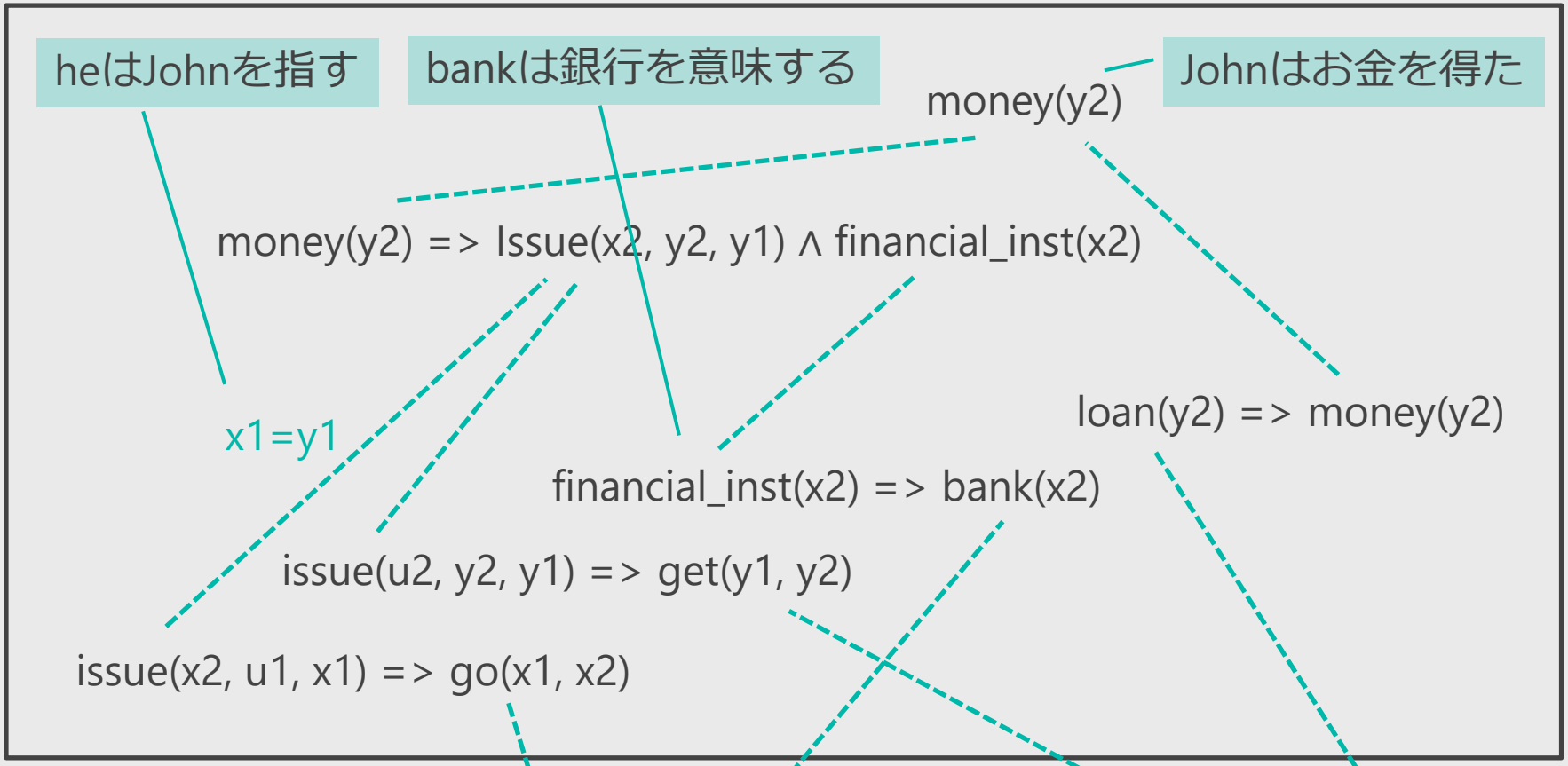
Machine Learning Approaches for Multi-hop Reasoning Over Relational Knowledge (関係知識上でのマルチホップ推論のための機械学習アプローチ)

東北大学 情報科学研究科システム情報科学専攻
乾 研究室
高橋 諒

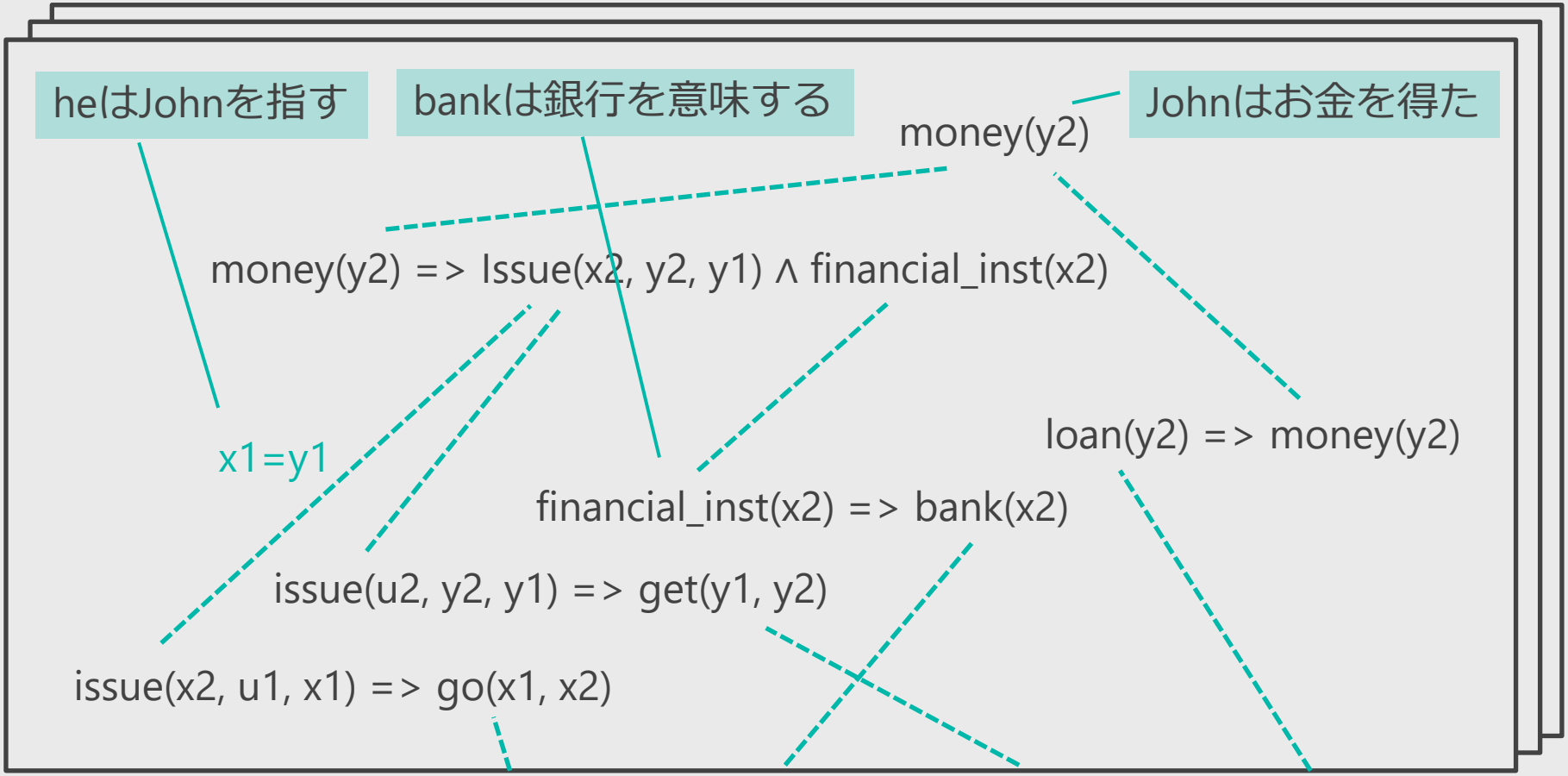
マルチホップ推論：記号的な知識を繋ぎ合わせて帰結を得る



観測： John(x1) ∧ go(x1, x2) ∧ bank(x2) ∧ he(y1) ∧ get(y1, y2) ∧ loan(y2)

入力： *John went to the bank. He got a loan.*

可能な繋ぎ合わせ方は無数にある



記号だけですべての条件を記述するのは難しい

=> 知識の**繋ぎ合わせ方**を**学習**したい

マルチホップ推論の**学習**の研究は限られている

- 古典的な記号論理の研究
  => **学習**と結びつけていない

- 深層学習に基づくend-to-endの枠組み
  => 明示的に記号的な**知識**を入れていない

本論文：**マルチホップ推論**（記号的な知識を繋ぎ合わせて帰結を得る）のための**機械学習アプローチ**を模索

本論文の構成

- 第１章　Introduction

問題設定①：**述語論理**上のマルチホップ推論（交通シーンにおける潜在的な危険予測）

- 第２章　Explaining Potential Risks in Traffic Scenes by Combining Logical Inference and Physical Simulation (Takahashi+, IJMLC'17)

問題設定②：**命題論理**上のマルチホップ推論（知識ベース補完）

- 第３章　Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder (Takahashi+, ACL'18)
- 第４章　Universal Graph Embedding: An Empirical Analysis（会誌『自然言語処理』査読中）

- 第５章　Conclusions

本論文の構成

- 第１章　Introduction

問題設定①：**述語論理**上のマルチホップ推論（交通シーンにおける潜在的な危険予測）

- 第２章　Explaining Potential Risks in Traffic Scenes by Combining Logical Inference and Physical Simulation (Takahashi+, IJMLC'17)

問題設定②：**命題論理**上のマルチホップ推論（知識ベース補完）

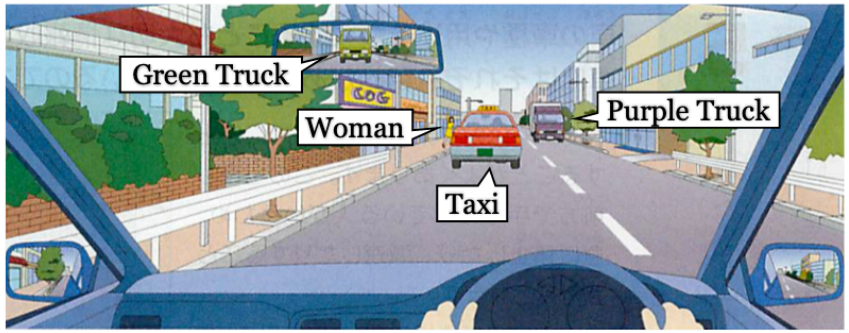
- 第３章　Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder (Takahashi+, ACL'18)
- 第４章　Universal Graph Embedding: An Empirical Analysis（会誌『自然言語処理』査読中）

- 第５章　Conclusions

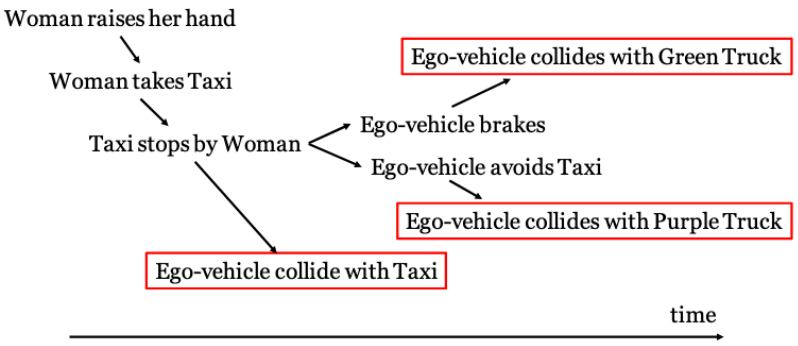
問題設定①：交通シーンにおける潜在的な危険予測 [Takahashi+'17 IJMLC]

- 仮説推論
  - 交通シーン（観測）から危険に至る推論のホップを観測されていない情報（仮説）を補いながら導く
- 知識ベース：
  - オントロジー
  - If-then ルール

貢献：
- 物理シミュレーションと統合し，物理法則に関わる推論も可能にした
- 実際の交通シーンからデータセット作成
  - 推論規則や仮説の重みを学習



(a) A risky traffic scene. (This illustration was cited from kik (1999).)



(b) A causality chains of above traffic scene. The red rectangles denote a potential risk.

交通シーンの危険予測のタスク定義

入力

センサーから得られる情報
（例：車載カメラやLiDAR）

- 定量データ（形状, 位置, 速度）
- 定性データ（交通シーンの記述）

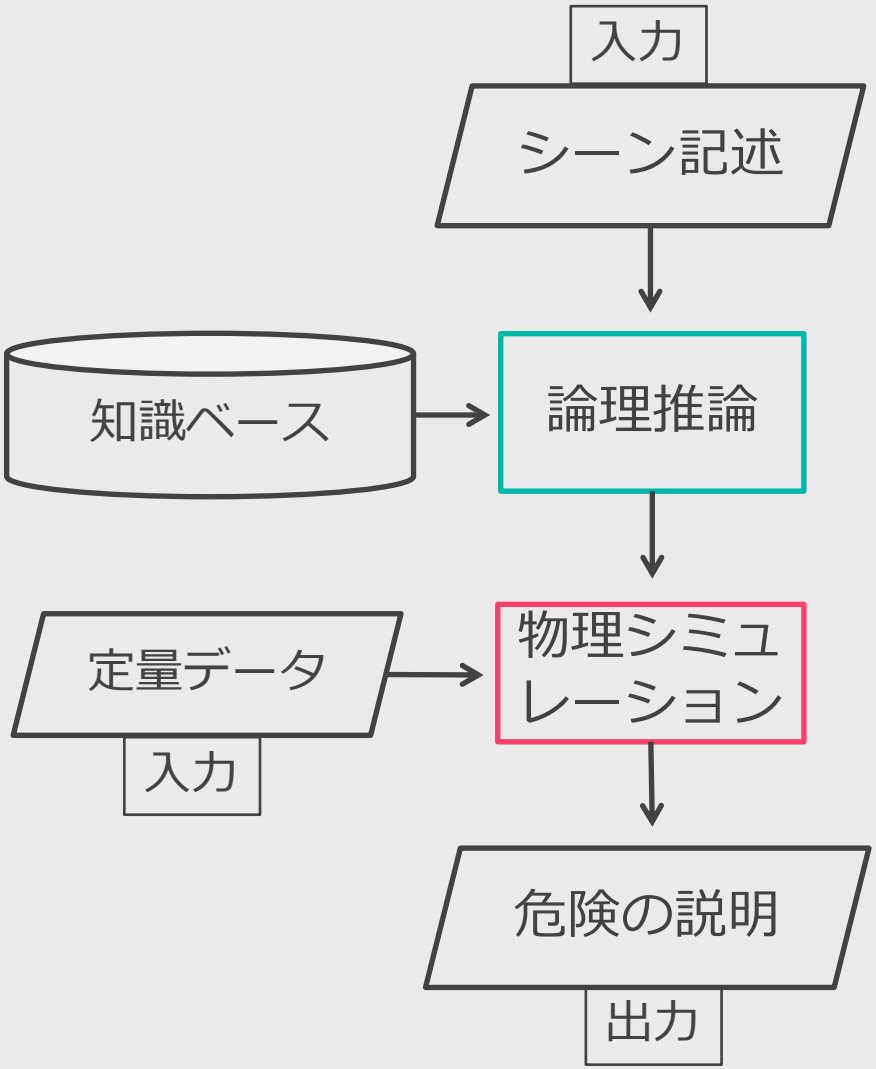
出力

尤度スコア付きの危険の説明
（二段階）

- 衝突するエンティティとその行動ペア
- 危険に至る説明全体

Score	Entity-action	Explanation
0.8	Taxi(T) stops	Woman raises her hand, Taxi suddenly stops, then the ego-vehicle collides.
0.5	Pedestrian(P) cross the road	...

リランキングに基づくパイプライン式モデル



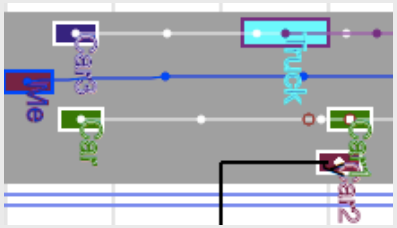
仮説推論器 [Inoue+ 15]

- 候補となる危険のランキングを定性情報を使って生成
- 「どのエンティティがどんな危険な行動を取りうるか？」

行動に基づく物理シミュレーション

- 危険の候補を定量情報を使ってリランキング

データセットとタスク設定



危険予測システム

危険なエンティティ・行動ペア

元データ

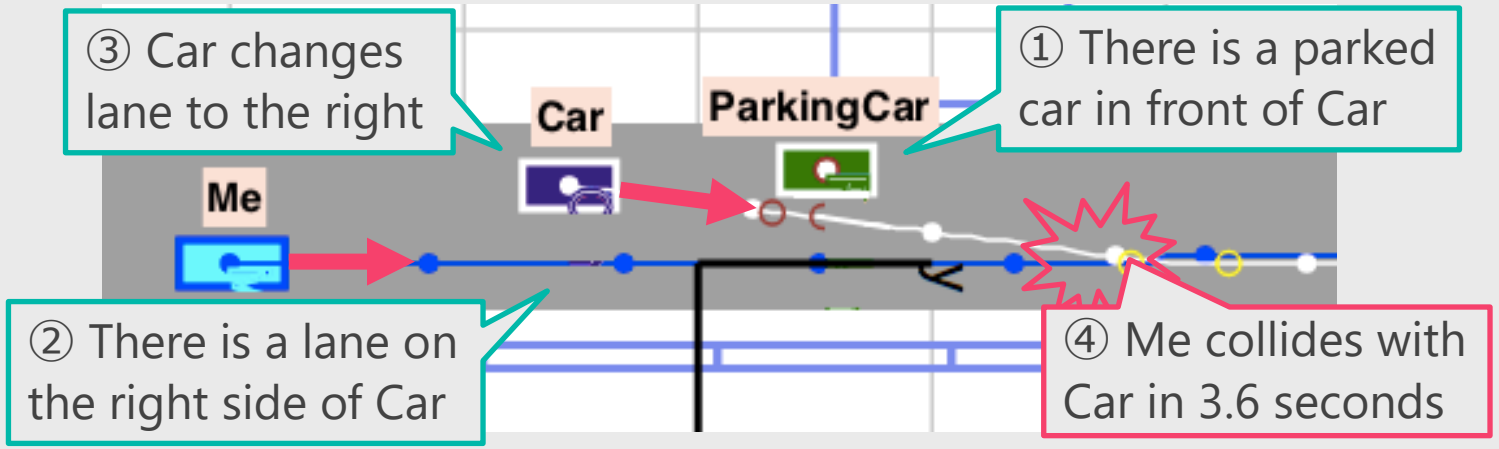
- 東京農工大学による「ヒヤリハットデータベース」
- 衝突や急ブレーキなどの危険に至る十数秒間の録画映像が10万件以上

2D俯瞰マップ

- 実際の危険の2秒前のスナップショット
- 人手で379件作成

- 知識ベースの規模：
  - If-thenルール：13種類
  - Is-a知識：211種類

実際のモデル出力例



Actual risk: Car will change lanes to avoid ParkingCar

Model	Predicted risky entity-action	Explanation	Quantitative prediction
Baseline	Car will stop	X	X
Proposed1	Car will change lanes	✓	X
Proposed2	Car will change lanes	✓	✓

本論文の構成

- 第１章　Introduction

問題設定①：**述語論理**上のマルチホップ推論（交通シーンにおける潜在的な危険予測）

- 第２章　Explaining Potential Risks in Traffic Scenes by Combining Logical Inference and Physical Simulation (Takahashi+, IJMLC'17)

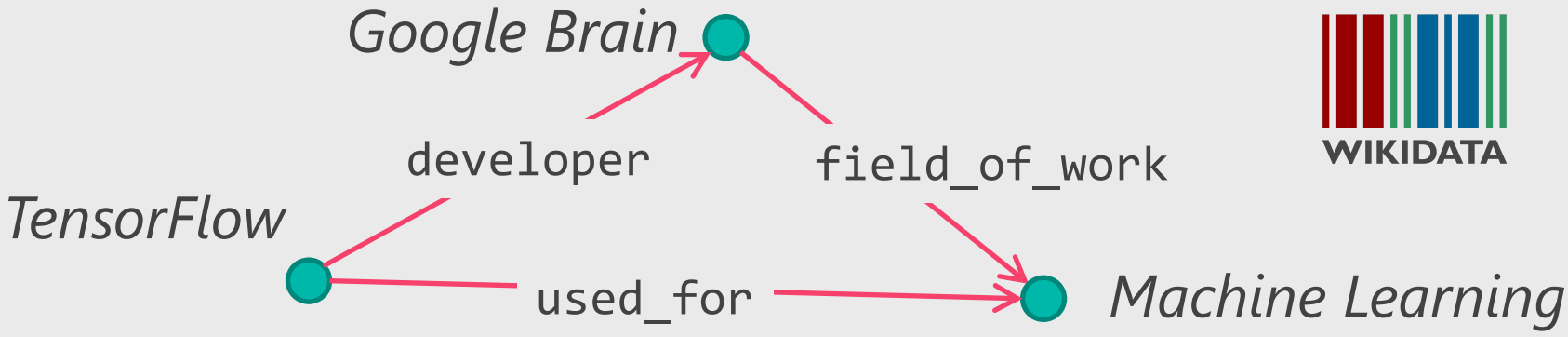
問題設定②：**命題論理**上のマルチホップ推論（知識ベース補完）

- 第３章　Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder (Takahashi+, ACL'18)
- 第４章　Universal Graph Embedding: An Empirical Analysis（会誌『自然言語処理』査読中）

- 第５章　Conclusions

知識ベース補完

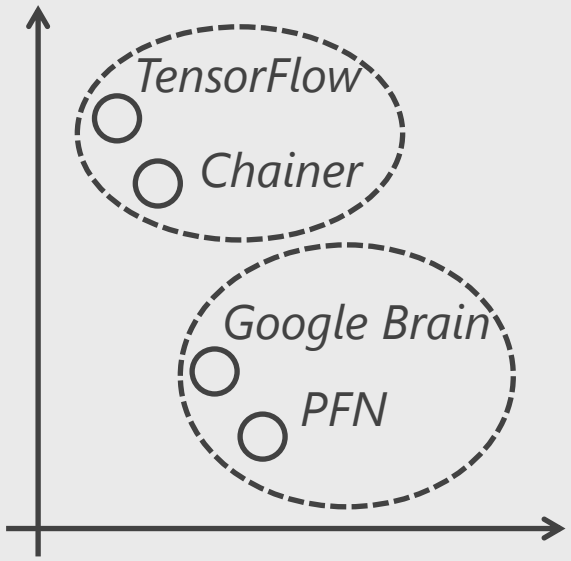
- 関係知識：「もの」と「もの」の間の関係
  - <ヘッドエンティティ, 関係, テールエンティティ> の三つ組
  - **知識ベース（知識グラフ）**：関係知識を蓄積したデータベース



- **知識ベース補完**：既知の関係知識を使って未知の関係知識を予測

Vector Based Approach：関係知識を低次元のベクトル空間でモデル化

エンティティ：似たエンティティが互いに近いような**低次元ベクトル**として表現



関係: ベクトル空間中の**変換**として表現

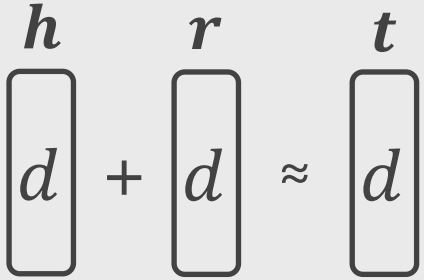
変換：

- 平行移動
- 線形写像
- 非線形変換
- …

関係の主要な２つの表現方法

TransE [Bordes+'13]

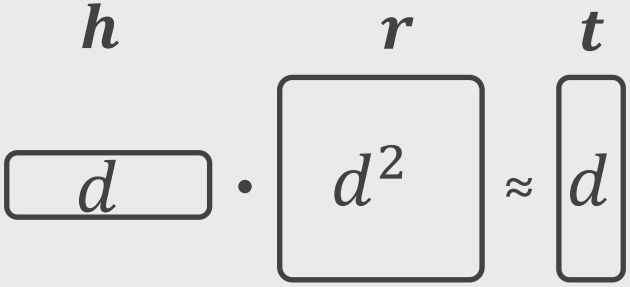
- 関係は平行移動



- 関係の表現能力が低い
  - 1対1の関係しか表現できない
- 実践的には他のモデルと遜色ない精度

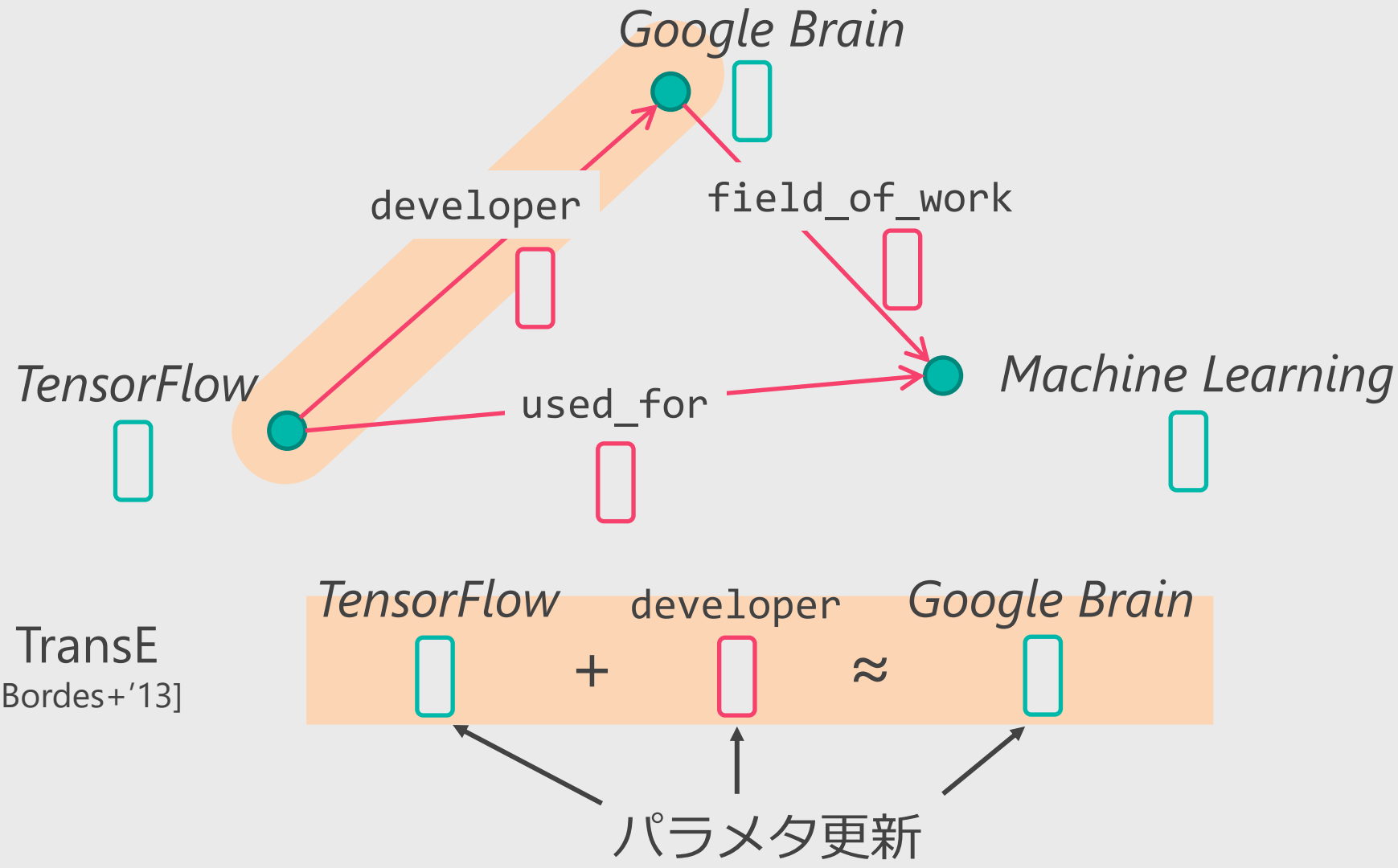
Bilinear [Nickel+'11]

- 関係は線形写像

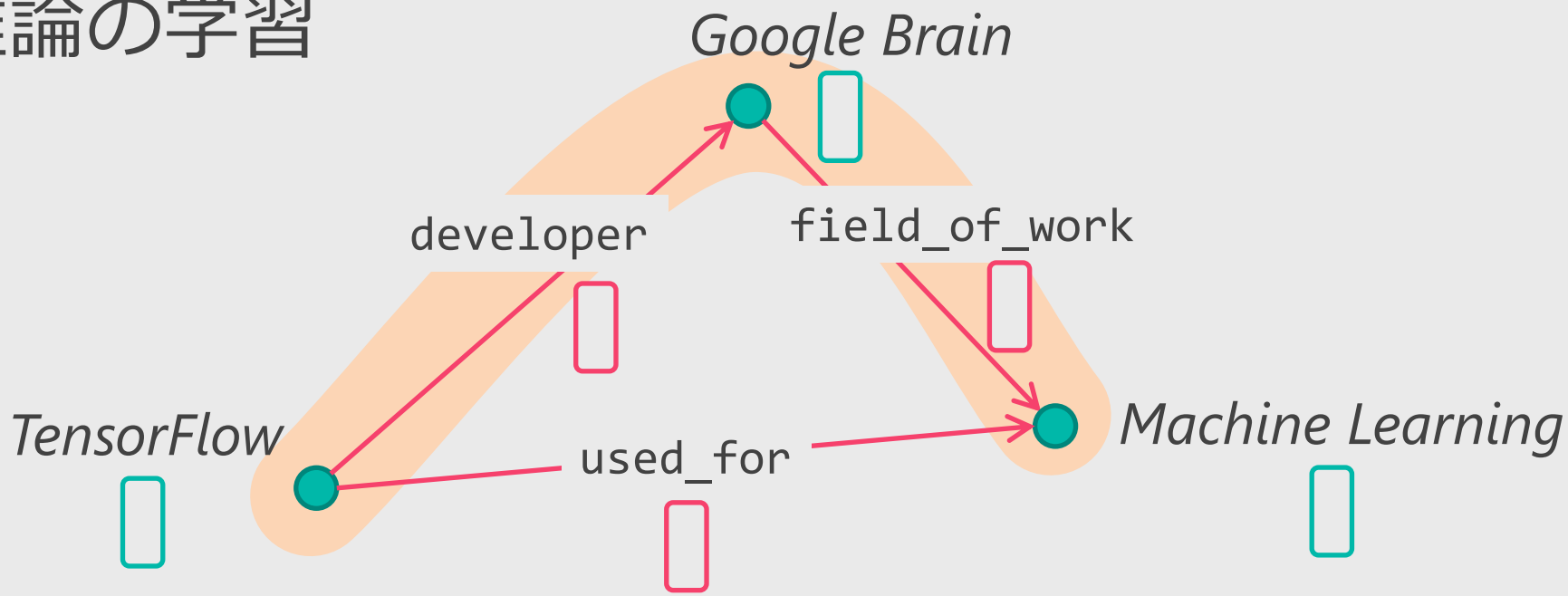


- 関係の表現能力が高い
  - 多対多の関係を表現できる
- 行列のパラメタが多く学習が難しい

Vector based modelの学習



Vector based modelにおけるマルチホップ推論の学習

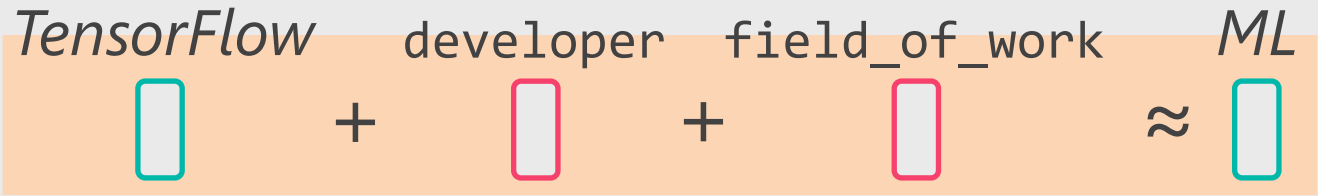


TransE
[Bordes+'13]

マルチホップTransE
[Guu+'15, Lin+'15]

Vector based modelにおけるマルチホップ推論の学習

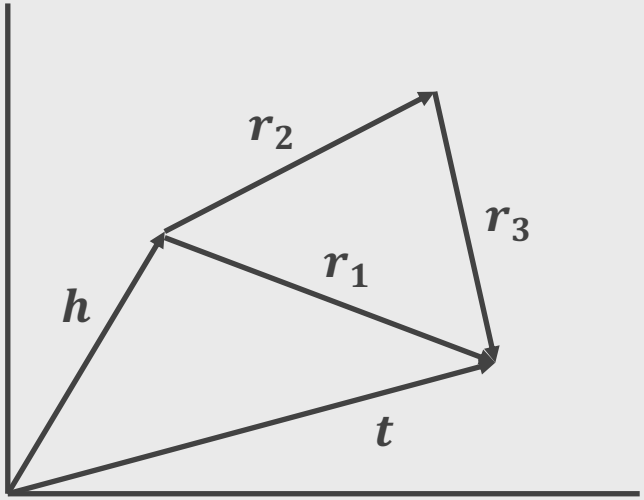
マルチホップTransE
[Guu+'15, Lin+'15]

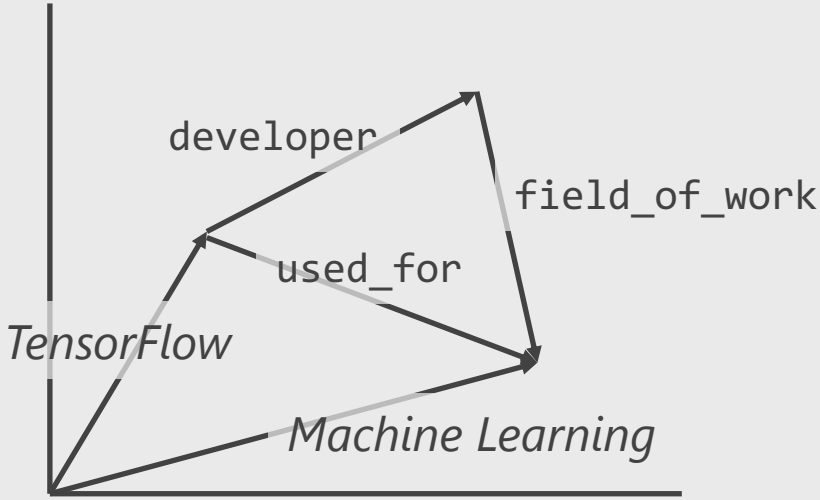


- 関係の合成をより良く捉える [Guu+'15]
  - developer + field_of_work ≈ used_for
  - それに伴う知識ベース補完の性能向上

定式化

- 直感：三つ組 $(h, r_1/\cdots/r_l, t)$ に対し，
$$\boldsymbol{h} + (\boldsymbol{r_1} + \cdots + \boldsymbol{r_l}) = \boldsymbol{t}$$
- グラフからサンプルされるパスに対してスコア関数 f が大きいほど良い
  - $f(h, r, t; \Theta) \coloneqq -\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|$
  - $f(h, p, t; \Theta) \coloneqq -\|\boldsymbol{h} + \boldsymbol{r_1} + \cdots + \boldsymbol{r_l} - \boldsymbol{t}\|$
    - $p = r_1/\cdots/r_l$





最適化

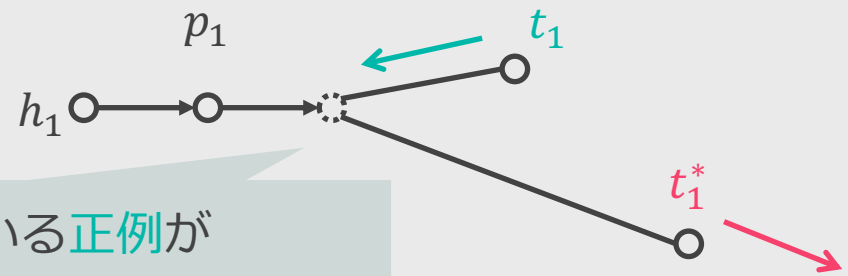
- Max-margin loss:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} \sum_{t_1^* \in \mathcal{N}(h_i, p_i)} \max(0, [\gamma + f^- - f^+])$$

$$f^- = f(h_i, p_i, t_1^*; \Theta), \qquad f^+ = f(h_i, p_i, t_i; \Theta)$$

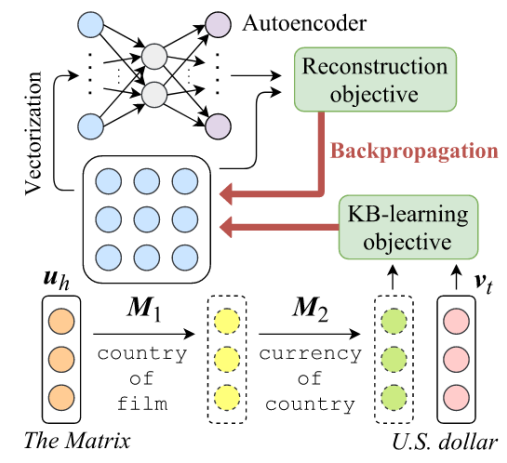
負例のスコア　　正例のスコア

  - γ: マージン
    - モデルが許容する正例と負例の間の最小の距離



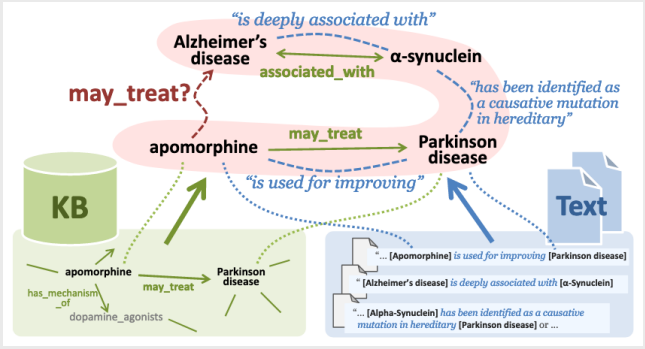
グラフ上で接続している正例が
ランダムにサンプルされた負例よりも
(h_1, p_1) で飛んだ先に近づくように学習

本論文：知識ベース補完において二つの課題に対処



Bilinearモデルのパラメタ過多問題

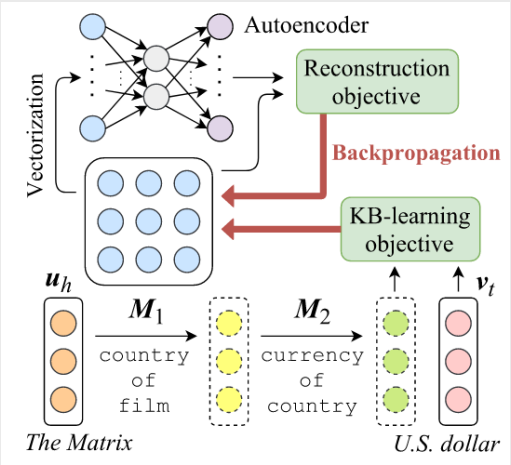
- 第３章 Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder (Takahashi+, ACL'18)
- アイデア：オートエンコーダとの同時学習による正則化



知識ベースの疎性

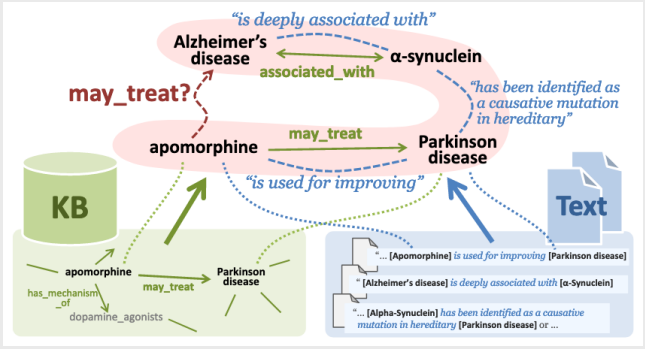
- 第４章 Universal Graph Embedding: An Empirical Analysis（会誌『自然言語処理』査読中）
- アイデア：テキストとして出現する関係との同時学習

本論文：知識ベース補完において二つの課題に対処



Bilinearモデルのパラメタ過多問題

- 第３章 Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder (Takahashi+, ACL'18)
- アイデア：オートエンコーダとの同時学習による正則化



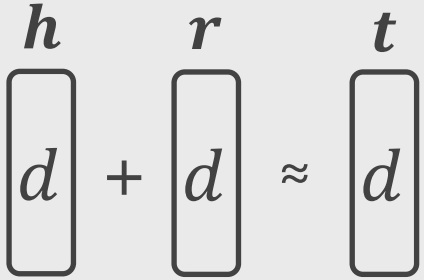
知識ベースの疎性

- 第４章 Universal Graph Embedding: An Empirical Analysis（会誌『自然言語処理』査読中）
- アイデア：テキストとして出現する関係との同時学習

関係の主要な２つの表現方法

TransE [Bordes+'13]

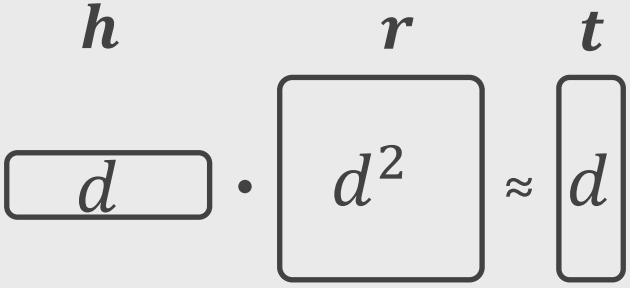
- 関係は平行移動



- 関係の表現能力が低い
  - 1対1の関係しか表現できない
- 実践的には他のモデルと遜色ない精度

Bilinear [Nickel+'11]

- 関係は線形写像



- 関係の表現能力が高い
  - 多対多の関係を表現できる
- **行列のパラメタが多く学習が難しい**

① オートエンコーダとの同時学習

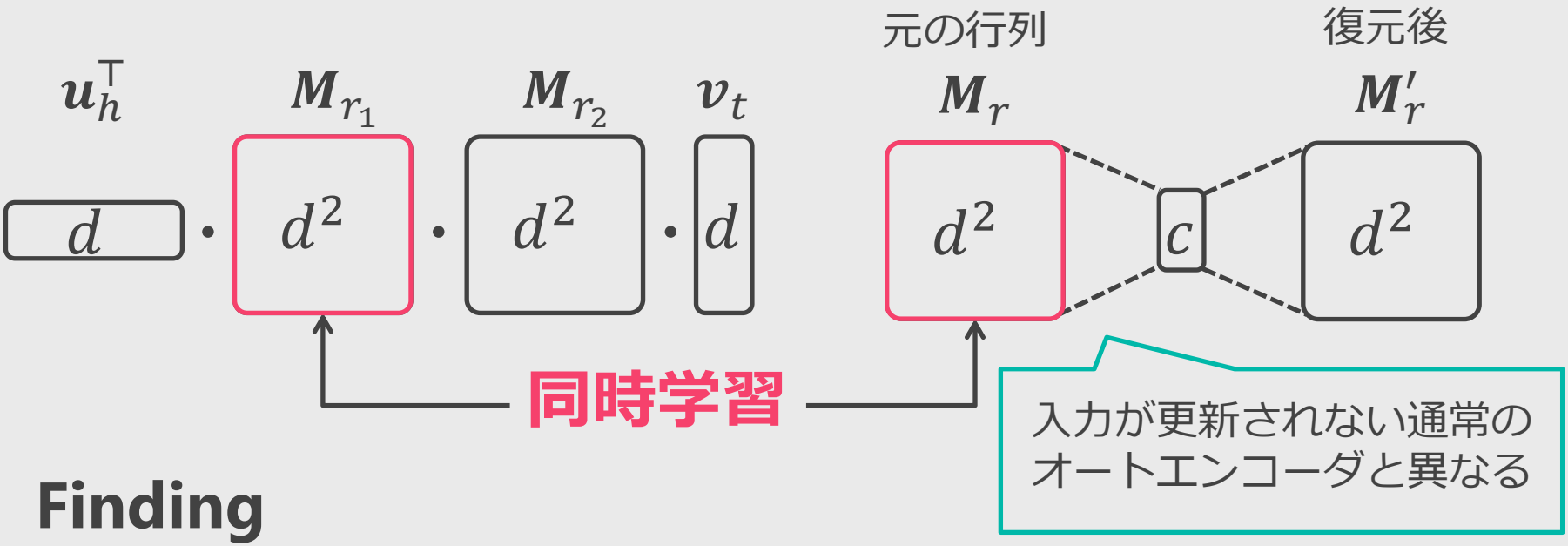
Base モデル

関係を行列で表現し，マルチホップの学習ができるよう拡張

[Nickel+'11, Guu+'15, Tian+'16]

提案手法

関係の行列を低次元のコードから復元する**オートエンコーダ**を学習



Finding

1. 関係の行列の高い次元を削減する
2. 関係の合成の学習を助ける

① オートエンコーダとの同時学習

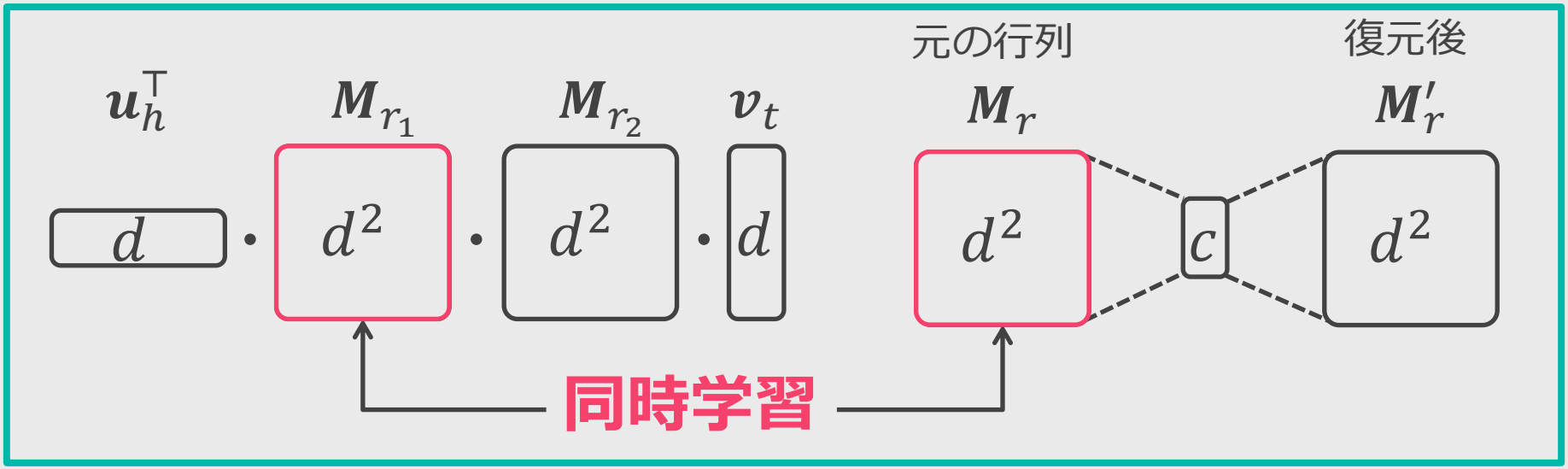
Base モデル

関係を行列で表現し，マルチホップの学習ができるよう拡張

[Nickel+'11, Guu+'15, Tian+'16]

提案手法

関係の行列を低次元のコードから復元する**オートエンコーダ**を学習



学習が難しい

目的関数が非凸

→ 簡単に局所的最小解に陥る

② SGDの改変（学習率の分離）

戦略
モデルの異なる部分には異なる学習率を持たせる

既存

SGDの学習率の一般的な設定方法
[Bottou, 2012]:

$$\alpha(\tau) \coloneqq \frac{\eta}{1+\eta\lambda\tau}$$

改変後

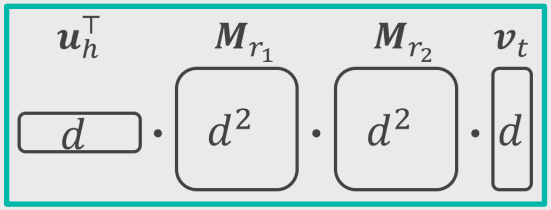
モデルの異なる部分には異なる学習率を持たせる

$$\alpha_{\mathrm{KB}}(\tau_r) \coloneqq \frac{\eta_{\mathrm{KB}}}{1+\eta_{\mathrm{KB}}\lambda_{\mathrm{KB}}\tau_r}$$

$$\alpha_{\mathrm{AE}}(\tau_r) \coloneqq \frac{\eta_{\mathrm{AE}}}{1+\eta_{\mathrm{AE}}\lambda_{\mathrm{AE}}\tau_r}$$

- η: 初期学習率
  - η_{KB}: KBの目的関数のための η
  - η_{AE}: オートエンコーダのための η
- λ: L2正則化項の係数
  - λ_{KB}: KBの目的関数のための λ
  - λ_{AE}: オートエンコーダのための λ
- τ: 訓練事例のカウンタ
  - τ_e: エンティティ e のカウンタ
  - τ_r: 関係 r のカウンタ

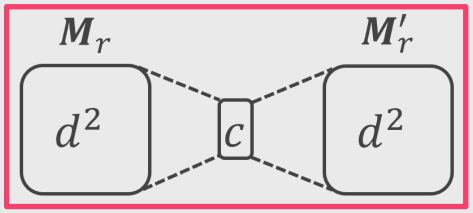
③ オートエンコーダとの同時学習のための学習率



知識ベース (KB) の目的関数

エンティティの予測

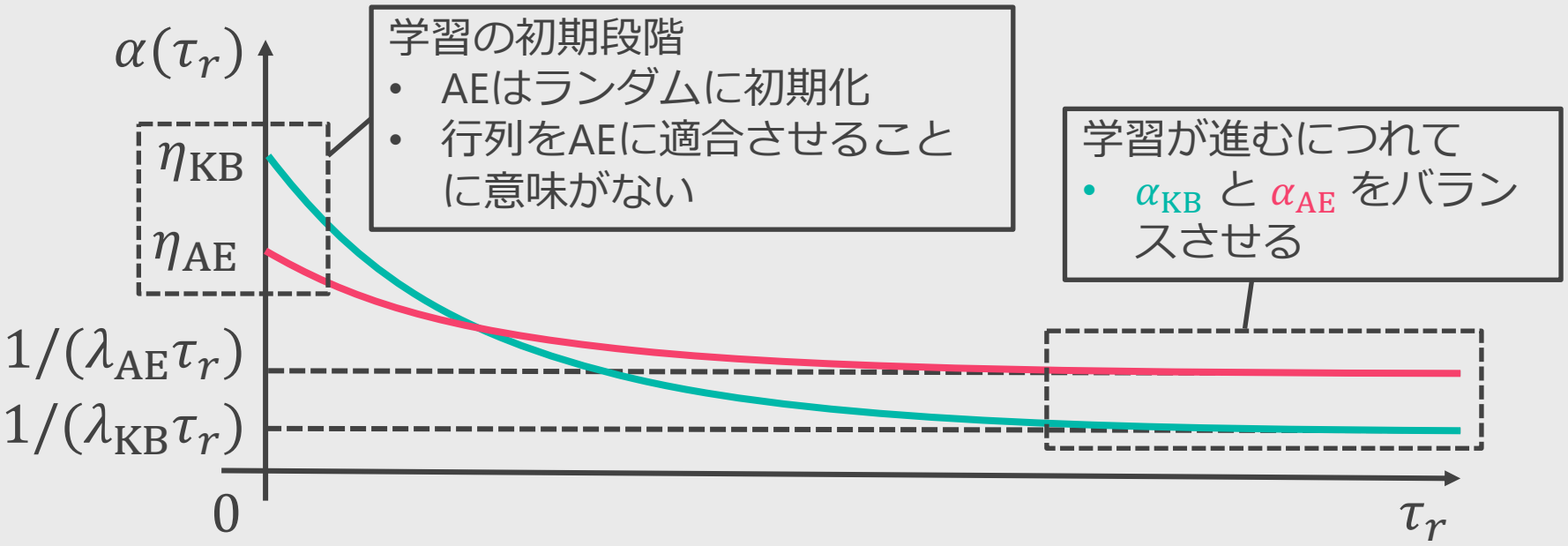
$$\alpha_{\mathrm{KB}}(\tau_r) := \frac{\eta_{\mathrm{KB}}}{1+\eta_{\mathrm{KB}}\lambda_{\mathrm{KB}}\tau_r}$$



オートエンコーダ (AE) の目的関数

低次元のコードに適合

$$\alpha_{\mathrm{AE}}(\tau_r) := \frac{\eta_{\mathrm{AE}}}{1+\eta_{\mathrm{AE}}\lambda_{\mathrm{AE}}\tau_r}$$



Base vs. オートエンコーダとの同時学習

Model	WN18RR			FB15k-237		
	MR ↓	MRR ↑	H10 ↑	MR ↓	MRR ↑	H10 ↑
Base	2447	.310	54.1	203	.328	51.5
提案手法	**2268**	**.343**	**54.8**	**197**	**.331**	**51.6**

モデル:

- **Base**: Bilinearモデル [Nickel+'11]
- **提案手法**: 関係の行列をオートエンコーダと同時に学習する

評価指標:

- **MR** (Mean Rank): **低い**方が良い
- **MRR** (Mean Reciprocal Rank): **高い**方が良い
- **H10** (Hits at 10): **高い**方が良い

オートエンコーダとの同時学習はベースのbilinearモデルの性能をさらに引き上げる

先行研究との比較

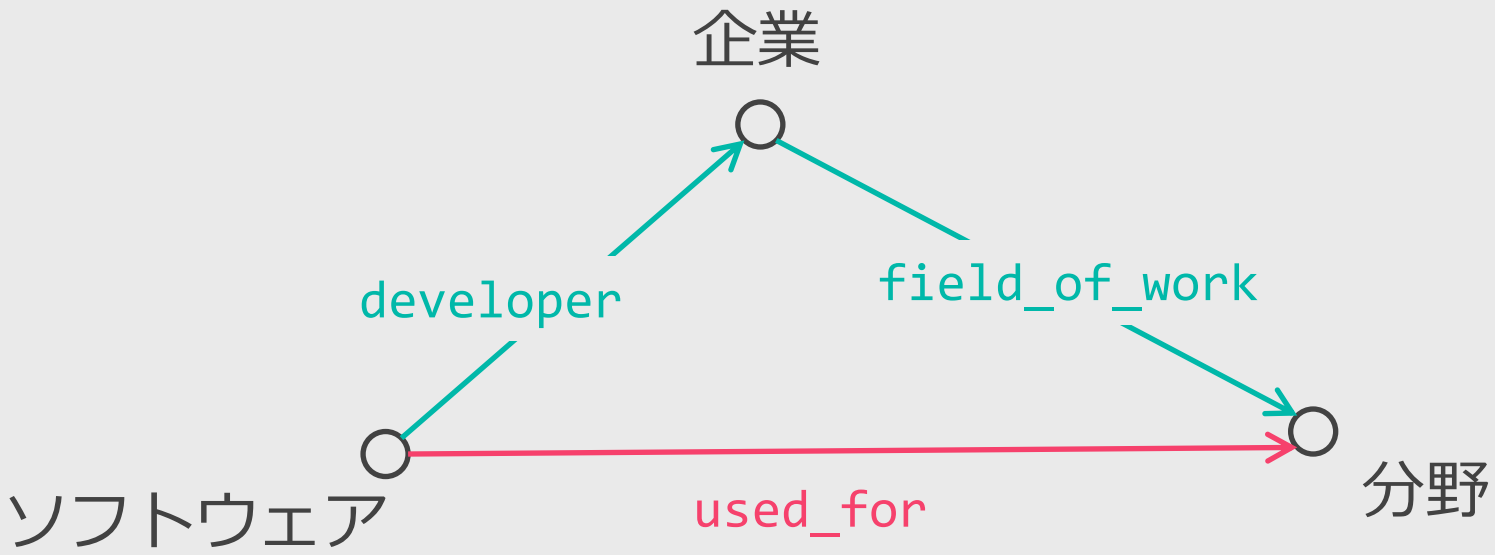
- Normalization
- Regularization
- Initialization

Model	WN18RR			FB15k-237		
	MR ↓	MRR ↑	H10 ↑	MR ↓	MRR ↑	H10 ↑
Ours						
Base	2447	.310	54.1	203	.328	51.5
提案手法	**2268**	.343	**54.8**	**197**	**.331**	**51.6**
Re-experiments						
TransE [Bordes+'13]	4311	.202	45.6	278	.236	41.6
RESCAL [Nickel+'11]	9689	.105	20.3	457	.178	31.9
HolE [Nickel+'16]	8096	.376	40.0	1172	.169	30.9
Published results						
ComplEx [Trouillon+'16]	5261	.440	51.0	339	.247	42.8
ConvE [Dettmers+'18]	5277	**.460**	48.0	246	.316	49.1

ベンチマークデータセットで提案手法が最先端の性能

関係の合成（マルチホップ推論）

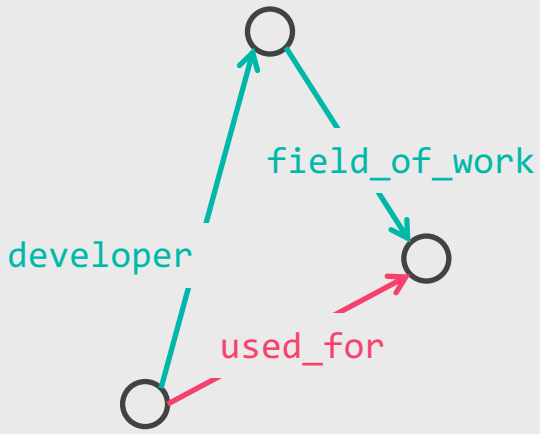
- 二つの関係の合成がもう一つの関係に一致：



- FB15k-237から154個の関係の合成を抽出

同時学習が関係の合成を見つけやすくする

関係の合成があれば…



学習した関係の行列は
合成に従うはず

$$\boldsymbol{M}_{\text{developer}} \cdot \boldsymbol{M}_{\text{field\_of\_work}} \approx \boldsymbol{M}_{\text{used\_for}}$$

Model	↓ MR	↑ MRR
Base	150±3	.0280±.0010
提案手法	**130±27**	**.0481±.0090**

オートエンコーダとの同時学習が合成の制約をより発見しやすくする

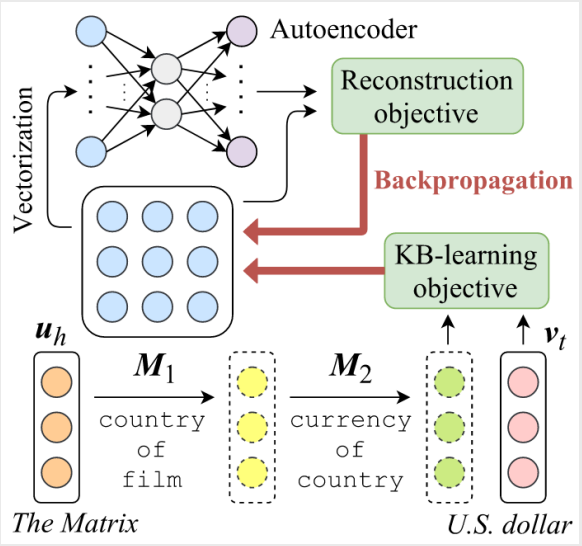
Bilinearモデルのパラメタ過多問題 [Takahashi+, ACL'18]

アプローチ

- 関係の行列を復元するオートエンコーダーと同時学習
  - 行列の実質的なパラメタ空間が小さくなる

結果

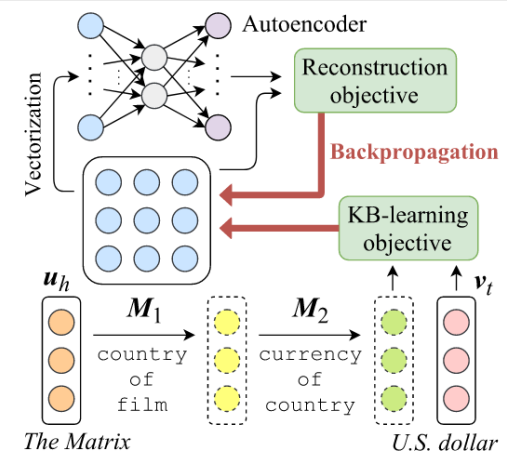
- ベンチマークデータセットFB15k-237で当時最先端の精度
- マルチホップ推論の精度向上
  - 「developer + field_of_work = used_for」をより良く捉える



Model	MR	MRR
JOINT+COMP	**130±27**	**.0481±.0090**
BASE+COMP	150±3	.0280±.0010
RANDOMM2	181±19	.0356±.0100

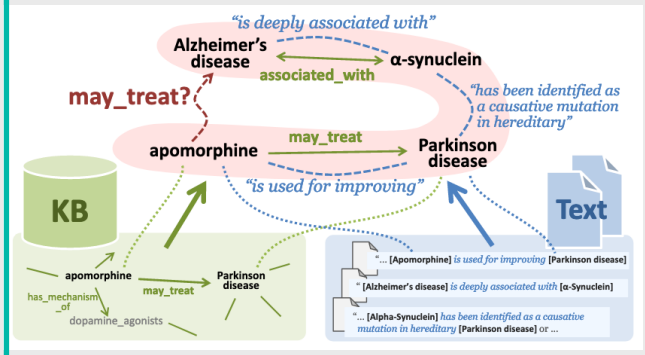
Model	WN18		FB15k		WN18RR			FB15k-237		
	MR	H10	MR	H10	MR	MRR	H10	MR	MRR	H10
JOINT	**277**	**95.8**	**53**	**82.5**	**4233**	**.461***	**53.4**	**212**	.336	**52.3***
BASE	286	**95.8**	**53**	**82.5**	4371	.459	52.9	215	**.337***	**52.3***
JOINT+COMP	**191***	**94.8**	**53**	**69.7**	**2268***	**.343**	**54.8***	**197***	**.331**	**51.6**
BASE+COMP	195	**94.8**	54	69.4	2447	.310	54.1	203	.328	51.5
TransE (Bordes et al., 2013a)	292	92.0	**66**	70.4	4311	.202	45.6	**278**	.236	41.6
TransR (Lin et al., 2015c)	**281**	93.6	76	**74.4**	**4222**	.210	**47.1**	320	**.282**	**45.9**
RESCAL (Nickel et al., 2011)	911	58.0	163	41.0	9689	.105	20.3	457	.178	31.9
HolE (Nickel et al., 2016b)	724	**94.3**	293	66.8	8096	**.376**	40.0	1172	.169	30.9
STransE (Nguyen et al., 2016)	206	93.4	69	79.9	-	-	-	-	-	-
ITransF (Xie et al., 2017)	**205**	94.2	65	81.0	-	-	-	-	-	-
ComplEx (Trouillon et al., 2016)	-	94.7	-	84.0	**5261**	.44	**51**	339	.247	42.8
Ensemble DistMult (Kadlec et al., 2017)	457	95.0	35.9	90.4	-	-	-	-	-	-
IRN (Shen et al., 2017)	249	95.3	38	**92.7***	-	-	-	-	-	-
ConvE (Dettmers et al., 2018)	504	95.5	64	87.3	5277	**.46**	48	**246**	**.316**	**49.1**
R-GCN+ (Schlichtkrull et al., 2017)	-	**96.4***	-	84.2	-	-	-	-	.249	41.7
ProjE (Shi and Weninger, 2017)	-	-	**34***	88.4	-	-	-	-	-	-

本論文：知識ベース補完において二つの課題に対処



Bilinearモデルのパラメタ過多問題

- 第３章 Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder (Takahashi+, ACL'18)
- アイデア：オートエンコーダとの同時学習による正則化

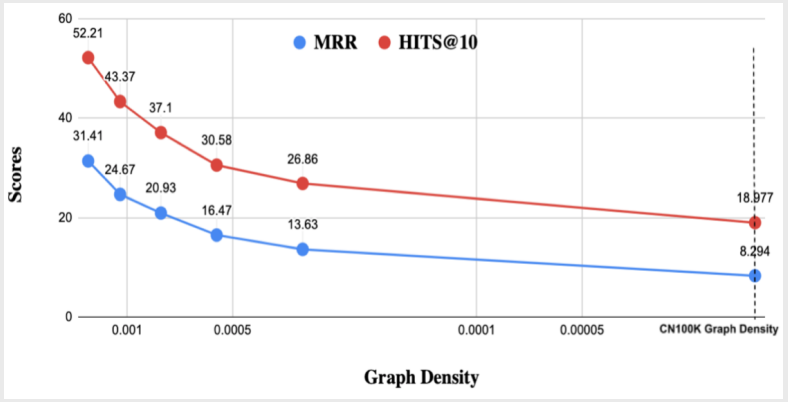


知識ベースの疎性

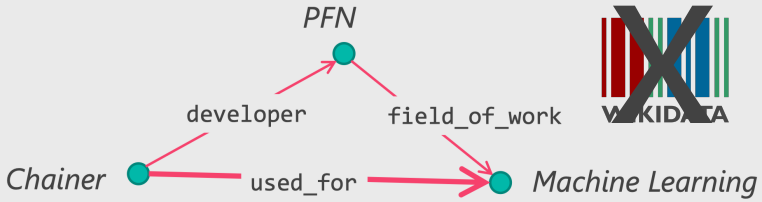
- 第４章 Universal Graph Embedding: An Empirical Analysis（会誌『自然言語処理』査読中）
- アイデア：テキストとして出現する関係との同時学習

知識ベースの疎性

- 現実の知識ベース (Wikidata, UMLS, ConceptNet, ATOMIC, …) は疎
  - 人手管理の壁
- ベンチマークデータセットは知識ベースの密な部分を取り出している
  - 既存の知識ベース補完モデルは疎なグラフで低性能 [Pujara+'17, Malaviya+'20]
- 既存の知識ベース補完の問題設定は非現実的



(Malaviya+'20)



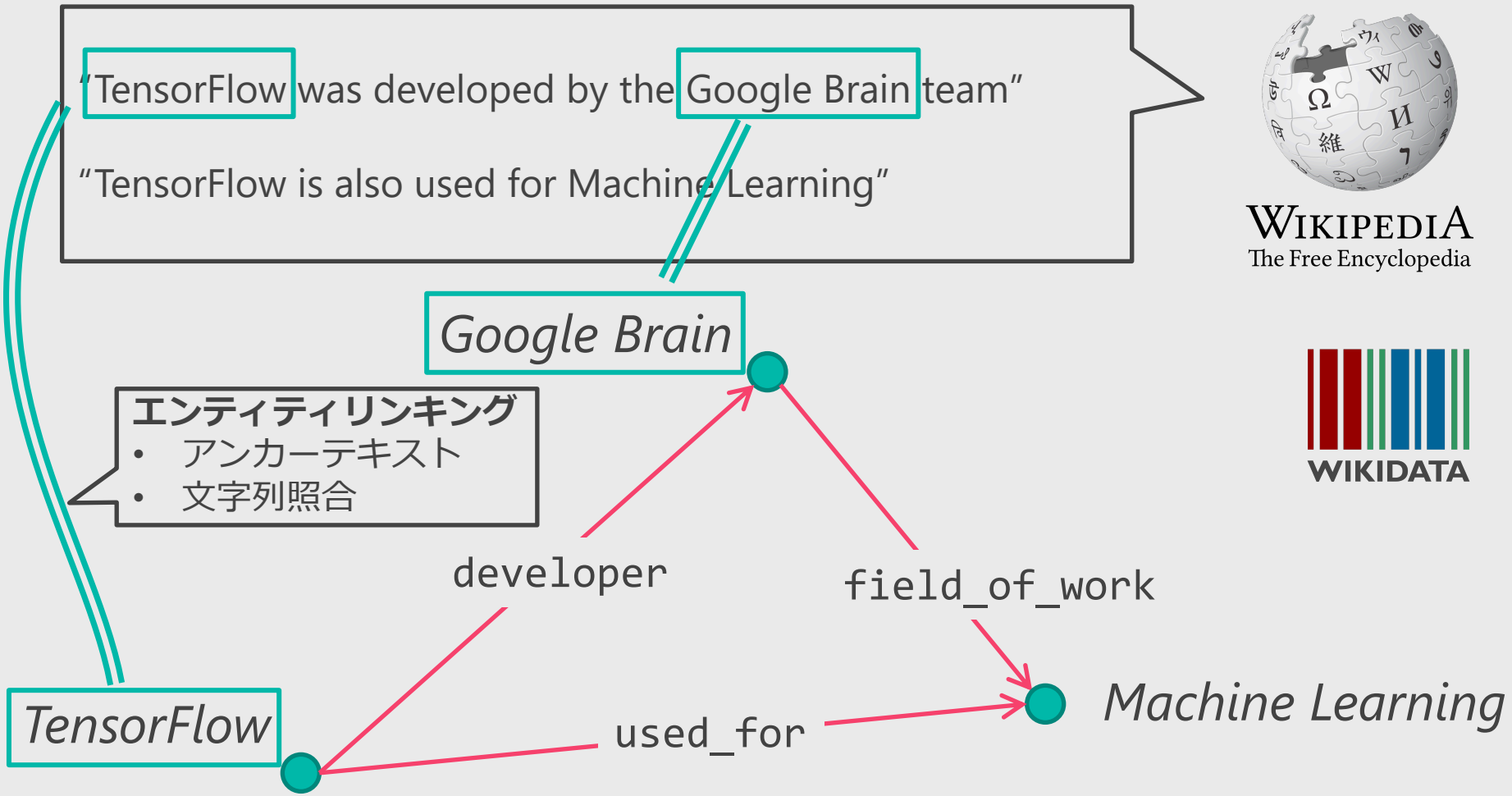
アイデア：多くの関係知識は知識ベースに存在せずとも**テキスト**して書かれている
→ **テキストの関係知識**を追加して密なグラフを構築

テキストコーパスから密なグラフを構築

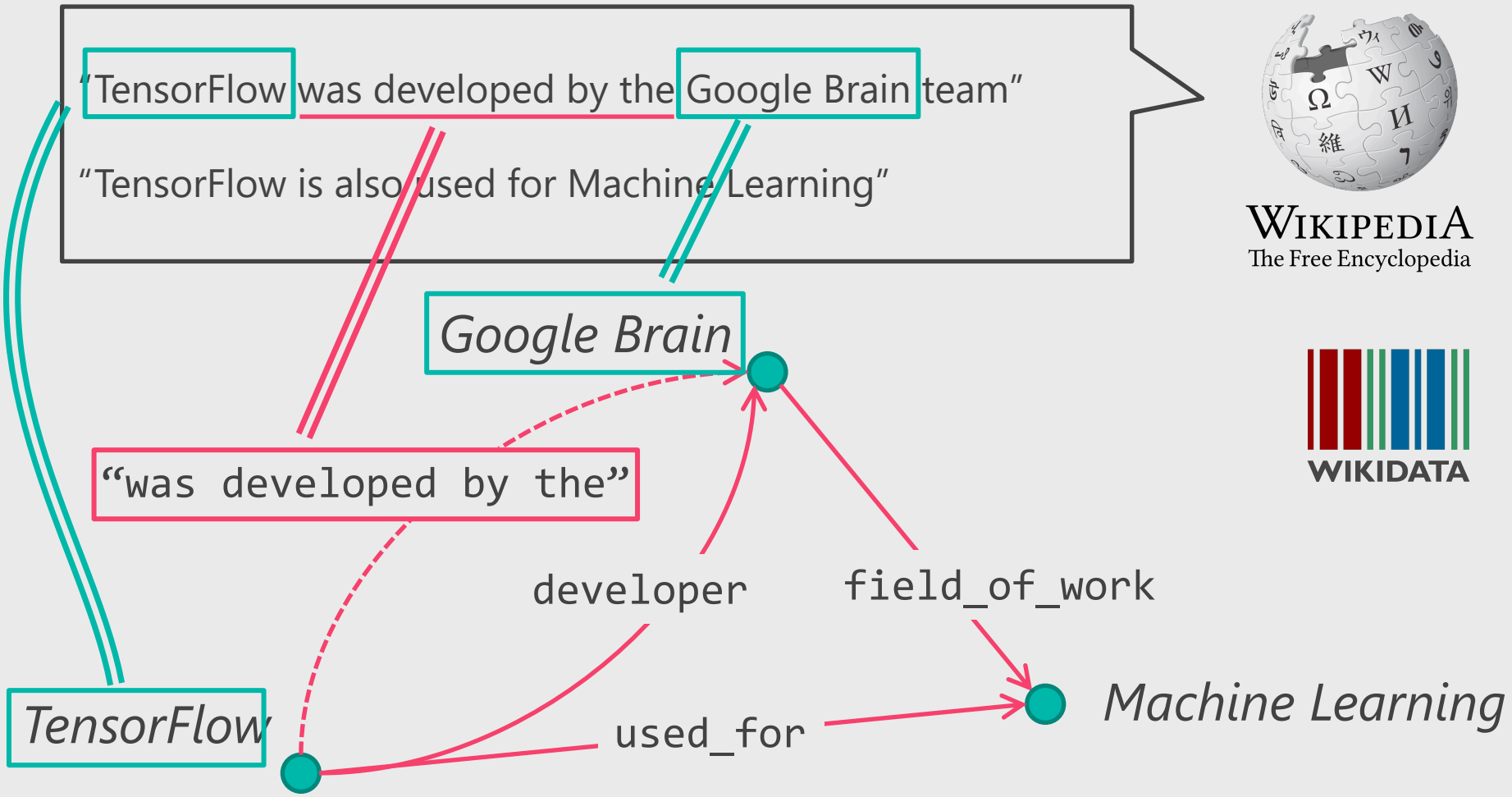
“TensorFlow was developed by the Google Brain team”

“TensorFlow is also used for Machine Learning”

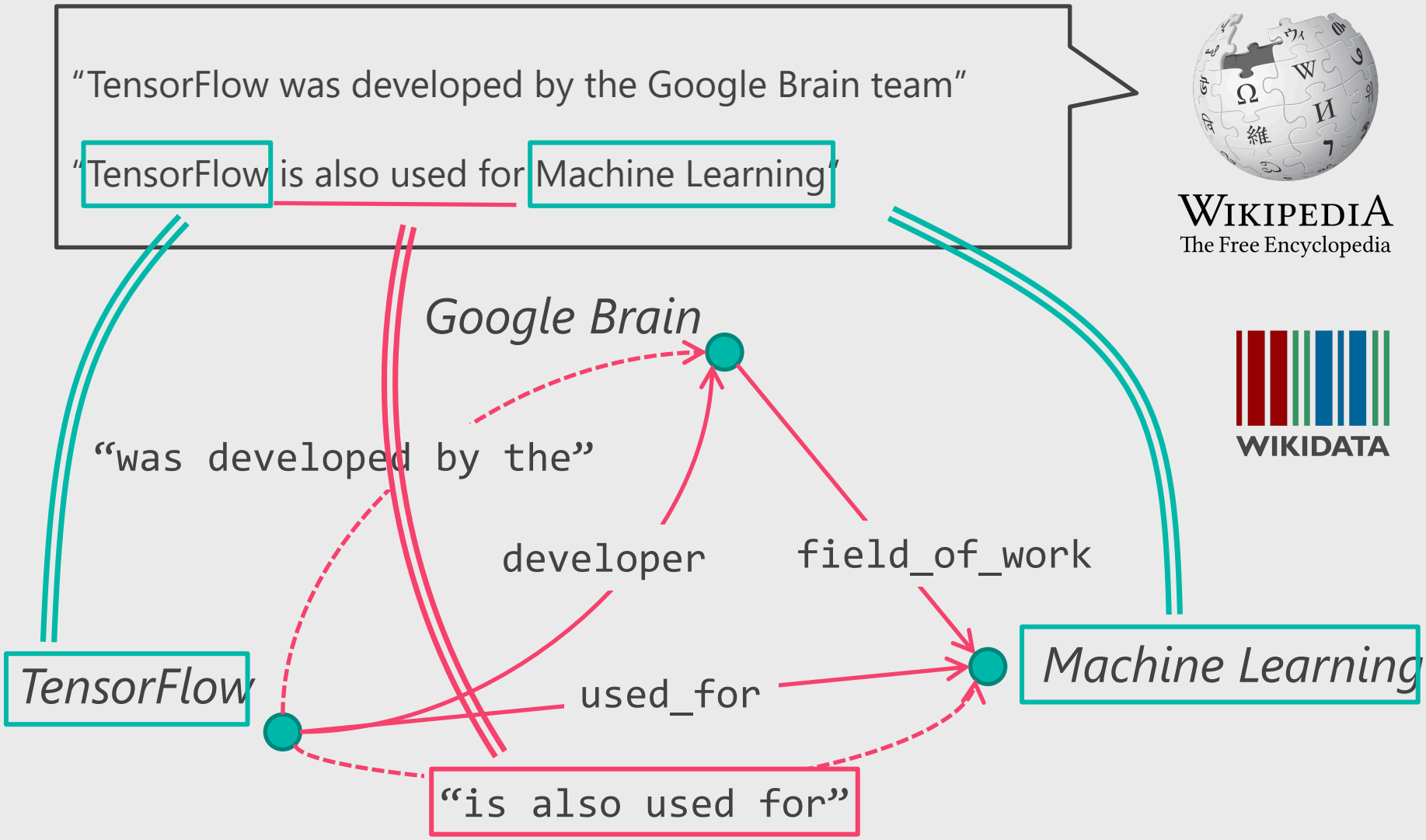
テキストコーパスから密なグラフを構築



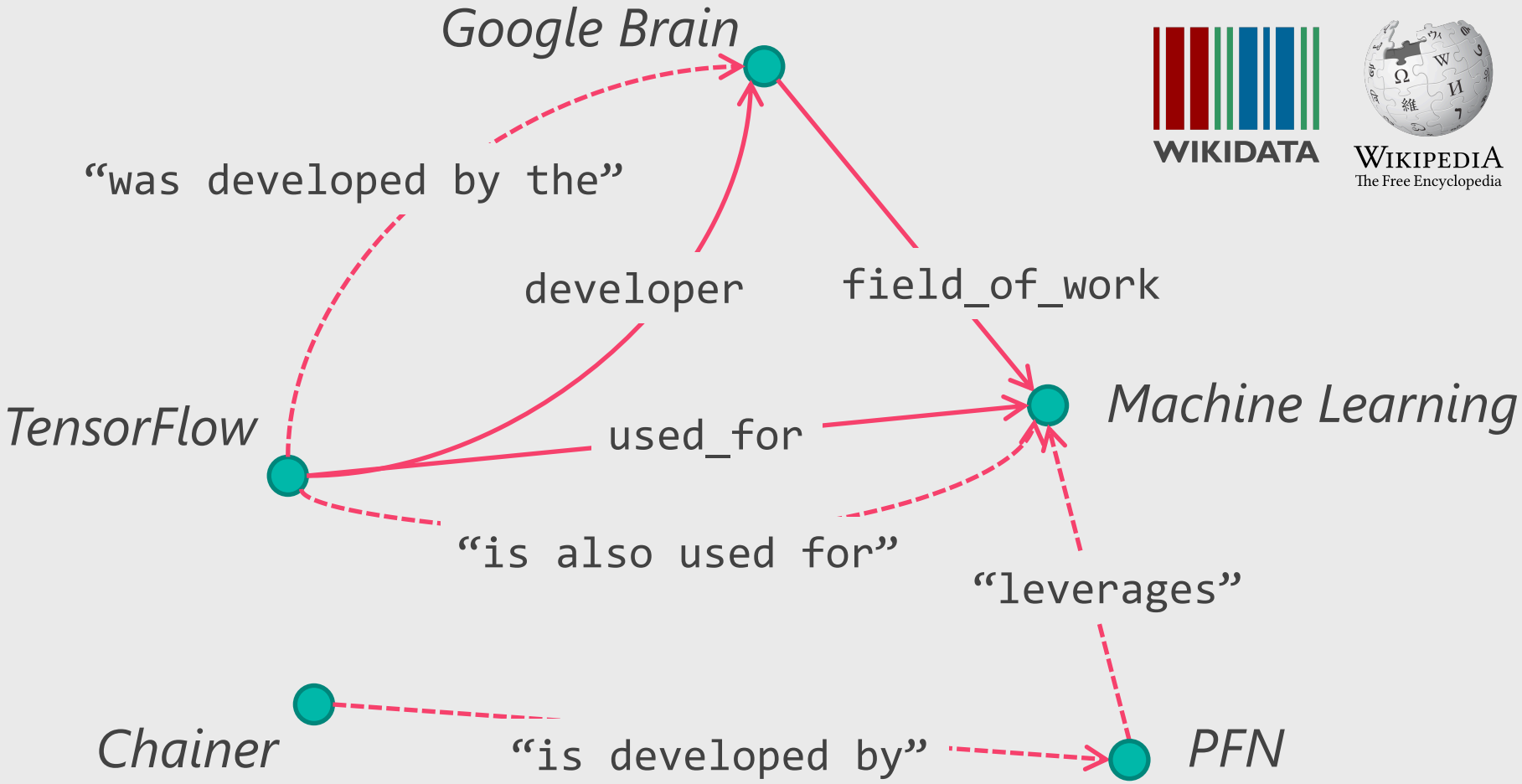
テキストコーパスから密なグラフを構築



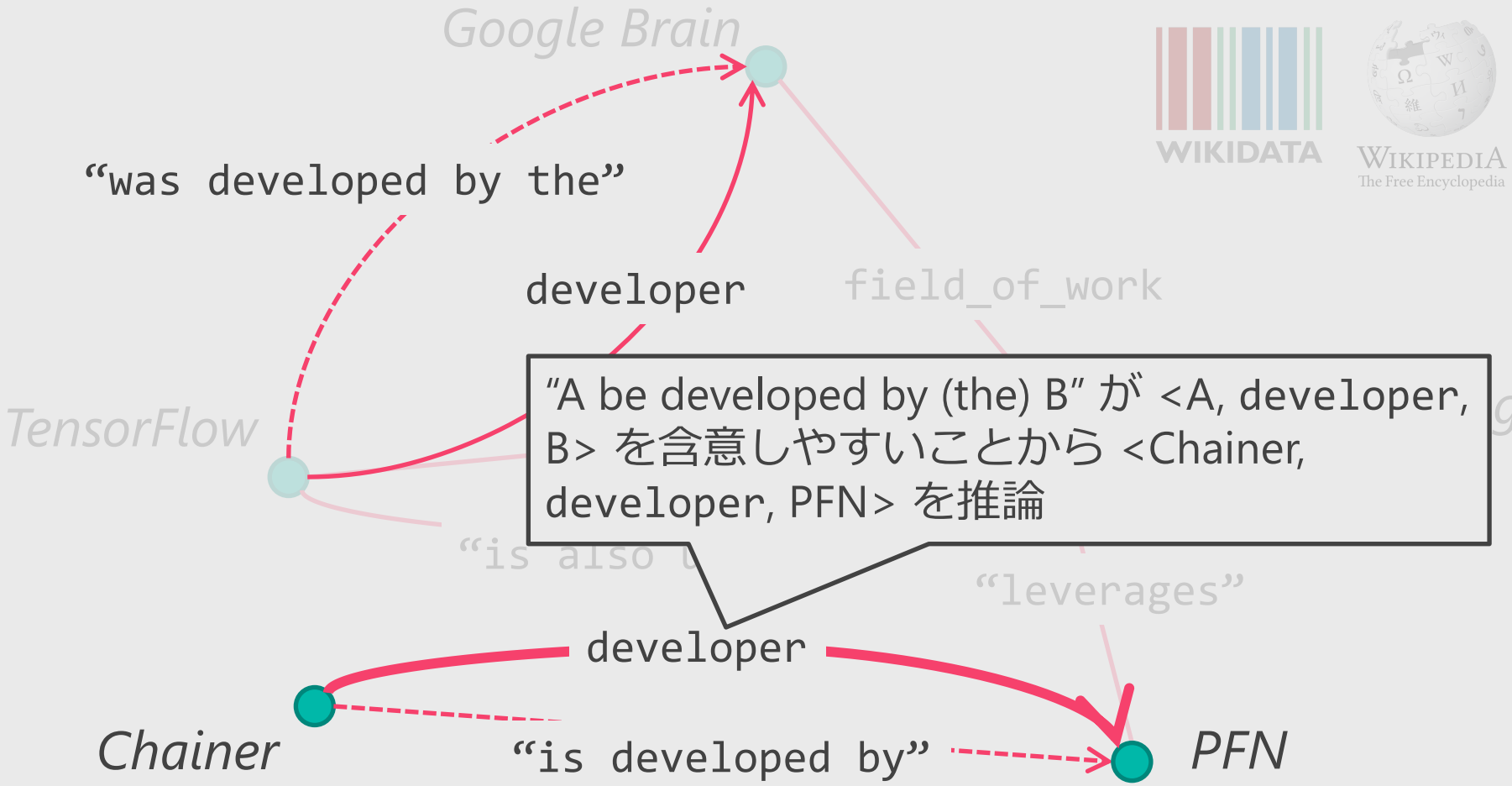
テキストコーパスから密なグラフを構築



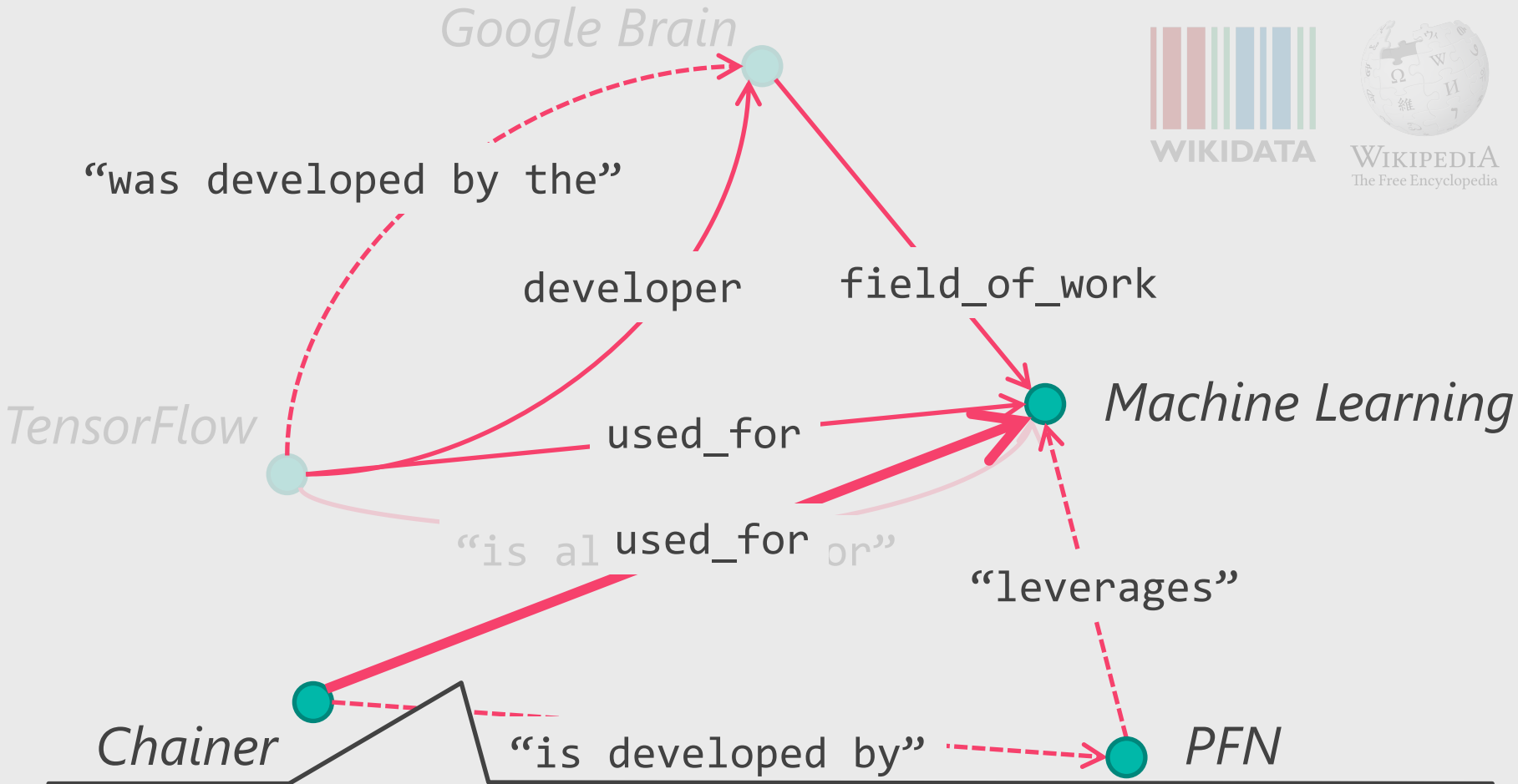
構造化データ（知識ベース）と非構造化データ（テキスト）の両方からなる密な “**Universal Graph**”



Universal Graph上でのシングルホップ推論例

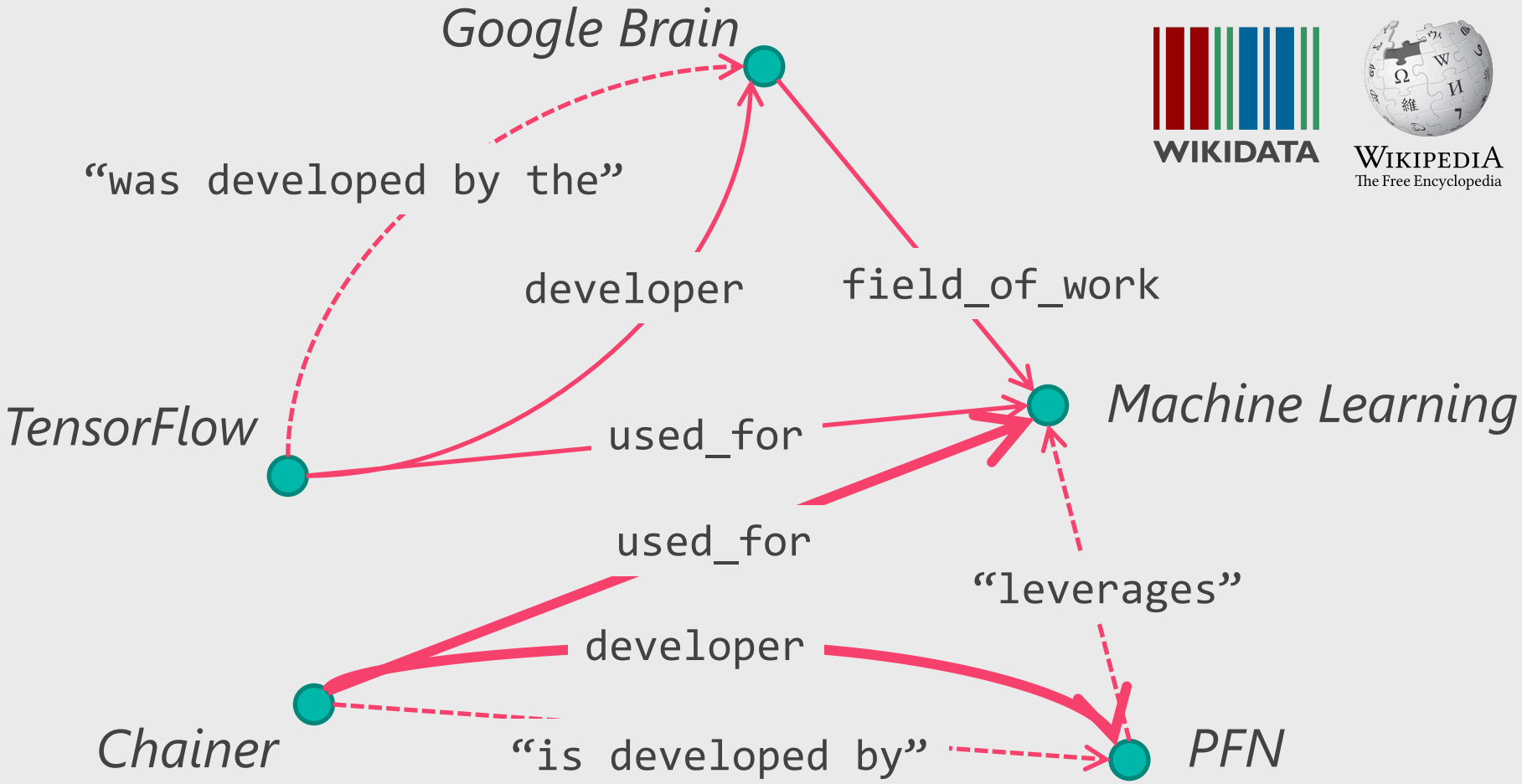


Universal Graph上でのマルチホップ推論例



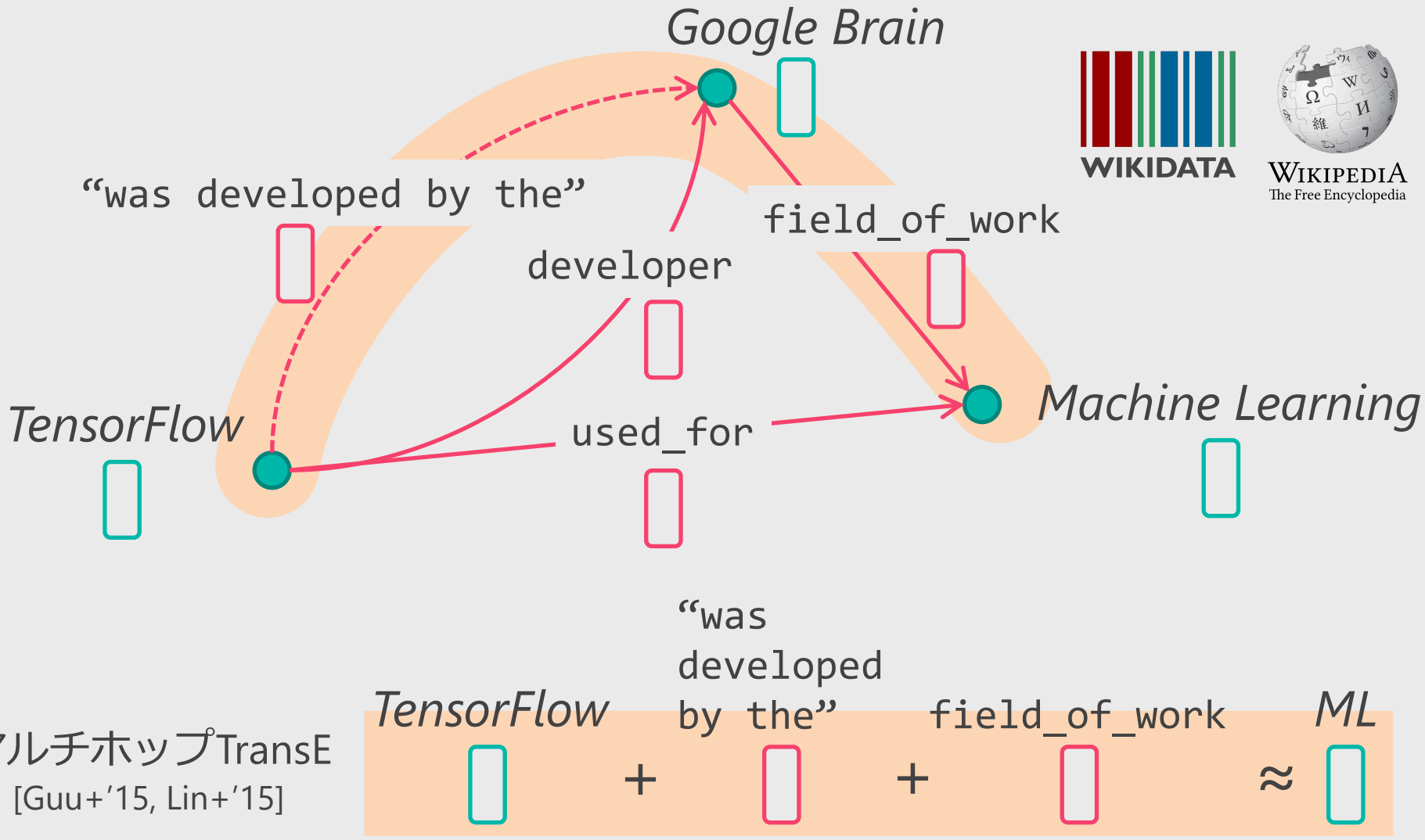
(1) 関係パス developer/field_of_work が used_for を含意しやすい
(2) “is developed by” ≈ developer, “leverages” ≈ field_of_work
ことから <*Chainer*, used_for, *Machine Learning*> を推論

知識ベースだけでは手がかりが得られない疎な部分の推論



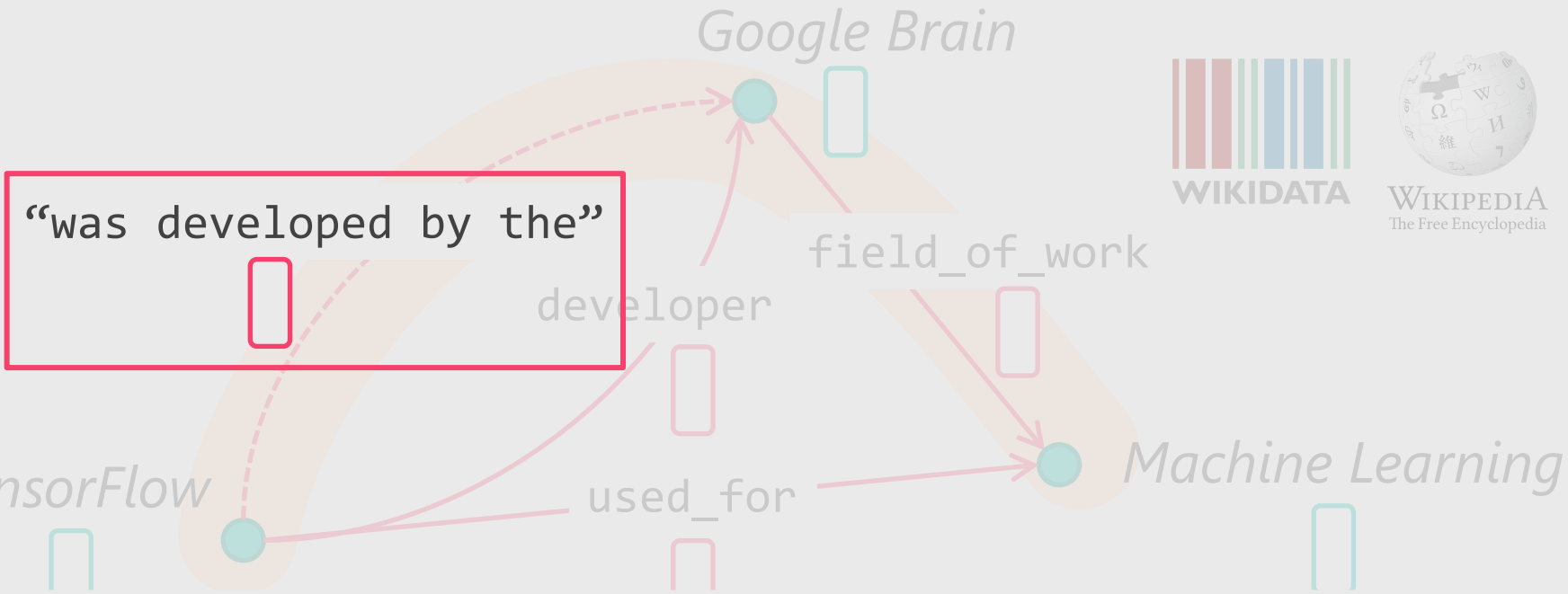
推論の実現方法：**Universal Graph上でのVector Based Model**

Universal Graph Embedding: Universal Graph上でのVector Based Model



マルチホップTransE
[Guu+’15, Lin+’15]

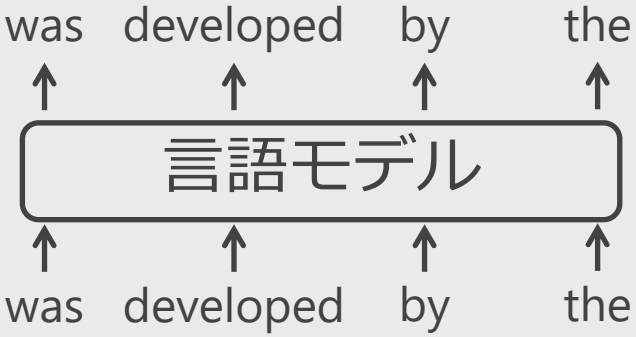
課題：Textual Relationをどのようにベクトルにするか？



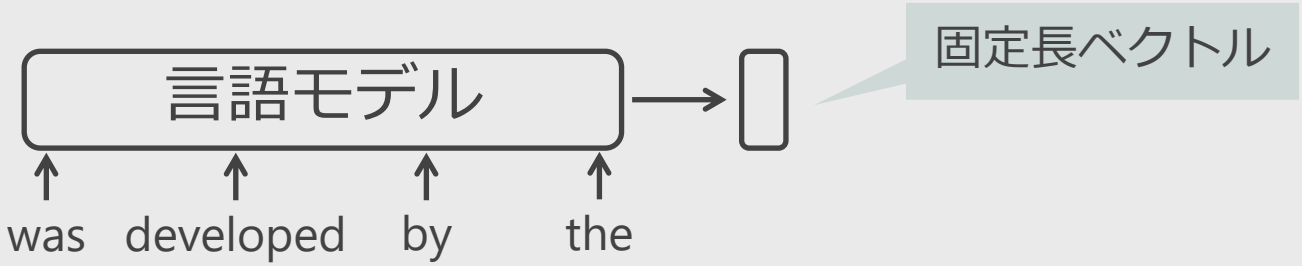
- Bad: 一つの表現に一つのベクトルを割り当てる
  - 関係を表す言語表現は多様
  - スパースでまともな表現を学習できない
- 本研究：**言語モデル**でエンコード

言語モデル

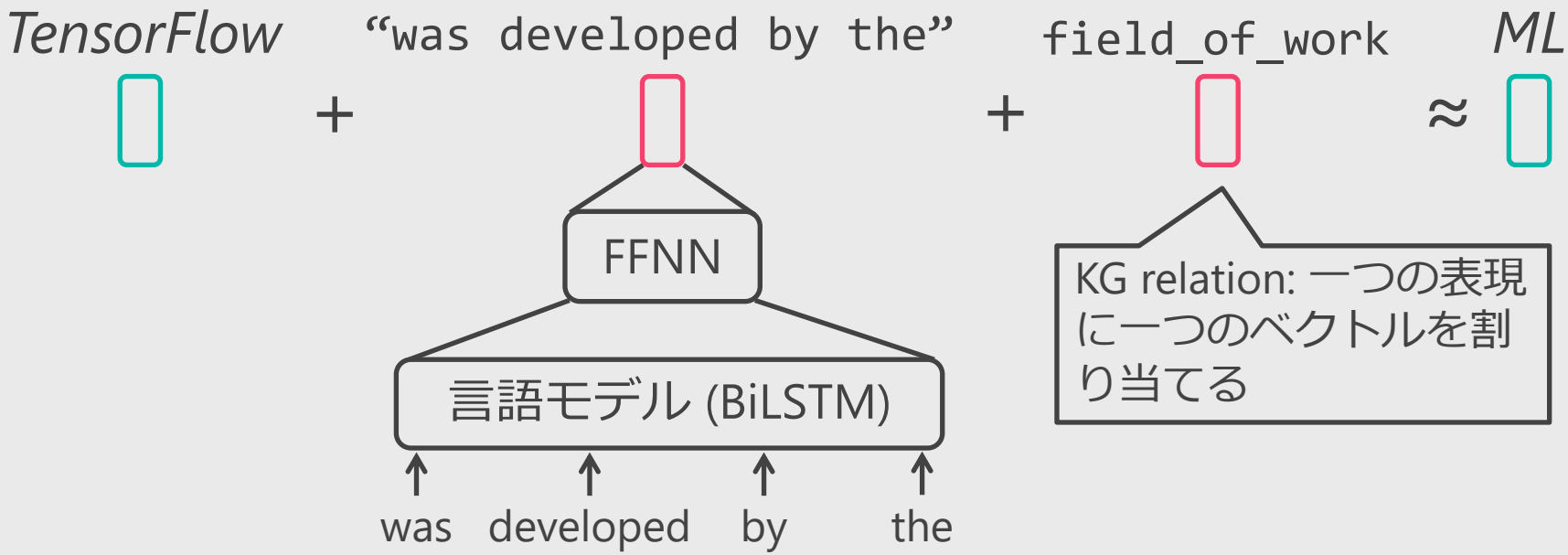
- 言語モデル：入力系列を復元するオートエンコーダ



- 大規模な（数千万文の）テキストコーパスで言語モデルを事前学習し，文分類など各タスクでファインチューニングする枠組みが流行
  - モデル例：ELMo, BERT, GPT-2, GPT-3, …
  - 画像分野におけるVGGNet, ImageNetなどに相当
- 単語列の素性抽出器として利用可能



Universal Graph Embeddingの学習と各コンポーネントの役割



- 言語モデル：textual relationを言語の世界での潜在的な表現に変換
  - 意味的に似た関係を表すtextual relationが似たベクトルになることを期待
- FFNN：言語の世界での潜在的な表現を知識ベースの表現に変換
  - Textual relationのエンコード結果のベクトルが，それと似た意味を表すKG relationのベクトル付近に射影されることを期待
  - “was developed by the” ≈ developer

本論文のResearch Questions

① **学習率のバランス**：異なる方法でエンコードされた関係のベクトルの更新量をどのようにバランスさせるか？

② **学習戦略**：textual relationを学習サイクルの中でどのように使っていくか？

- ノイジーなtextual relationをフルに使った学習だけで良いか？

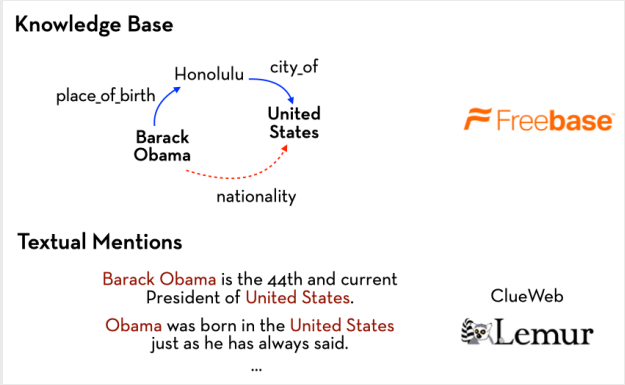
③ **言語モデルの事前学習**：テキストで表された関係 (textual relation) を事前に学習する効果はあるか？

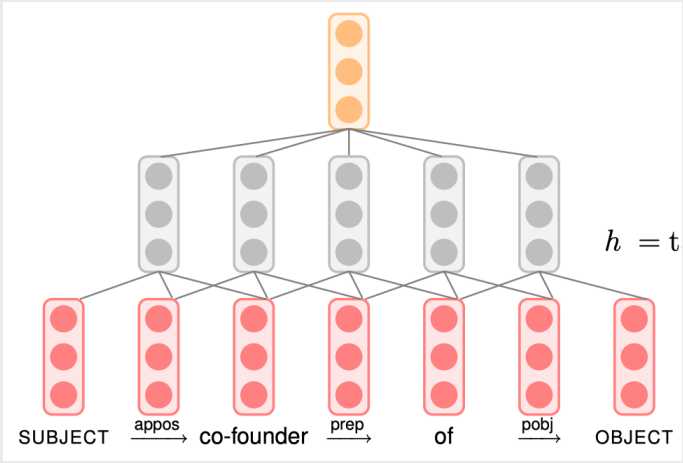
- （KG relation: 知識ベースで予め定義された関係）
- 似た意味を表すtextual relationのベクトルは予め似ているべき
  - “was developed by the” ≈ “is developed by”

④ **マルチホップ学習**：知識グラフとは性質が異なるUniversal Graphでのマルチホップ学習は効果があるか？

関連研究：Toutanova+’15

- アプローチ：Universal Graph上でのvector based model (Universal Graph Embedding) に該当
- 結果：知識ベース補完の性能をテキストがブースト
- 本研究のRQへの解はない
  - ① **学習率のバランス**
    - データ単位の重み付け
  - ② **学習戦略**
    - 記述なし
  - ③ **言語モデルの事前学習**
    - 非言語モデル
    - CNNによるエンコード
  - ④ **マルチホップ学習**
    - シングルホップのみ
- Toutanova+’15以降，Universal Graph Embeddingに関する研究は存在しない



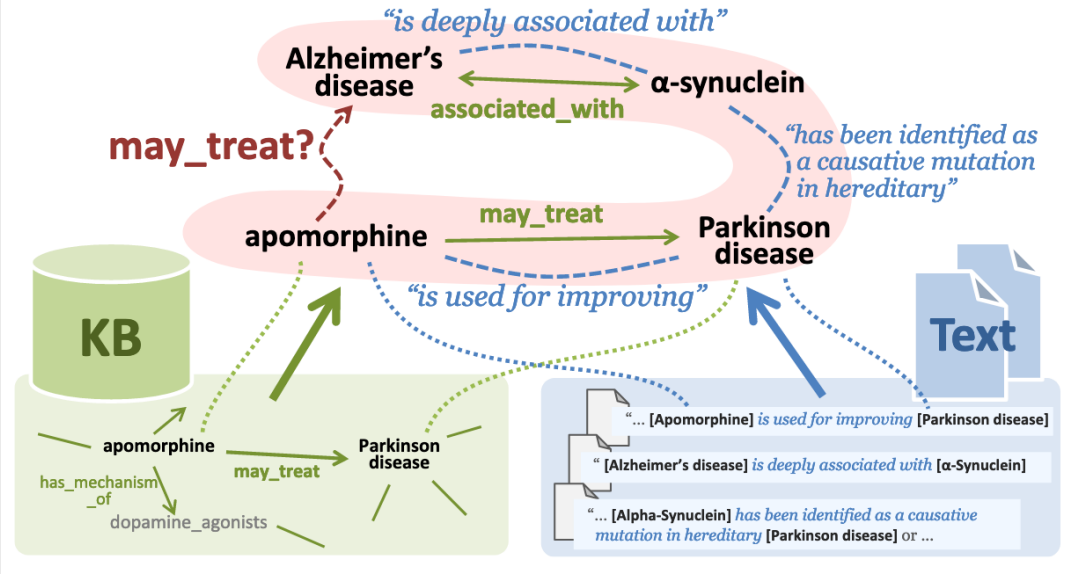


本研究：Universal Graph Embeddingへの様々な（基礎的な）洞察の提供

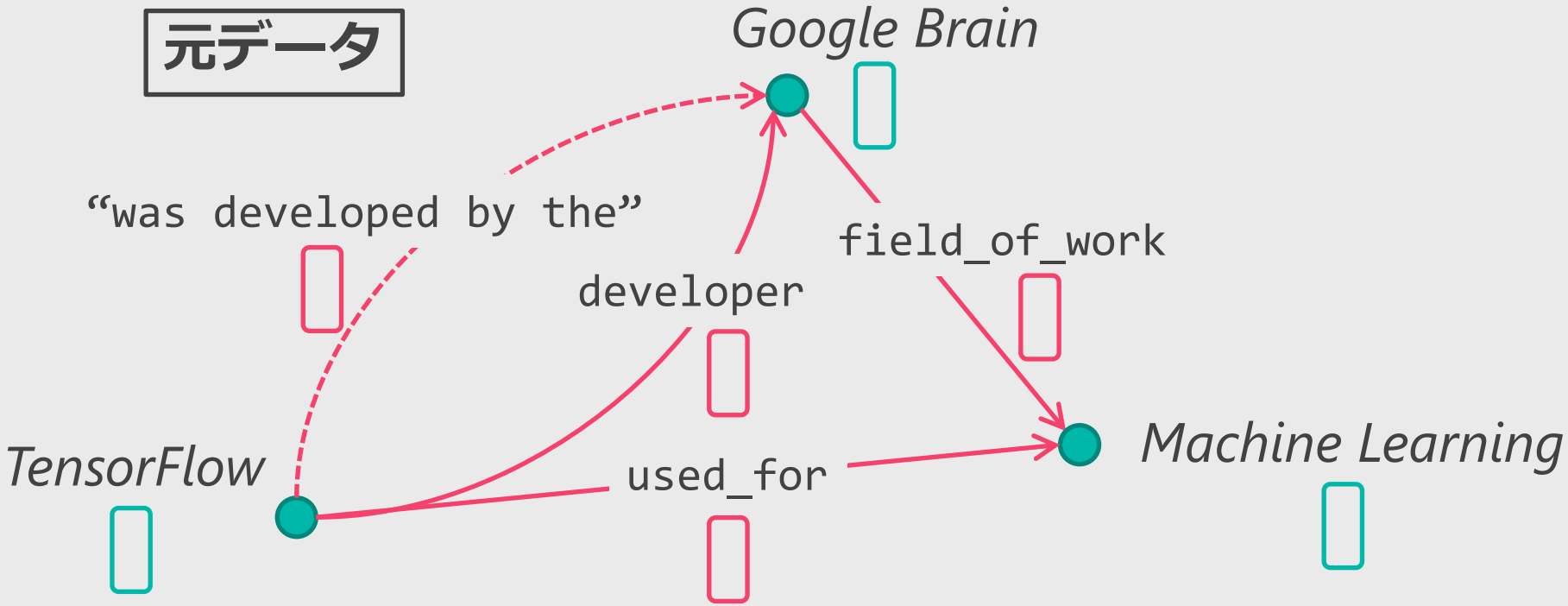
EXPERIMENTS

Universal Graphの構築

- 生物医学ドメインの知識ベース (UMLS) とテキストコーパス (MEDLINE) からUniversal Graphを構築
- UMLS：有用な関係知識を得るためフィルタリング（論文参照）
  - 約30万個のKG relation
- MEDLINE：フィルタリングされた知識ベース上のエンティティをつなぐtextual relationのみを抽出
  - 約5000万個のtextual relation

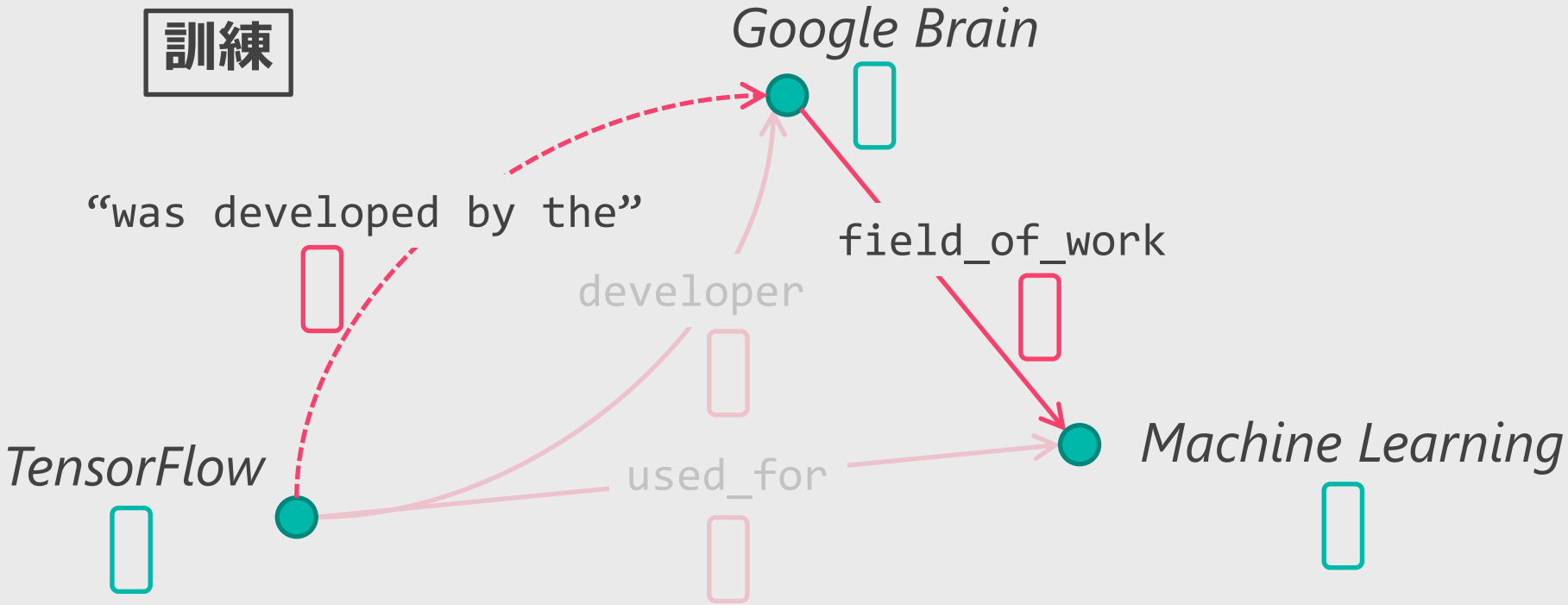


訓練データと評価データの分割



知識ベースの関係知識だけを評価データとして使う

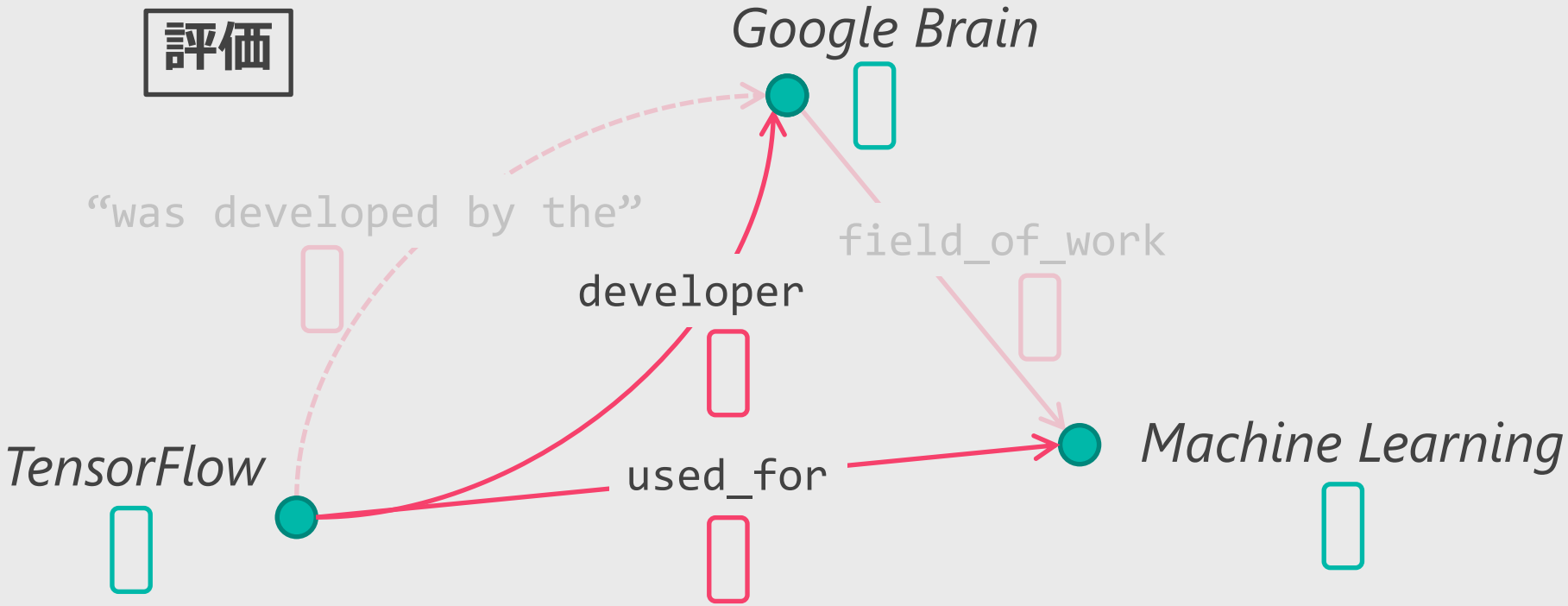
訓練データと評価データの分割



知識ベースの関係知識だけを評価データとして使う

（例：<TensorFlow, developer, Google Brain> と <TensorFlow, used_for, Machine Learning> を評価データにする）

訓練データと評価データの分割



知識ベースの関係知識だけを評価データとして使う

- Textual relationを追加の情報として有効に使えたか？を確かめる
- 例：Textual relationを使えていない場合，<TensorFlow, developer Google Brain>の予測は難しそう

評価方法：テール予測

<*Chainer*, developer, *?*>

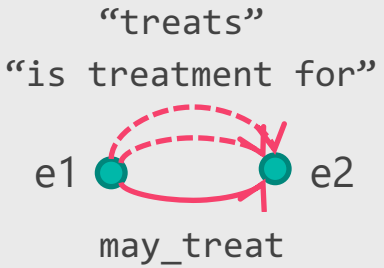
	テールエンティティ	スコア	順位
	Google Brain	5.0	1
✓	PFN	3.1	**2**
	Tohoku University	1.5	4
	Chainer	2.6	3

- すべてのエンティティについてスコア付け
  - TransE: $-\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|$
- ターゲットエンティティの順位に基づいて評価
  - 平均順位 (Mean Rank; MR)
  - 平均逆順位 (Mean Reciprocal Rank; MRR)
  - 10位以内の割合 (Hits at 10; H10)

評価用ベンチマークデータセットの構築

UMLSとMEDLINEから構築したUGのサブセット：

- Syntheticデータ
  - 条件「すべての評価対象のKG relationに，**対となるtextual relation**が必ず存在する」を満たすように辺を抽出
  - 「対となるtextual relationを正しくエンコードできているか？」を確かめる
- Naturalデータ
  - Syntheticデータの条件で辺を抽出した後，元のUniversal Graphからランダムに辺を追加
  - 「Textual relationとKG relationを組み合わせたマルチホップ推論ができるか？」を確かめる



		$\lvert\mathscr{E}\rvert$	$\lvert\mathscr{R}\rvert$	#Train	#Valid	#Test
Syntheticデータ	$\mathscr{T}_{\text{KG}}$	2.137	31	2,280	281	253
	$\mathscr{T}_{\text{text}}$	2,137	17,716	18,498	0	0
Naturalデータ	$\mathscr{T}_{\text{KG}}$	988	25	2,382	294	264
	$\mathscr{T}_{\text{text}}$	715	20,663	21,595	0	0

学習時のパスのサンプリング方法

- **KG-single (Ks)**: KGからシングルホップパスをサンプル
  - 通常の知識ベース補完の設定と同一
- **KG-multi (Km)**: KGからシングルホップパスとマルチホップパスをサンプル
  - [Guu+'15] と同一の設定
- **UG-single (Us)**: UGからシングルホップパスをサンプル
  - [Toutanova+'15] と同一の設定
- **UG-multi (Um)**: UGからシングルホップパスとマルチホップパスをサンプル
  - 本研究が初めて

本論文のResearch Questions

① **学習率のバランス**：異なる方法でエンコードされた関係のベクトルの更新量をどのようにバランスさせるか？

② **学習戦略**：textual relationを学習サイクルの中でどのように使っていくか？

- ノイジーなtextual relationをフルに使った学習だけで良いか？

③ **言語モデルの事前学習**：テキストで表された関係 (textual relation) を事前に学習する効果はあるか？

- （KG relation: 知識ベースで予め定義された関係）
- 似た意味を表すtextual relationのベクトルは予め似ているべき
  - “was developed by the” ≈ “is developed by”

④ **マルチホップ学習**：知識グラフとは性質が異なるUniversal Graphでのマルチホップ学習は効果があるか？

RQ1: 学習率のバランス

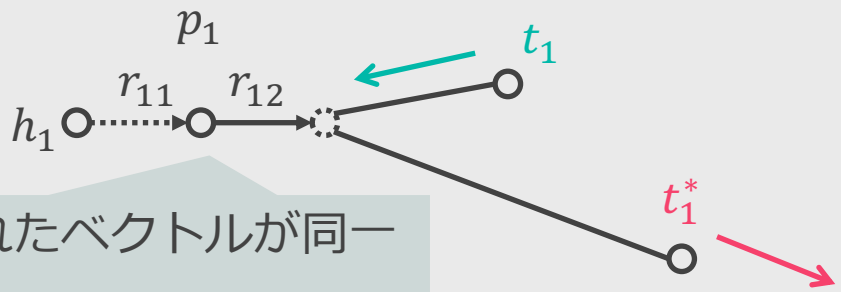
- Max-margin loss:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} \sum_{t_1^* \in \mathcal{N}(h_i, p_i)} \max(0, [\gamma + f^- - f^+])$$

$$f^- = f(h_i, p_i, t_1^*; \Theta), \qquad f^+ = f(h_i, p_i, t_i; \Theta)$$

負例のスコア　　正例のスコア

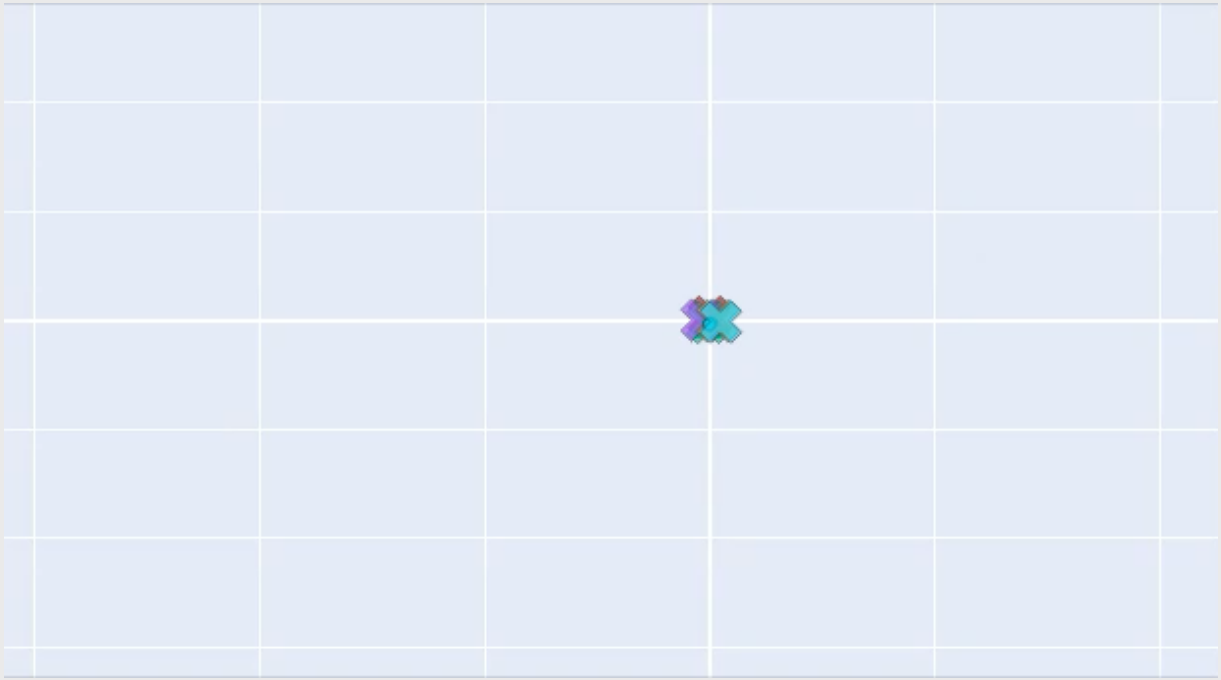
  - γ: マージン
    - モデルが許容する正例と負例の間の最小の距離



異なる方法で計算されたベクトルが同一の損失を受け取る

=> スケールを合わせなければ**学習が不安定**になる

スケールを合わせない場合 (UG-single)

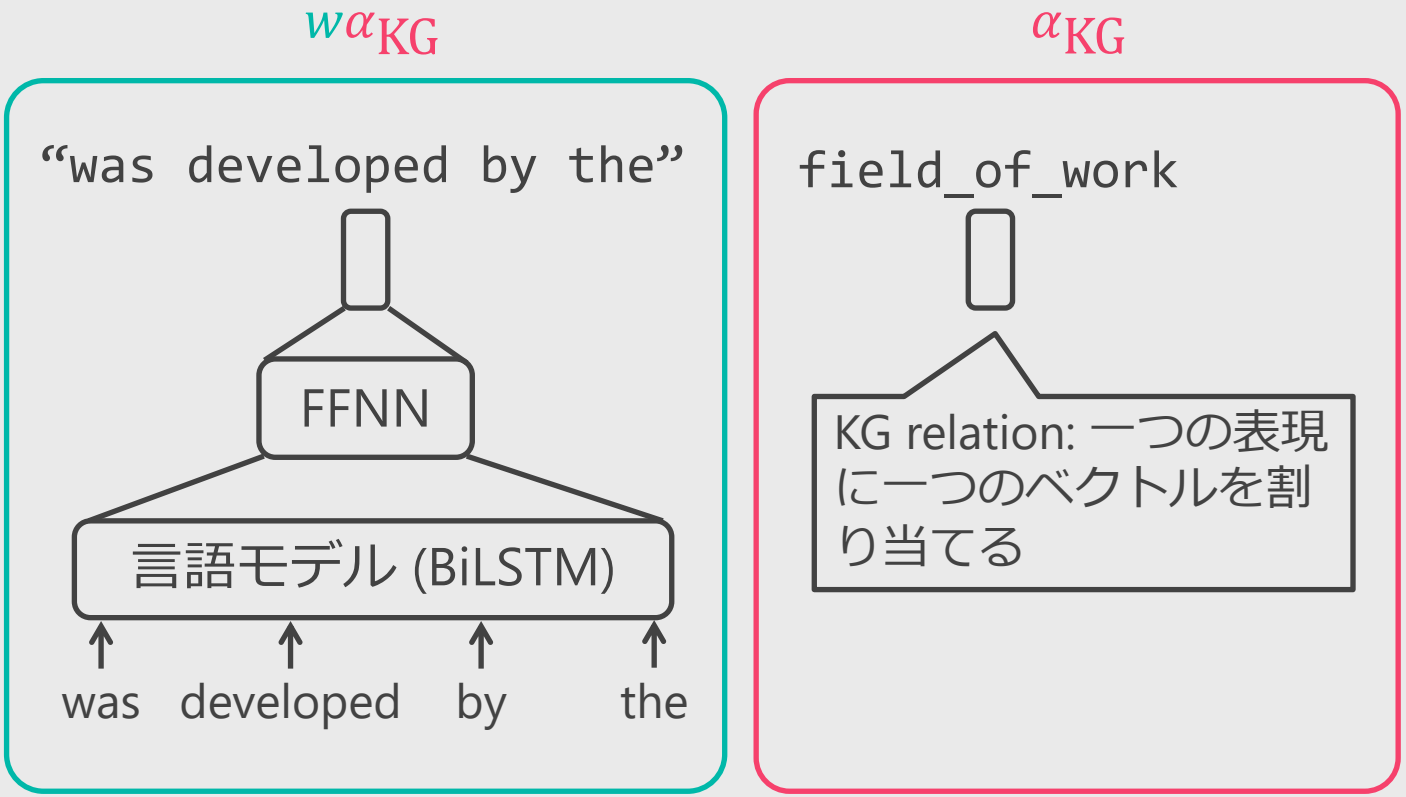


●: textual relation

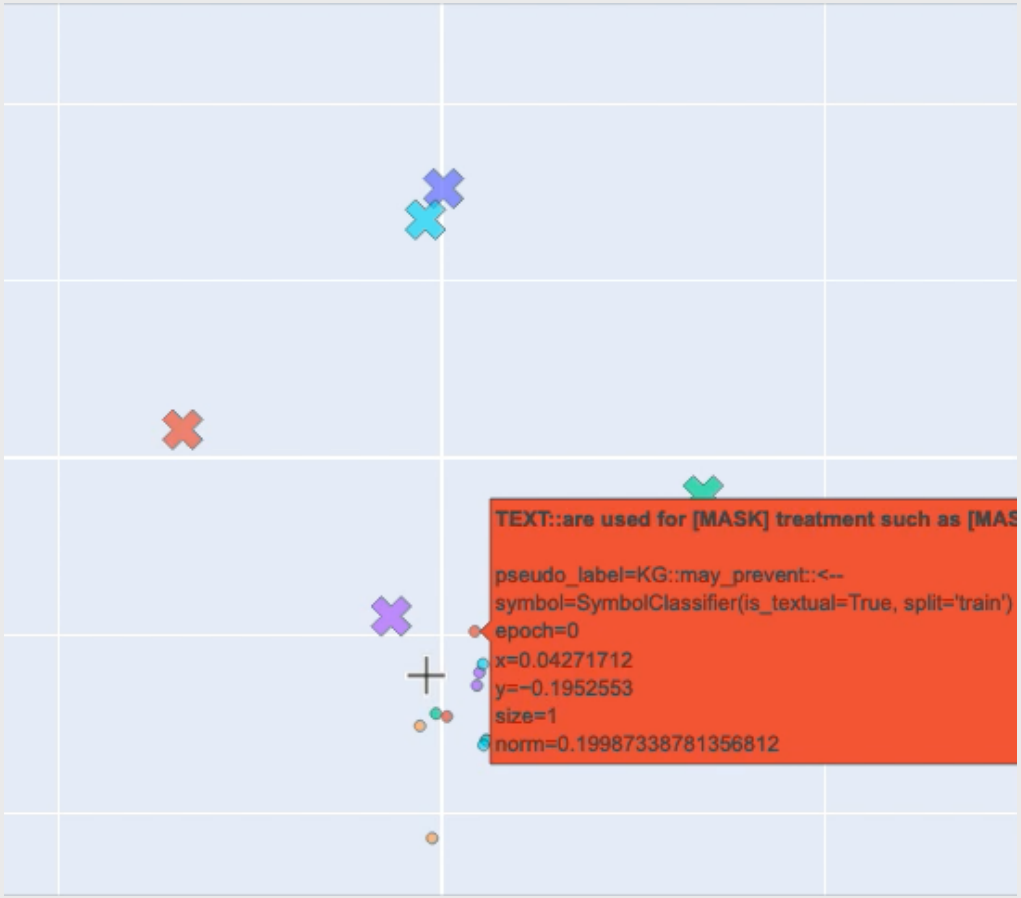
✕: KG relation

Textual relationのベクトルがKG relationの外側で大きく振動を続ける

モデルの部分ごとに別々の初期学習率を設定



スケールを合わせることで学習が安定化



●: textual relation

✕: KG relation

w	MRR
10^{1}	0.184 ± 0.161
10^{0}	0.256 ± 0.104
10^{-1}	0.319 ± 0.010
10^{-2}	$\mathbf{0.329 \pm 0.039}$
10^{-3}	0.291 ± 0.023

wの値はSyntheticデータで調整

✓RQ1: 学習率のバランスは全てのデータセット・設定で有効

本論文のResearch Questions

① **学習率のバランス**：異なる方法でエンコードされた関係のベクトルの更新量をどのようにバランスさせるか？

② **学習戦略**：textual relationを学習サイクルの中でどのように使っていくか？

- ノイジーなtextual relationをフルに使った学習だけで良いか？

③ **言語モデルの事前学習**：テキストで表された関係 (textual relation) を事前に学習する効果はあるか？

- （KG relation: 知識ベースで予め定義された関係）
- 似た意味を表すtextual relationのベクトルは予め似ているべき
  - “was developed by the” ≈ “is developed by”

④ **マルチホップ学習**：知識グラフとは性質が異なるUniversal Graphでのマルチホップ学習は効果があるか？

RQ2: Naturalデータでの学習戦略の実験

- Textual relationはノイジー
  - 文中に共起する二つのエンティティは必ずしも特定の関係性を持たない
  - 二つのエンティティが偶然共起するだけでUniversal Graphに追加

- 直感：Universal Graphから大まかに学習した後，Knowledge Graphで精緻な調整をしたほうが良さそう

- 二つの戦略を実験的に比較
  - Noisy-to-clean:「UG -> KG」の順に学習を進める
  - Clean-to-noisy:「KG -> UG」の順に学習を進める

RQ2: Naturalデータでの学習戦略の実験結果

Model	LM-pret	検証 MR	検証 MRR	検証 H10	評価 MR	評価 MRR	評価 H10
Noisy-to-clean							
Um		102.6	0.089	0.162	96.1	0.061	0.148
Um-Us		73.8	0.150	0.313	82.4	0.104	0.258
Um-Us-Km		55.2	0.164	0.389	58.8	0.127	0.328
Um-Us-Km-Ks		55.8	0.166	0.383	59.8	0.129	0.337
Um	✓	91.8	0.089	0.184	84.1	0.075	0.184
Um-Us	✓	71.9	0.150	0.313	65.0	0.119	0.267
Um-Us-Km	✓	57.4	0.168	0.379	53.8	0.131	0.331
Um-Us-Km-Ks	✓	58.7	0.169	0.405	55.1	0.139	0.345
Us	✓	75.5	0.155	0.330	77.8	0.122	0.273
Us-Km	✓	61.2	0.173	0.371	62.7	0.137	0.318
Us-Km-Ks	✓	63.1	0.178	0.374	63.2	0.139	0.331
Clean-to-noisy							
Ks		115.9	0.123	0.255	106.8	0.115	0.267
Ks-Km		76.6	0.140	0.274	70.6	0.126	0.267
Ks-Km-Us		77.3	0.145	0.301	75.3	0.115	0.265
Ks-Km-Us-Um		76.9	0.144	0.298	74.9	0.115	0.263

RQ2: **学習戦略**は**効果あり**

- 「UG -> KG」の順に学習を進めることで**一貫して性能を改善**
- Knowledge Graph単体だけで学習するよりも最終的な精度が高くなる

検証 MR	検証 MRR	検証 H10
115.9	0.123	0.255

- => データとモデルの大規模化を進めれば，「事前学習-ファインチューニング」パラダイムが知識ベース埋め込みの分野でも有効であることを示唆

RQ2: Naturalデータでの学習戦略の実験結果

		検証			評価		
Model	LM-pret	MR	MRR	H10	MR	MRR	H10
Noisy-to-clean							
Um		102.6	0.089	0.162	96.1	0.061	0.148
Um-Us		73.8	0.150	0.313	82.4	0.104	0.258
Um-Us-Km		55.2	0.164	0.389	58.8	0.127	0.328
Um-Us-Km-Ks		55.8	0.166	0.383	59.8	0.129	0.337
Um	✓	91.8	0.089	0.184	84.1	0.075	0.184
Um-Us	✓	71.9	0.150	0.313	65.0	0.119	0.267
Um-Us-Km	✓	57.4	0.168	0.379	53.8	0.131	0.331
Um-Us-Km-Ks	✓	58.7	0.169	0.405	55.1	0.139	0.345
Us	✓	75.5	0.155	0.330	77.8	0.122	0.273
Us-Km	✓	61.2	0.173	0.371	62.7	0.137	0.318
Us-Km-Ks	✓	63.1	0.178	0.374	63.2	0.139	0.331
Clean-to-noisy							
Ks		115.9	0.123	0.255	106.8	0.115	0.267
Ks-Km		76.6	0.140	0.274	70.6	0.126	0.267
Ks-Km-Us		77.3	0.145	0.301	75.3	0.115	0.265
Ks-Km-Us-Um		76.9	0.144	0.298	74.9	0.115	0.263

「Noisy-to-clean」戦略は精度が頭打ち

=> UGのノイズの影響は無視できない

本論文のResearch Questions

① **学習率のバランス**：異なる方法でエンコードされた関係のベクトルの更新量をどのようにバランスさせるか？

② **学習戦略**：textual relationを学習サイクルの中でどのように使っていくか？

- ノイジーなtextual relationをフルに使った学習だけで良いか？

③ **言語モデルの事前学習**：テキストで表された関係 (textual relation) を事前に学習する効果はあるか？

- （KG relation: 知識ベースで予め定義された関係）
- 似た意味を表すtextual relationのベクトルは予め似ているべき
  - “was developed by the” ≈ “is developed by”

④ **マルチホップ学習**：知識グラフとは性質が異なるUniversal Graphでのマルチホップ学習は効果があるか？

RQ3: 言語モデルの事前学習

- MEDLINEから得られた100kのtextual relationで言語モデルを訓練

Synthetic

Model	LM-pret	検証 MR	検証 MRR	検証 H10	評価 MR	評価 MRR	評価 H10
Ks		287.3	0.167	0.311	305.1	0.155	0.300
Km		256.3	0.178	0.342	265.0	0.160	0.330
Us		51.6	0.383	0.699	54.3	0.346	0.617
Us	✓	43.0	0.377	0.698	45.9	0.342	0.617
Um		65.7	0.368	0.683	61.2	0.338	0.638
Um	✓	70.5	0.357	0.694	66.3	0.305	0.642

Natural

Model	LM-pret	検証 MR	検証 MRR	検証 H10	評価 MR	評価 MRR	評価 H10
Ks		115.9	0.123	0.255	106.8	0.115	0.267
Km		76.7	0.141	0.274	70.6	0.125	0.269
Us		73.1	0.138	0.282	73.5	0.118	0.261
Us	✓	75.5	0.155	0.330	77.8	0.122	0.273
Um		102.6	0.089	0.162	96.1	0.061	0.148
Um	✓	91.8	0.089	0.184	84.1	0.075	0.184

言語モデルの事前学習

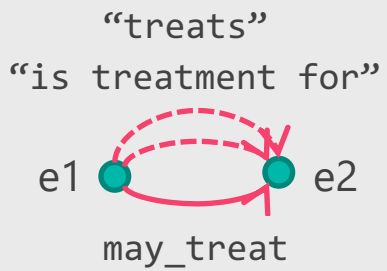
Natural

Model	LM-pret	検証 MR	検証 MRR	検証 H10	評価 MR	評価 MRR	評価 H10
Noisy-to-clean							
Um		102.6	0.089	0.162	96.1	0.061	0.148
Um-Us		73.8	0.150	0.313	82.4	0.104	0.258
Um-Us-Km		55.2	0.164	0.389	58.8	0.127	0.328
Um-Us-Km-Ks		55.8	0.166	0.383	59.8	0.129	0.337
Um	✓	91.8	0.089	0.184	84.1	0.075	0.184
Um-Us	✓	71.9	0.150	0.313	65.0	0.119	0.267
Um-Us-Km	✓	57.4	0.168	0.379	53.8	0.131	0.331
Um-Us-Km-Ks	✓	58.7	0.169	0.405	55.1	0.139	0.345
Us	✓	75.5	0.155	0.330	77.8	0.122	0.273
Us-Km	✓	61.2	0.173	0.371	62.7	0.137	0.318
Us-Km-Ks	✓	63.1	0.178	0.374	63.2	0.139	0.331
Clean-to-noisy							
Ks		115.9	0.123	0.255	106.8	0.115	0.267
Ks-Km		76.6	0.140	0.274	70.6	0.126	0.267
Ks-Km-Us		77.3	0.145	0.301	75.3	0.115	0.265
Ks-Km-Us-Um		76.9	0.144	0.298	74.9	0.115	0.263

言語モデルの事前学習の有無で精度に変化は見られない

RQ3: 言語モデルの事前学習の実際の効果

- 関係may_treatを表すtextual relationのベクトルの周辺は似たものが集まっているか
  - 詳細：関係may_treatと対となるtextual relationのベクトルそれぞれに対し，最近傍ベクトルを10件取得．そのうち何件がmay_treatと対になっているかの割合の平均を算出



言語モデル事前学習	UGの学習	Acc
		0.117
✓		0.149
	✓	0.201
✓	✓	0.201

言語モデルの事前学習には一定の効果が認められるものの，その後のUniversal Graphの学習が支配的

=> Textual relationのために設計されたエンコーダが必要

本論文のResearch Questions

① **学習率のバランス**：異なる方法でエンコードされた関係のベクトルの更新量をどのようにバランスさせるか？

② **学習戦略**：textual relationを学習サイクルの中でどのように使っていくか？

- ノイジーなtextual relationをフルに使った学習だけで良いか？

③ **言語モデルの事前学習**：テキストで表された関係 (textual relation) を事前に学習する効果はあるか？

- （KG relation: 知識ベースで予め定義された関係）
- 似た意味を表すtextual relationのベクトルは予め似ているべき
  - “was developed by the” ≈ “is developed by”

④ **マルチホップ学習**：知識グラフとは性質が異なるUniversal Graphでのマルチホップ学習は効果があるか？

RQ4: マルチホップ学習

Synthetic

		検証			評価		
Model	LM-pret	MR	MRR	H10	MR	MRR	H10
Ks		287.3	0.167	0.311	305.1	0.155	0.300
Km		256.3	0.178	0.342	265.0	0.160	0.330
Us		51.6	0.383	0.699	54.3	0.346	0.617
Us	✓	43.0	0.377	0.698	45.9	0.342	0.617
Um		65.7	0.368	0.683	61.2	0.338	0.638
Um	✓	70.5	0.357	0.694	66.3	0.305	0.642

Natural

		検証			評価		
Model	LM-pret	MR	MRR	H10	MR	MRR	H10
Ks		115.9	0.123	0.255	106.8	0.115	0.267
Km		76.7	0.141	0.274	70.6	0.125	0.269
Us		73.1	0.138	0.282	73.5	0.118	0.261
Us	✓	75.5	0.155	0.330	77.8	0.122	0.273
Um		102.6	0.089	0.162	96.1	0.061	0.148
Um	✓	91.8	0.089	0.184	84.1	0.075	0.184

- 学習時にKnowledge Graphだけからパスをサンプリングする場合，**マルチホップ学習は性能向上に貢献**
  - Textual relationは使わない
- [Guu+'15, Takahashi+'18]などと一貫する結果

RQ4: マルチホップ学習

Synthetic

		検証			評価		
Model	LM-pret	MR	MRR	H10	MR	MRR	H10
Ks		287.3	0.167	0.311	305.1	0.155	0.300
Km		256.3	0.178	0.342	265.0	0.160	0.330
Us		51.6	0.383	0.699	54.3	0.346	0.617
Us	✓	43.0	0.377	0.698	45.9	0.342	0.617
Um		65.7	0.368	0.683	61.2	0.338	0.638
Um	✓	70.5	0.357	0.694	66.3	0.305	0.642

Natural

		検証			評価		
Model	LM-pret	MR	MRR	H10	MR	MRR	H10
Ks		115.9	0.123	0.255	106.8	0.115	0.267
Km		76.7	0.141	0.274	70.6	0.125	0.269
Us		73.1	0.138	0.282	73.5	0.118	0.261
Us	✓	75.5	0.155	0.330	77.8	0.122	0.273
Um		102.6	0.089	0.162	96.1	0.061	0.148
Um	✓	91.8	0.089	0.184	84.1	0.075	0.184

- 学習時にUniversal Graphからパスをサンプリングする場合，**マルチホップ学習は性能向上に貢献しない**
- => Universal Graphのノイズの影響

Universal Graphのマルチホップのノイズ

- 評価インスタンスを，訓練データでの最短経路長に基づいて分類
  - 各分類ごとにMRRを算出

		最短経路長		
Model	LM-pret	1	2	3 以上
Ks		0.175	0.184	0.041
Km		0.205	0.220	0.038
Us		0.226	0.149	0.041
Us	✓	0.256	0.179	0.037
Um		0.130	0.096	0.043
Um	✓	0.123	0.102	0.048
Um-Us-Km-Ks	✓	0.292	0.162	0.041
Us-Km-Ks	✓	0.317	0.157	0.040

- 最短経路長が3以上の場合，まともな予測ができない
  - Textual relationが三つ連なる経路が最も多い
  - Textual relationそのもののノイズと，マルチホップによるコンテキストの捨象のため，極めて困難な問題設定
  - => パスの選択機構やコンテキストをグローバルに計算する機構

第４章の実験まとめ

- Universal Graph Embeddingという枠組みが知識ベース補完の精度向上に有用であることを再確認（Toutanova+’15と一貫）
- 四つのResearch Questionsにフォーカスをあてた
  - ① **学習率のバランス**
    - モデルの異なる部分に異なる初期学習率を設定する戦略が有効
    - 全てのデータセットで効果あり
  - ② **学習戦略**
    - 「Noisy-to-clean」戦略が効果あり
    - 「事前学習-ファインチューニング」パラダイムの一つと見ることができる
  - ③ **言語モデルの事前学習**
    - 精度に直接的な影響は見られない
    - Textual relationのためのエンコーダが必要
  - ④ **マルチホップ学習**
    - 性能向上に貢献しない
    - ノイズへの対処のため，パスの選択機構やコンテキストを捨象しないことが必要

本論文のまとめと今後の展望

- 関係知識上でのマルチホップ推論について，「述語論理」と「命題論理」という二つの問題設定における機械学習アプローチを論じた
- 三つ組のテールを予測する「知識ベース補完」という，関係推論を単純化した問題設定においても，解決すべき課題が山積
- 構造化データと非構造化データの統合という観点は「知識ベース補完」にとどまらず，あらゆる問題設定で有効
- 実験設定をより現実に近づけていく努力が業界全体で必要

APPENDIX

Key idea

Combine **logical inference** and **physical simulation**

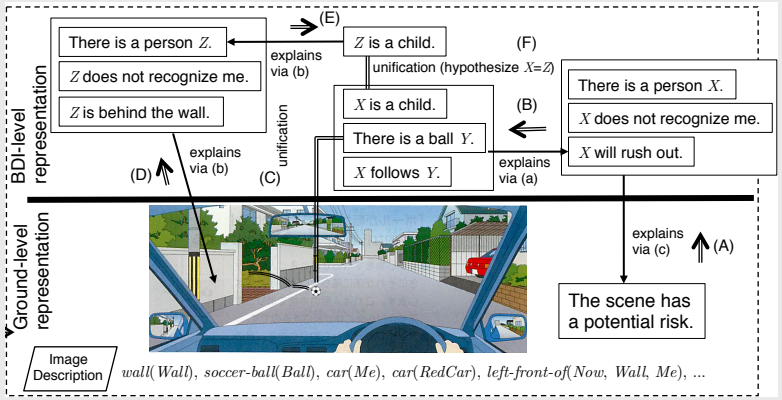
Logical Inference

(apply symbolic rules to a scene description and infer new facts)
[Armand+ 14, Inoue+ 15, Mohammad+ 15, Zhao+ 15, etc.]

PROS: Easy to take causal chain into account

PROS: Can combine human writing knowledge

CONS: The prediction is not based on a precise physical prediction



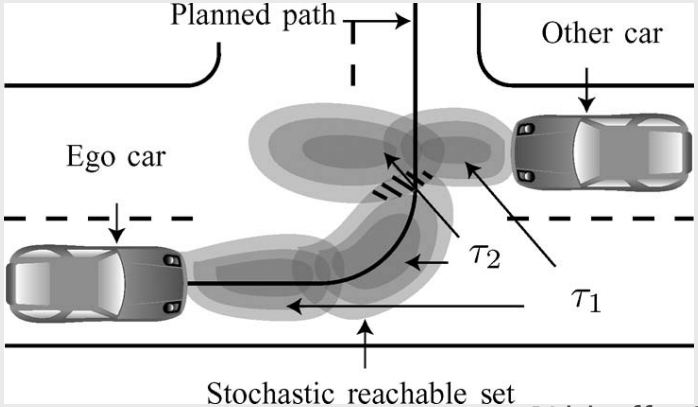
[Inoue+ 15]

Physical Simulation

(simulate a scene and detect future collisions)
[Broadhurst+ 05, Althoff+ 09, etc.]

PROS: Precise prediction grounded in physical quantities

CONS: Difficult to take causal chain into account



[Althoff+ 09]

Key idea

Combine **logical inference** and **physical simulation**

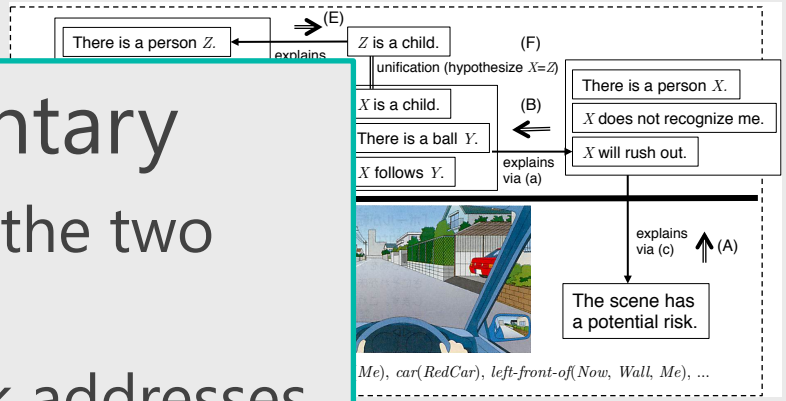
Logical Inference

(apply symb ic rules
infer new
[Armand+ 14, + 15, M

PROS: Easy ake ca

PROS: Can bine h

CONS: The diction
physical pr on



[Inoue+ 15]

Complementary

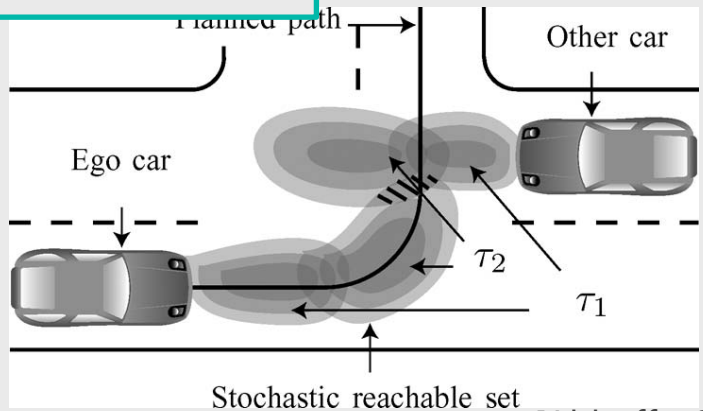
- How to combine the two components?
- No previous work addresses this issue

Physical Simulation

(simulate a scene and detect future collisions)
[Broadhurst+ 05, Althoff+ 09, etc.]

PROS: Precise prediction grounded in physical quantities

CONS: Difficult to take causal chain into account



[Althoff+ 09]

Evaluation | Models 1/3

Compare three models

Baseline (SVM):

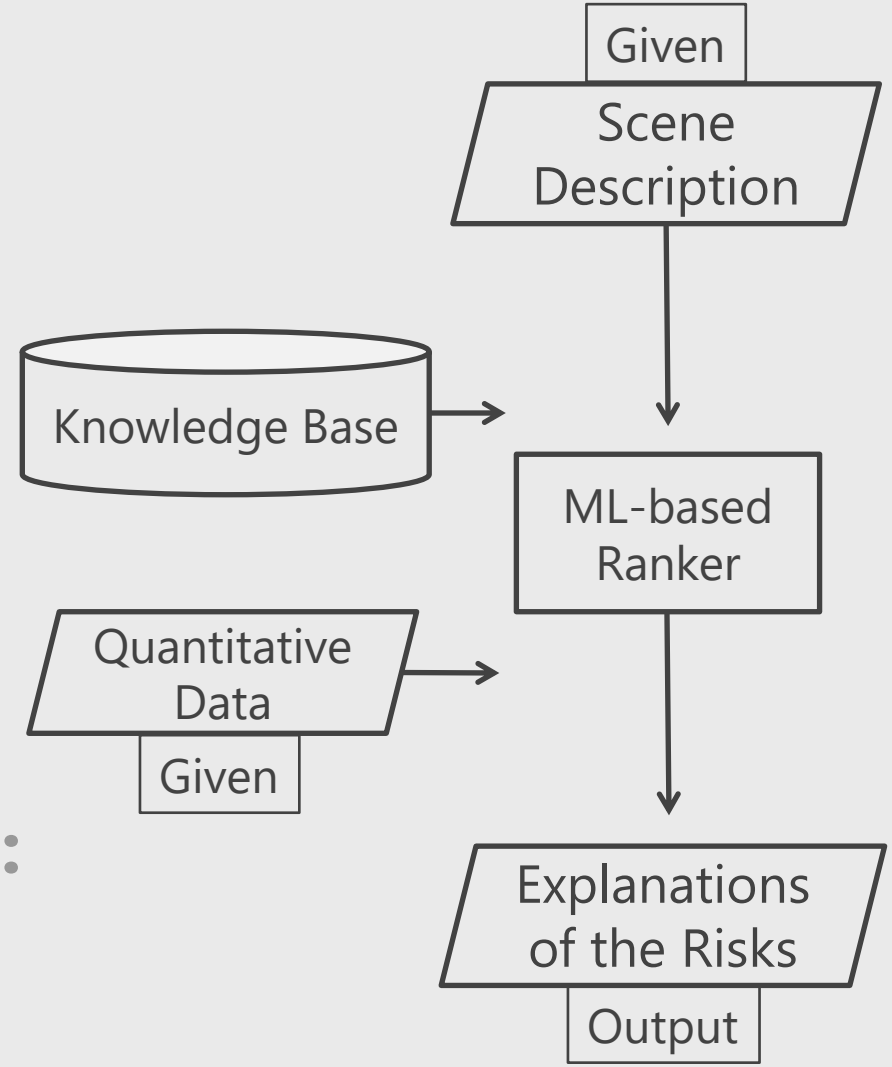
- Directly models between scene description and a risky entity-action pair
- Trained by using ranking-SVM [Joachims 02]

Proposed1 (Inference):

- Uses *only qualitative information*

Proposed2 (Inf+PhySim):

- The full model



Evaluation | Models 1/3

Compare three models

Baseline (SVM):

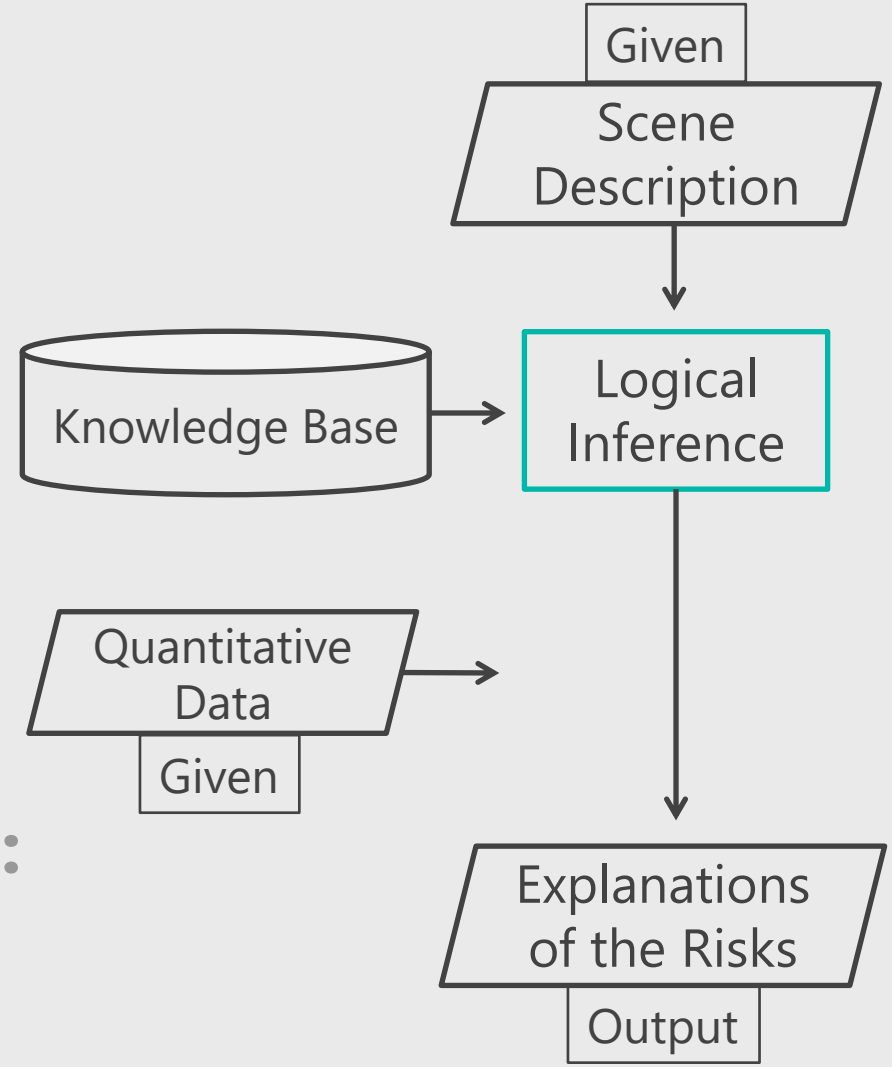
- Directly models between scene description and a risky entity-action pair
- Trained by using ranking-SVM [Joachims 02]

Proposed1 (Inference):

- Uses *only qualitative information*

Proposed2 (Inf+PhySim):

- The full model



Evaluation | Models 1/3

Compare three models

Baseline (SVM):

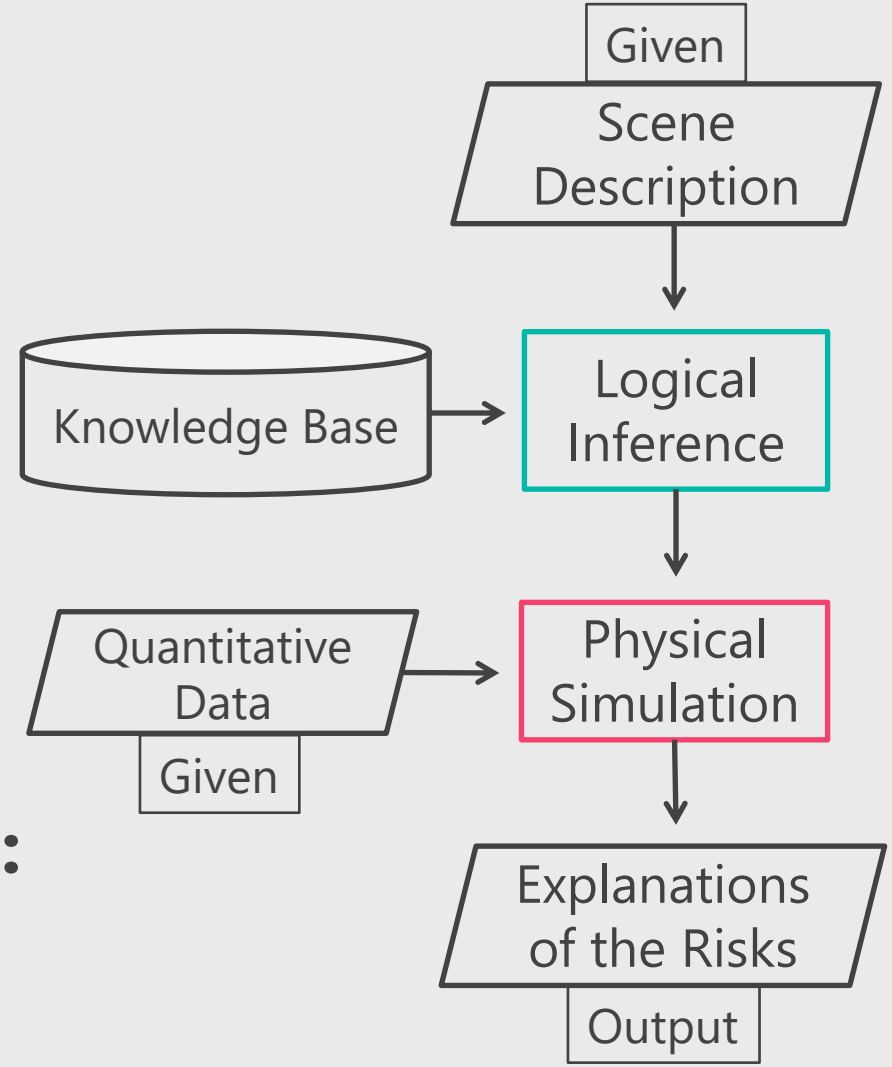
- Directly models between scene description and a risky entity-action pair
- Trained by using ranking-SVM [Joachims 02]

Proposed1 (Inference):

- Uses *only qualitative information*

Proposed2 (Inf+PhySim):

- The full model



Quantitative evaluation

Baseline (SVM):

- Directly models between scene description and a risky entity-action pair
- Trained by using ranking-SVM [Joachims 02]

Proposed1 (Inference):

- Uses *only qualitative information*

Proposed2 (Inf+PhySim):

- The full model

	Test		
Model	Acc@1	Acc@3	Acc@5
BASELINE	55.6 (40/72)	**77.8 (56/72)**	**93.1 (67/72)**
INFERENCE	**58.3 (42/72)**	**77.8 (56/72)**	91.7 (66/72)
INF+PHYSIM	**58.3 (42/72)**	**77.8 (56/72)**	91.7 (66/72)

- Adding abductive reasoning and physical simulation has *not yet* affected prediction accuracy positively
- Future work: improve the accuracy

② Modified SGD (Separated Learning Rates)

Our strategy

Different learning rates for different parts of our model

Rationale

NN usually can be decomposed into several parts, each one is convex when other parts are fixed

↓

NN ≈ joint co-training of many simple convex models

↓

Natural to assume different learning rate for each part

Modified

Different parts in a neural network may have different learning rates

$$\alpha_{\mathrm{KB}}(\tau_r) := \frac{\eta_{\mathrm{KB}}}{1 + \eta_{\mathrm{KB}}\lambda_{\mathrm{KB}}\tau_r}$$

$$\alpha_{\mathrm{AE}}(\tau_r) := \frac{\eta_{\mathrm{AE}}}{1 + \eta_{\mathrm{AE}}\lambda_{\mathrm{AE}}\tau_r}$$

η_{KB}: η for KB-learning objective

η_{AE}: η for autoencoder objective

λ_{KB}: λ for KB-learning objective

λ_{AE}: λ for autoencoder objective

τ_e: counter of each entity e

τ_r: counter of each relation r

④ Other Training Techniques

Normalization

normalize relation matrices to $\|\boldsymbol{M}_r\| = \sqrt{d}$ during training

$$\|\boldsymbol{M}_r\| = \sqrt{d}$$

+2.6 in Hits@10 on FB15k-237

Regularization

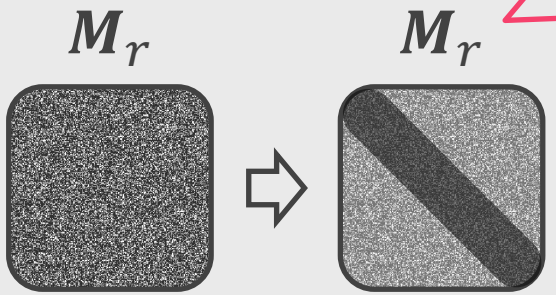
push $\boldsymbol{M}_r$ toward an orthogonal matrix

Minimize $\left\| \boldsymbol{M}_r^\top \boldsymbol{M}_r - \frac{1}{d}\mathrm{tr}(\boldsymbol{M}_r^\top \boldsymbol{M}_r) I \right\|$

+1.2 in Hits@10

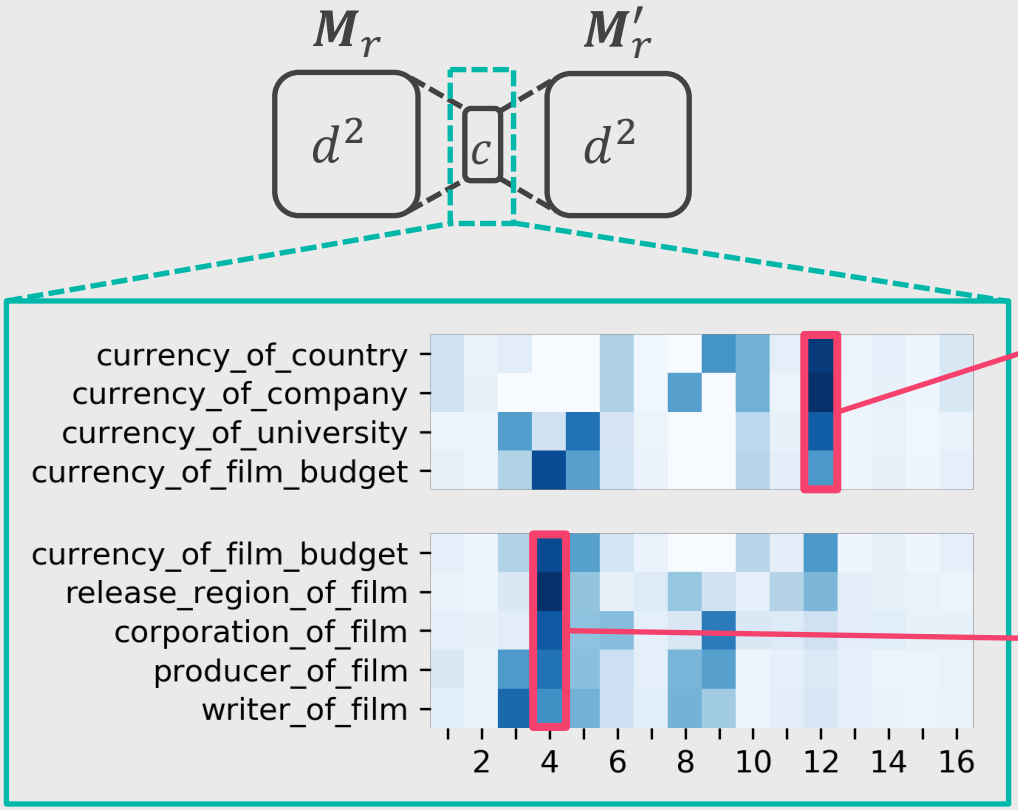
Initialization

initialize $\boldsymbol{M}_r$ as $(I + G)/2$ instead of pure Gaussian



+0.4 in Hits@10

What Does the Trained Autoencoder Look Like?



Dimension 12 strongly correlates with `currency`

Dimension 4 strongly correlates with `film`

- **Sparse coding of relation matrices**
- **Interpretable to some extent**

	Ks	Km	Us	Um
KG relation からなるシングルホップパス	✓	✓	✓	✓
KG relation からなるマルチホップパス		✓		✓
Textual relation からなるシングルホップパス			✓	✓
Textual relation を含むマルチホップパス				✓