッジグラフの数を増やすと（1KG→2KG→5KG）、犯人の順位が上昇
➡他のナレッジグラフからの知識流入により、犯人推論の精度向上が可能

■ 5KGの場合

＜悪魔の足(case2)＞

- **IPTransEは、
犯人を1位で当てている**
(ITransEは2位)

＜まだらの紐＞

- **IPTransEは、
犯人を2位で当てている**
(ITransEも2位)

- 順位詳細は次ページ

	TransE	ITransE	
	1KG	2KG	5KG
悪魔の足(case2)	6(480)	1(3)	2(3)
まだらの紐	-	3(3)	2(2)

	PTransE	IPTransE	
	1KG	2KG	5KG
悪魔の足(case2)	5(71)	4(229)	1(1)
まだらの紐	-	2(2)	2(2)

※括弧内は事後フィルタ前の順位

■ IPTransE(5KG)の順位詳細

	1位	2位
悪魔の足(case2) (?, kill, mortimer)	DevilsFoot_standale	
まだらの紐 (?, kill, julia)	SpeckledBand_helen	SpeckledBand_roylott

■ cf. ITransE(5KG)の順位詳細

まだらの紐はいずれもhelenを本命視

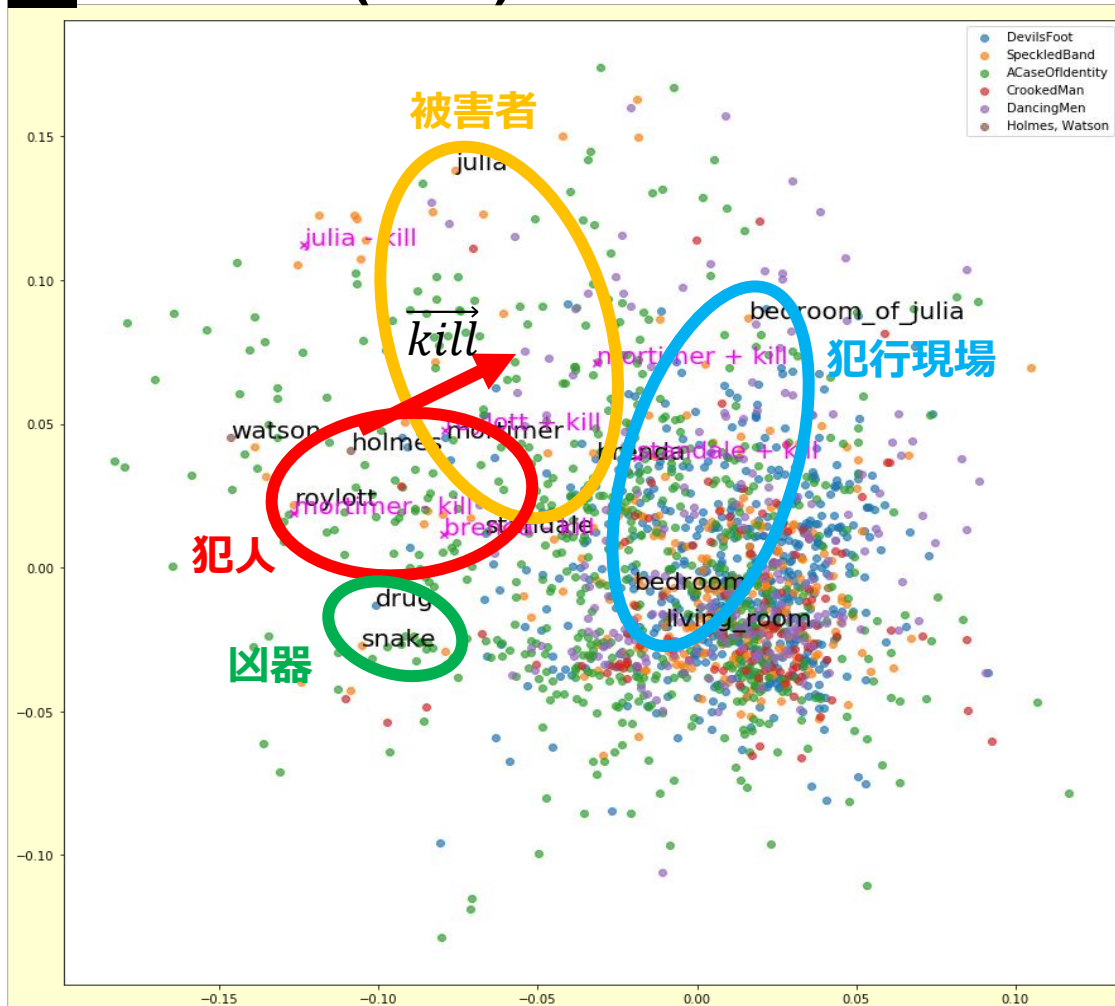
	1位	2位	3位
悪魔の足(case2) (?, kill, mortimer)	holmes	DevilsFoot_roundhay	DevilsFoot_standale
まだらの紐 (?, kill, julia)	SpeckledBand_helen	SpeckledBand_roylott	

holmesは事後フィルタにより除外
ITransEはroundhayを2位で誤答

誤答 正答

Q 特徴空間は事件の構造を捉えられているか？

A IPTransE(5KG)の特徴空間の2D可視化(t-SNE)



- 犯行場所は被害者に近い領域にある
- 凶器は犯人に近い領域にある
- ホームス、ワトソン(とくにホームス)は犯人に近い
- \overrightarrow{kill} は犯人と被害者をつなぐノルム、方向を持っているように見える

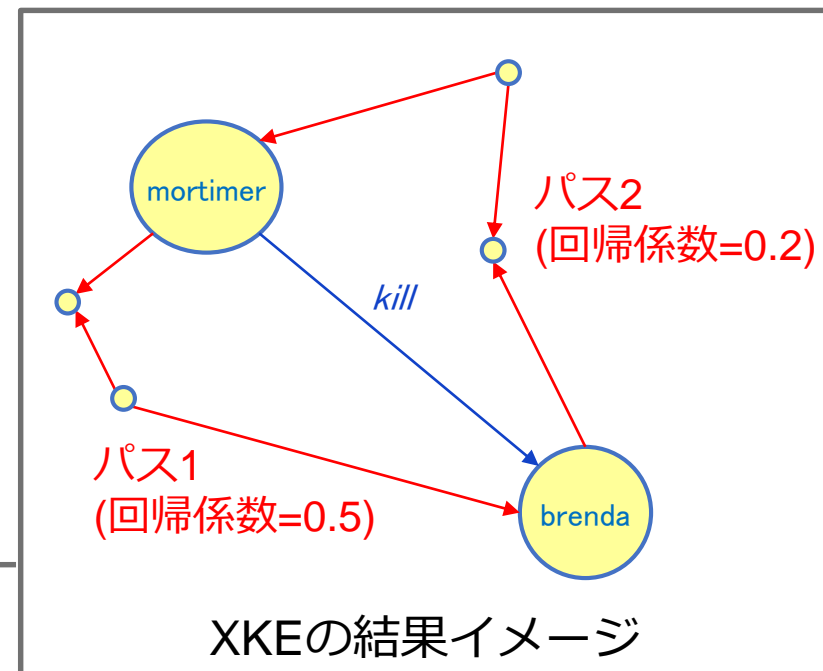
Q killは何をあらわしているか？

A 下記手法により \overrightarrow{kill} を説明する重要パスを探索

*ナレッジグラフ埋め込みの説明法 (XKE[1]) を改変

特徴空間上の三つ組のスコアを迂回パスの有無で当てられる？

- 正例作成：killを含む三つ組に対し、特徴量=ヘッドとテールを結ぶパスの有無を表すマルチホットベクトル、ラベル $=\text{softmax}(-\|\vec{h} + \vec{r} - \vec{t}\|)$ とする正例サンプルを作成
- 負例作成：killを含まない三つ組に対し、リレーションをkillで置き換えた場合の負例サンプル(特徴量、ラベルの生成方法は正例サンプルと同様)を作成
- 以上の正例、負例集合をもとに線形回帰を実行し、回帰係数を導出



➤ IPTransE(5KG)のkillに対する正の回帰係数（正例由来に限定）

パス	回帰係数	由来
-shoot-	8.48E-05	踊る人形(abe_slaney kill qubit)
-shoot_shoot-shoot-	8.48E-05	踊る人形(abe_slaney kill qubit)
-_it_was_burning-it_was_burning-observe-i_have_loved-	5.46E-06	悪魔の足(case_1 kill brenda)
-_it_was_burning-it_was_burning-observe-i_have_loved-	5.46E-06	悪魔の足(case_1 kill brenda)

※_が付いたものは逆向きリンクを意味

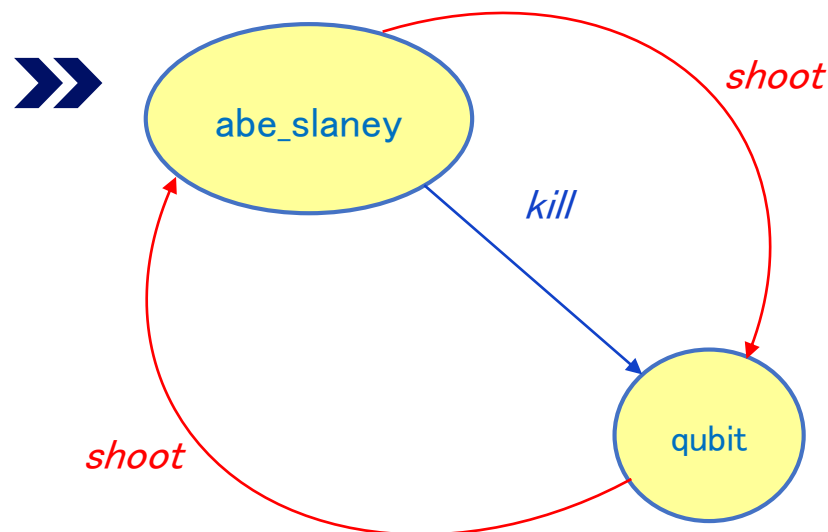
[1] Gusmão, A.C. et al., ICML 2018 Workshop (WHI 2018).

Q killは何をあらわしているか？

A IPTransE(5KG)のkillに対する正の回帰係数（正例由来に限定）

パス	回帰係数	由来
-shoot-	8.48E-05	踊る人形(abe_slaney <i>kill</i> qubit)
-shoot_shoot-shoot-	8.48E-05	踊る人形(abe_slaney <i>kill</i> qubit)
-_it_was_burning-it_was_burning-observe-i_have_loved-	5.46E-06	悪魔の足(case_1 <i>kill</i> brenda)
-_it_was_burning-it_was_burning-observe-i_have_loved-	5.46E-06	悪魔の足(case_1 <i>kill</i> brenda)

※_が付いたものは逆向きリンクを意味



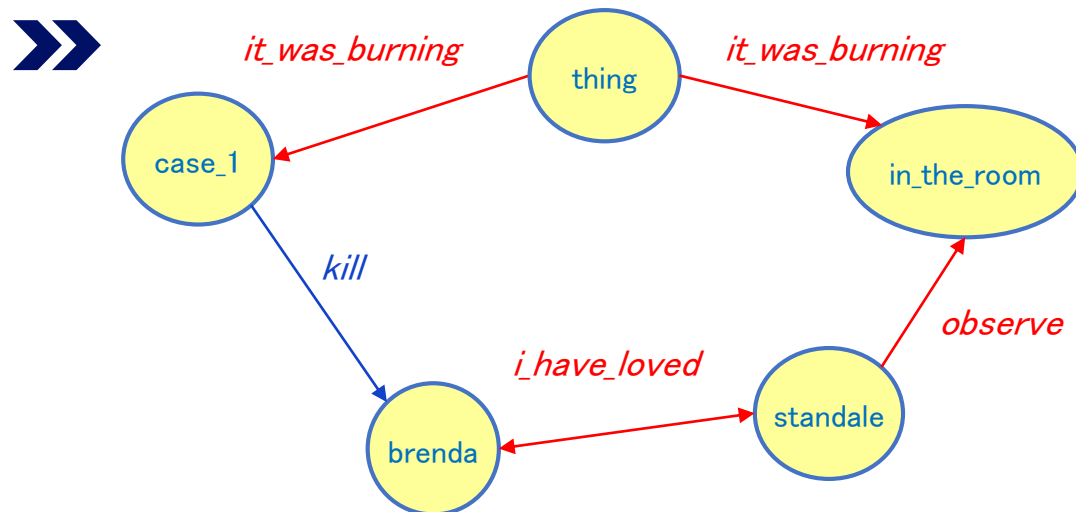
➤➤ 原文：
キュービットはエイブ・スレイニを撃った
エイブ・スレイニはキュービットを撃った
→怨恨を感じさせる
（「悪魔の足」と類似した状況を学習）

Q killは何をあらわしているか？

A IPTransE(5KG)のkillに対する正の回帰係数

パス	回帰係数	由来
-shoot-	8.48E-05	踊る人形(abe_slaney kill qubit)
-shoot_shoot-shoot-	8.48E-05	踊る人形(abe_slaney kill qubit)
-_it_was_burning-it_was_burning-observe-i_have_loved-	5.46E-06	悪魔の足(case_1 kill brenda)
-_it_was_burning-it_was_burning-observe-i_have_loved-	5.46E-06	悪魔の足(case_1 kill brenda)

※_が付いたものは逆向きリンクを意味



原文：
スタンデールは部屋の中を観察した
事件1と事件2において、部屋の中でものが燃えていた
→事件1の状況証拠を見ている→動機を示唆

スタンデールは何年もブレンダを愛してきた
→愛憎を示唆

Q なぜ悪魔の足で犯人をstandaleと当てられたのか？

A 重要パス（正の回帰係数を持つ上位のパス）

パス	回帰係数
-shoot-	8.48E-05
-shoot-_shoot-shoot-	8.48E-05

Q なぜまだらの紐で犯人をroylottと当てられたのか？

A 重要パス（正の回帰係数を持つ上位のパス）

パス	回帰係数
-come-_come-have-_have-	2.61E-07
-come-_come-come-_come-	2.41E-07

Q なぜまだらの紐で犯人をhelenと間違えたのか？

A 重要パス（正の回帰係数を持つ上位のパス）

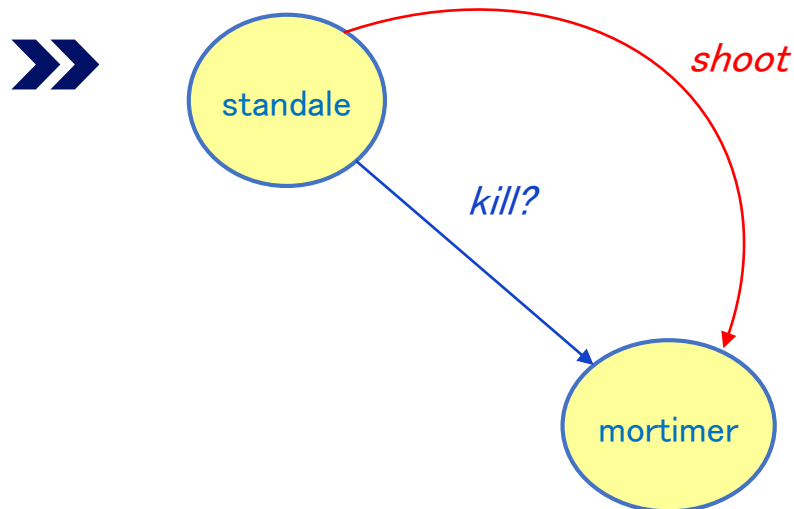
パス	回帰係数
-exist-_exist-	6.85E-06
-exist-_exist-exist-_exist-	6.85E-06

Q なぜ悪魔の足で犯人をstandaleと当てられたのか？

A 重要パス（正の回帰係数を持つ上位のパス）

パス	回帰係数
-shoot-	8.48E-05
-shoot-_shoot-shoot-	8.48E-05

※_が付いたものは逆向きリンクを意味



➤➤ 原文：
スタンデールは以下のように脅した
「スタンデールはモーティマーを撃つ」

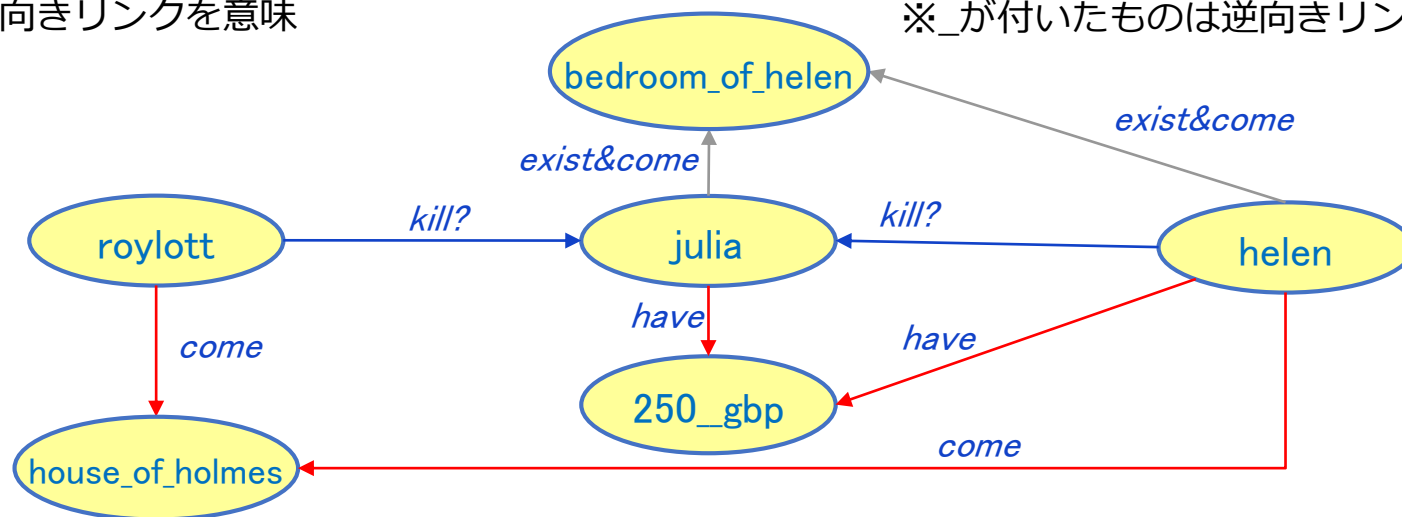
→死因ではないが、怨恨を示唆

Q なぜまだらの紐で犯人をroylottと当てられた？

A 重要パス（正の回帰係数を持つ上位のパス）

パス	回帰係数
-come-_come-have-_have-	2.61E-07
-come-_come-come-_come-	2.41E-07

※_が付いたものは逆向きリンクを意味



➡ 原文：ロイロット博士はホームズのもとへ訪ねて来た。

（ロイロット博士はヘレンをつけてきた。）

原文：ヘレンとジュリアは結婚すると1人につき
年収250ポンドずつ受け取れる。

→動機を示唆

Q なぜまだらの紐で犯人をhelenと間違えた？

A 重要パス（正の回帰係数を持つ上位のパス）

パス	回帰係数
-exist-_exist-	6.85E-06
-exist-_exist-exist-_exist-	6.85E-06

※_が付いたものは逆向きリンクを意味

原文：

事件当夜、ジュリアは部屋にひとりだった。

ヘレンは今は真ん中の部屋で寝ている。

→機会を示唆（ただし、シーンは異なる）

■ 正解が分かっている(?, kill, brenda)は1位では当てられていない

→ *kill* は汎化されているのではないか

■ 「悪魔の足」「まだらの紐」以外で有用なKGは？

→ *kill* を含む「踊る人形」が重要だが「同一事件」も有効そう

query	answer	PTransE				IPTransE							
		1KG		2KG		3KG(+踊る人形)		3KG(+同一事件)		3KG(+背中の曲がった男)		5KG	
		登場人物	全体	登場人物	全体	登場人物	全体	登場人物	全体	登場人物	全体	登場人物	全体
(?, kill, brenda)	mortimer	4	21	5	435	3	8	4	129	4	412	5	106
(mortimer, kill, ?)	brenda	3	13	6	19	4	12	6	8	7	52	7	10
(?, kill, mortimer)	standale	5	71	4	229	4	12	1	3	8	532	1	1
(standale, kill, ?)	mortimer	8	247	1	2	1	13	3	4	3	10	1	1
(?, kill, julia)	roylott	-	-	2	2	1	1	1	1	2	8	2	2
(roylott, kill, ?)	julia	-	-	3	3	3	8	1	1	2	3	1	1

■ 犯人の推論

- ナレッジグラフの埋め込みに基づき推論した結果、悪魔の足では犯人を当てることに成功
* IPTransEで5つのナレッジグラフ全てを用いた場合

- 悪魔の足では犯人 (standale) を1位で当てることに成功
- まだらの紐では犯人 (roylott) は2位 (1位はhelen)

- 利用するナレッジグラフの数を増やすと (1KG→2KG→5KG)、精度 (ランキング) が上昇

➡他のナレッジグラフからの知識流入により、犯人推論の精度向上が可能

■ 犯人の説明

- ナレッジグラフ埋め込みの可視化により、事件の構造をおおまかに捉えていることを発見
- ナレッジグラフ埋め込みの説明法を用い、動機に相当するリレーションパスを発見

サーベイ論文

Wang, Q. et al., Knowledge Graph Embedding: A Survey of Approaches and Applications, IEEE Trans. KDE2017.

単一ナレッジグラフの手法

(TransE) Bordes, A. et al., Translating embeddings for modeling multi-relational data, NeurIPS2013.

(TransH) Wang, Z. et al., Knowledge graph embedding by translating on hyperplanes, AAAI2014.

(TransR) Lin, Y. et al., Learning entity and relation embeddings for knowledge graph completion, AAAI2015.

(PTransE) Lin, Y. et al., Modeling relation paths for representation learning of knowledge bases, EMNLP2015.

(KBGAN) Cai, L. and Wang W.Y., KBGAN: Adversarial Learning for Knowledge Graph Embeddings, NAACL-HLT 2018.

(TorusE) Ebisu, T. and Ichise, R., TorusE: Knowledge Graph Embedding on a Lie Group, AAAI2018.

(RESCAL) Nickel, M. et al., A three-way model for collective learning on multi-relational data, ICML2011.

(DistMult) Yang, B. et al., Embedding entities and relations for learning and inference in knowledge bases, ICLR2015.

(SME) Bordes, A. et al., A semantic matching energy function for learning with multi-relational data, Machine Learning 2014.

(NTN) Socher, R. et al., Reasoning with neural tensor networks for knowledge base completion, NeurIPS2013.

(MLP) Dong X.L., et al., Knowledge vault: A web-scale approach to probabilistic knowledge fusion, SIGKDD2014.

(NAM) Liu, Q. et al., Probabilistic reasoning via deep learning: Neural association models, arXiv:1603.07704.

複数ナレッジグラフの手法

(MTransE) Chen, M. et al., Multilingual knowledge graph embeddings for cross-lingual knowledge alignment, IJCAI2017.

(IPTransE) Zhu, H. et al., Iterative entity alignment via joint knowledge embeddings, IJCAI2017.

(BootEA) Sun, Z. et al., Bootstrapping entity alignment with knowledge graph embedding, IJCAI2018.

(Attention-based GCN) Xu, K. et al., Crosslingual knowledge graph alignment via graph matching neural network, ACL2019.

(Highway GCN) Wu, Y. et al., Jointly learning entity and relation representations for entity alignment, EMNLP2019.

ナレッジグラフ埋め込みの解釈

(XKE) Gusmão, A.C., et al., Interpreting Embedding Models of Knowledge Bases: A Pedagogical Approach, ICML Workshop (WHI 2018).

■ 埋め込みツール

- OpenKE: <https://github.com/thunlp/OpenKE>
- Pykg2vec: <https://github.com/Sujit-O/pykg2vec>
- PTransE: <https://github.com/thunlp/KB2E/tree/master/PTransE>
- ITransE: <https://github.com/thunlp/IEAJKE>
- まとめサイト: <https://gist.github.com/mommi84/07f7c044fa18aaaa7b5133230207d8d4>

■ 可視化ツール

- Embedding Projector: <https://projector.tensorflow.org/>

■ 解釈ツール

- XKE: <https://github.com/arthurcgusmao/XKE>

■ 利用したツール

- PTransE: <https://github.com/thunlp/KB2E/tree/master/PTransE>
- ITransE: <https://github.com/thunlp/IEAJKE>
- XKE: <https://github.com/arthurcgusmao/XKE>

■ 各手法の実行方法

- ITransE...上記ツールをそのまま実行（ソフトアラインメントを外して実行）
- TransE... ITransEに対し、共通項なしの場合に相当（共通項を与えずに実行）
- IPTransE... 実装が公開されていないため、ITransEにPTransEを独自に組み込んで実行（ソフトアラインメントを外して実行）
- PTransE... IPTransEに対し、共通項なしの場合に相当（共通項を与えずに実行）
- XKE... 推論結果の説明(2)において参考

■ 計算機のスペック

- MacBook Pro 2016（CPU : Intel Core i5 2.9GHz、メモリ : 16GB RAM）

■ 実行時間

- 9.0秒 (ITransE、5つのナレッジグラフ全て使用の場合)
- 112.5秒 (IPTransE、5つのナレッジグラフ全て使用の場合)

■ 使用メモリ量

- 約1.2MB (ITransE、5つのナレッジグラフ全て使用の場合)
- 約4.3MB (IPTransE、5つのナレッジグラフ全て使用の場合)

