

# ナレッジグラフとオントロジー

※ 本資料の著作権は、引用元の論文および記事に準じます

# Agenda

- ・ナレッジグラフ概観
- ・ナレッジグラフ構築の技術
- ・オントロジー学習の技術

# ナレッジグラフ概観

# Gartner Hype Cycle 2018 にナレッジグラフが登場

Gartner Solutions Insights What We Do Conferences | About | Careers

## Newsroom

### Press Releases

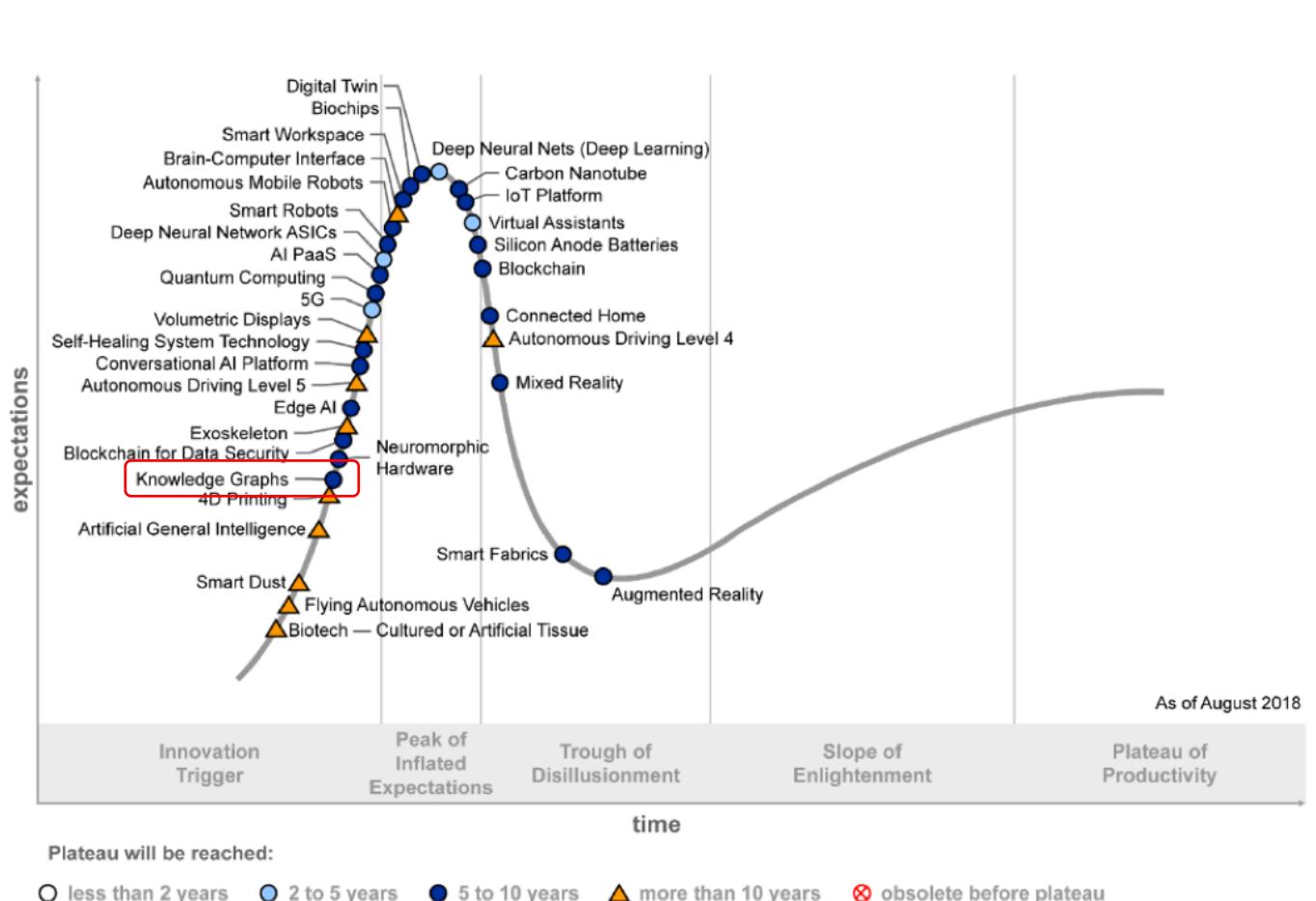
STAMFORD, Conn., August 20, 2018

### Gartner Identifies Five Emerging Technology Trends That Will Blur the Lines Between Human and Machine

2018 Emerging Technologies Hype Cycle Garners Insights From More Than 2,000 Technologies

The 35 must-watch technologies represented on the Gartner Inc. [Hype Cycle for Emerging Technologies, 2018](#) revealed five distinct emerging technology trends that will blur the lines between humans and machines. Emerging technologies, such as [artificial intelligence](#) (AI), play a critical role in enabling companies to be ubiquitous, always available, and connected to business ecosystems to survive in the near future.

Source: Gartner (August 2018)



# 知識とは

- ・人工知能で解決する問題
  - ・外界からの入力と,外界に対する出力の関係を決定
- ・知的な状況判断や意思決定
  - ・外界の入力だけで出力を決めるのは不可能
    - 例) 「ぼくはたぬき」
      1. たぬき蕎麦またはうどんの注文
      2. 本人が実は狸であることを告白
      3. 本人はややすしい性格であることを表現
  - ・人間は予め保持する「知識」を援用して判断

文献[1]をもとに作成

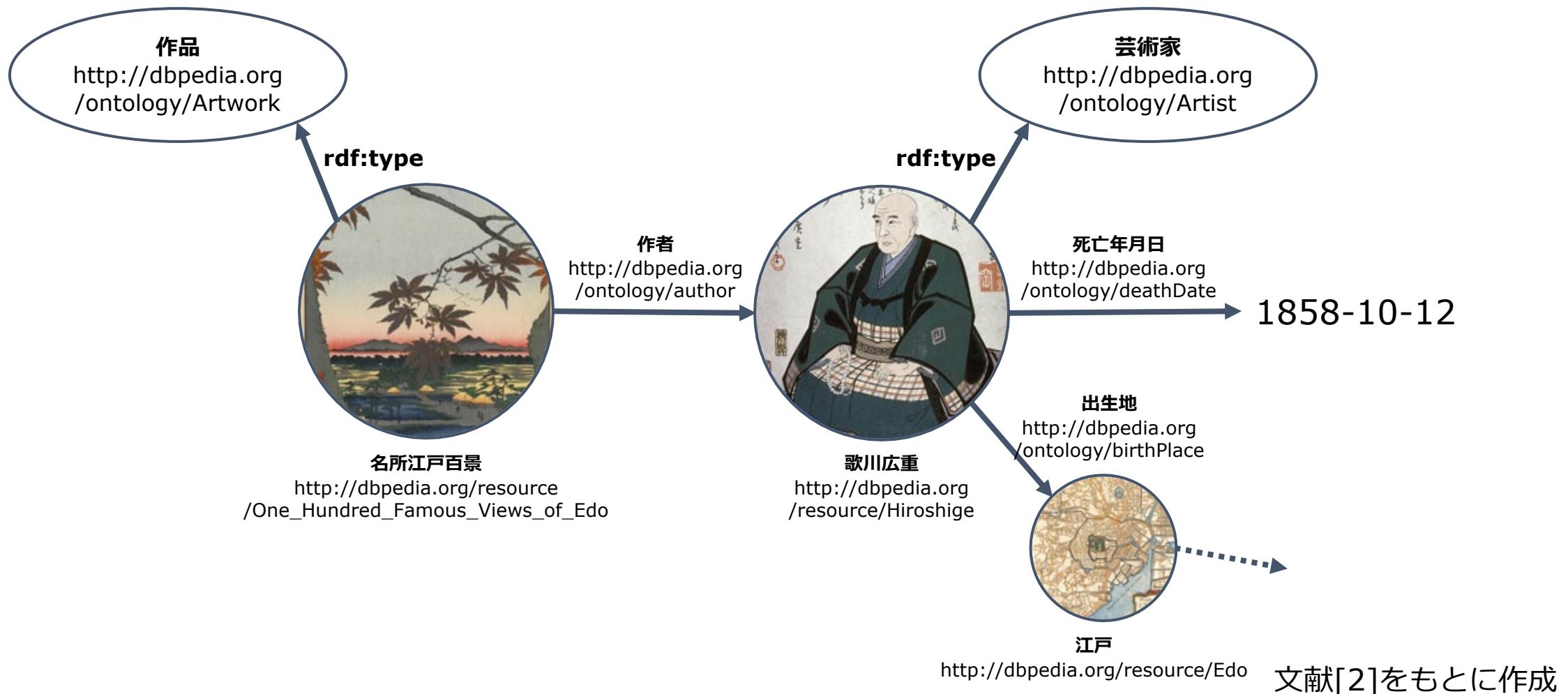
# 知識処理

計算機上で知識を取り扱う人工知能の技術

- 知識獲得
  - 知識をどのように習得するか
- 推論
  - 知識を使ってどのように思考するか
- 知識表現
  - 計算機上でどのように知識を表現するのか → ナレッジグラフ

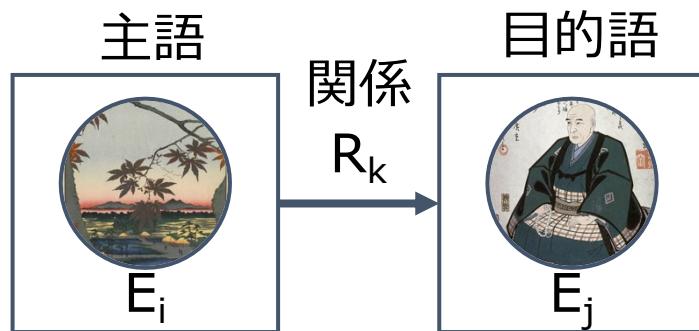
文献[1]をもとに作成

# ナレッジグラフの例：日本語DBpedia



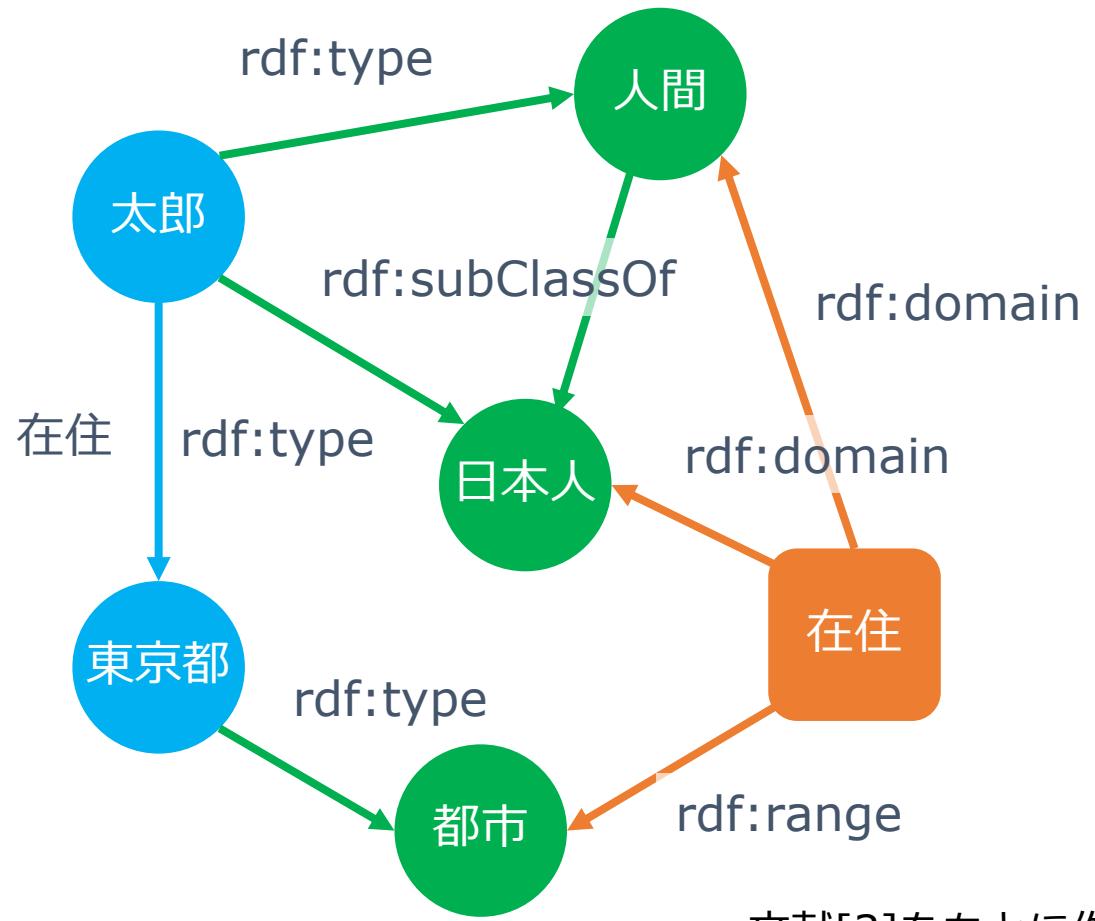
# ナレッジグラフの定義

- $G = (E, R, E)$



$E$ : エンティティ集合  
• インスタンス (物, 場所, 人)  
• クラス (ジャンル, 地域, 職業)

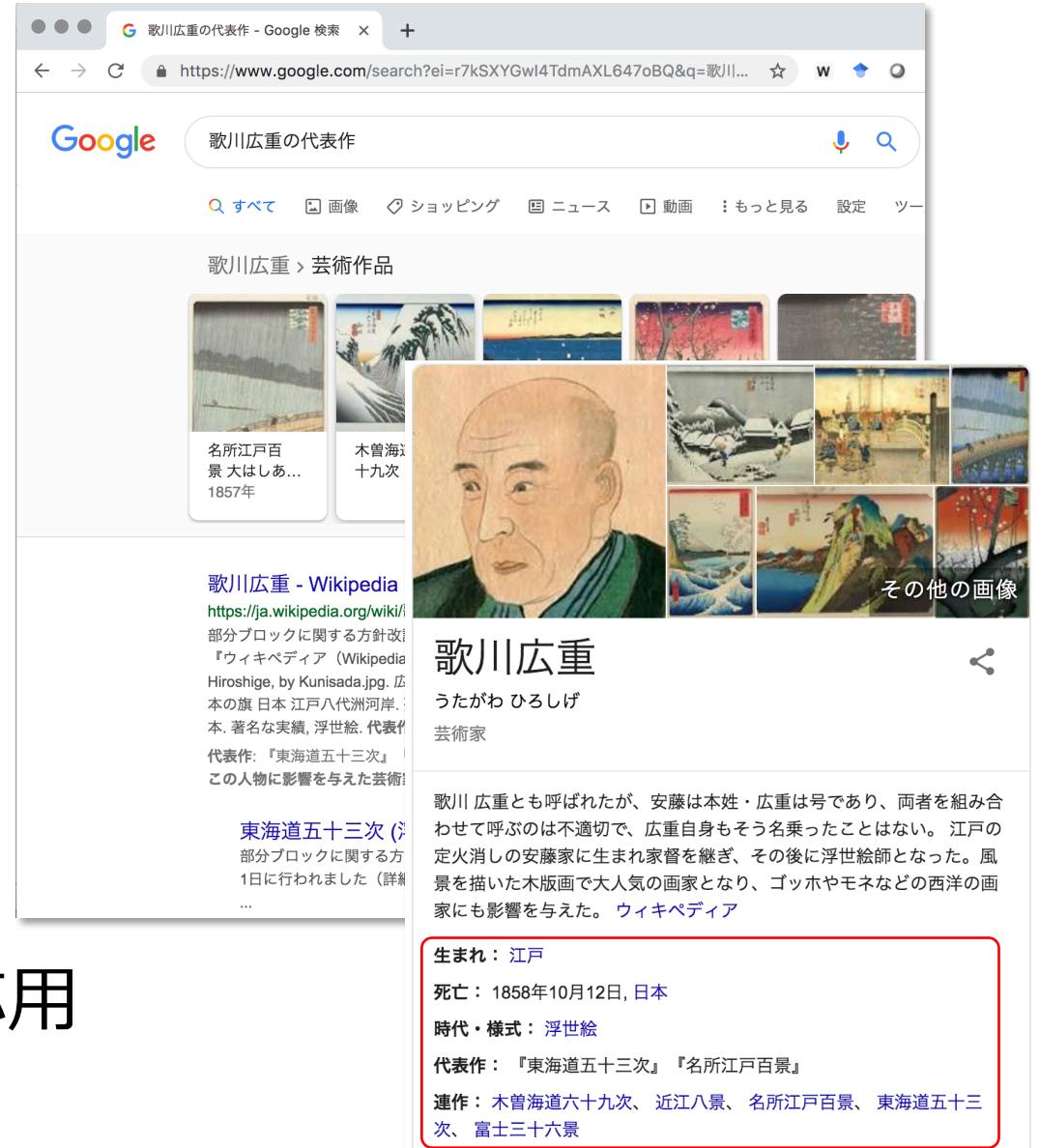
$R$ : 関係集合  
• リレーション (作者, 出生地, 在住)



文献[3]をもとに作成

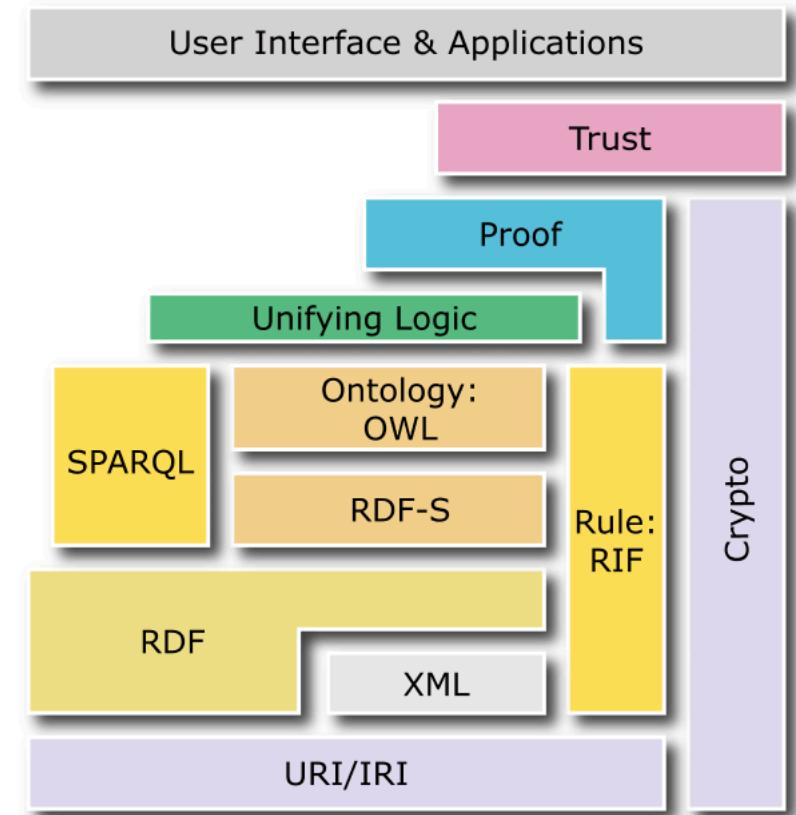
# ナレッジグラフ

- Web のように  
関連のある知識を辿って探せる  
→ 検索や質問応答システムで利用
- Googleなど多くのネット企業が  
独自のナレッジグラフを構築  
→ 検索目的の多くは、エンティティが  
どういうものかを知ること
- セマンティック Web の技術を応用



# セマンティックWeb (2000年頃~)

- Web上のデータを計算機で自動処理
  - 情報の意味的処理が可能
  - 効率的な検索・情報集約などを実現
- 規約を W3C で規定
  - XML マークアップ言語
  - RDF メタ情報の記述言語
  - SPARQL RDF検索言語
  - OWL オントロジー記述言語



<https://www.w3.org/2007/03/layerCake.png>

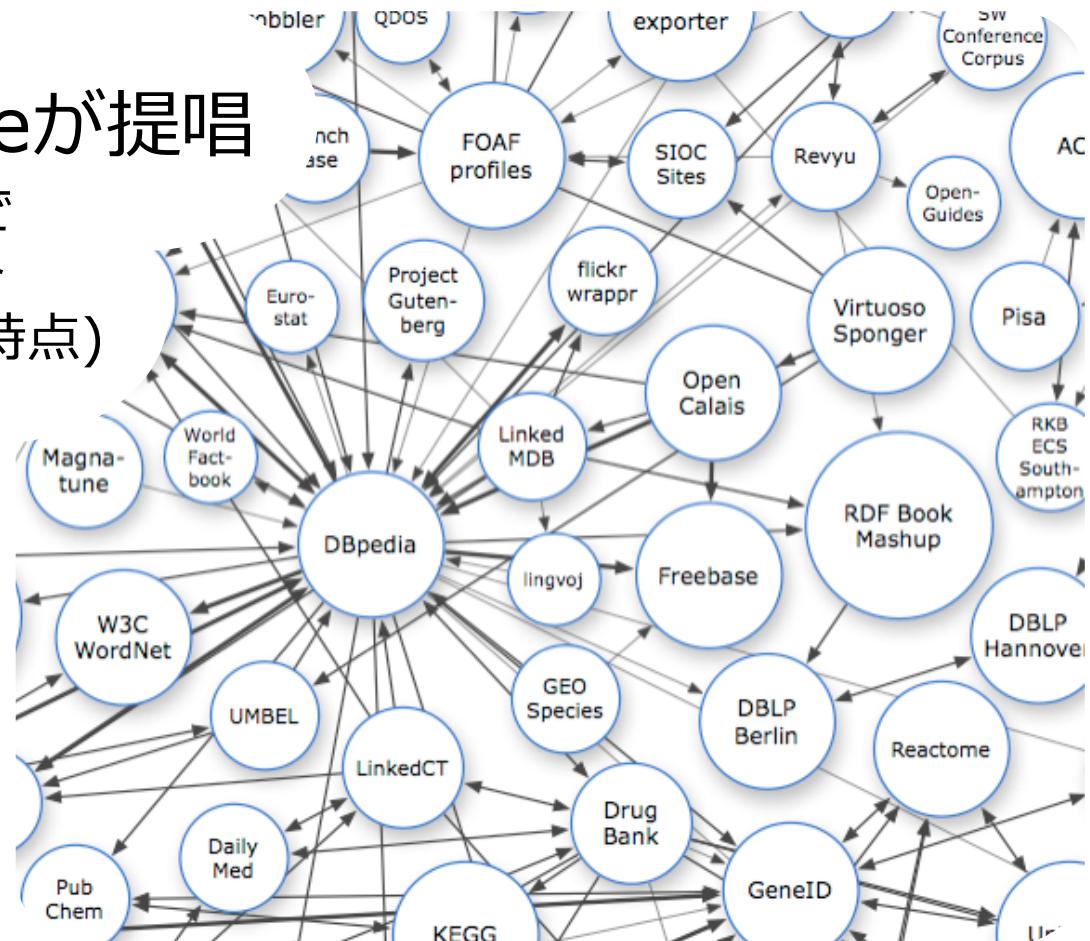
# Linked Data

- Webの創始者Tim Berners-Leeが提唱

- Web上のデータを相互リンクすることで新しい価値を生み出そうという取り組み
- 2007年 12個 →1,239個 (2019年3月時点)  
<https://lod-cloud.net/>

- 4つの基本原則

- すべてのデータにURIを付与
- URIを使ってデータの参照解決可能
- 標準の技術(RDFやSPARQL)を使用して役立つデータを提供
- 外部へのリンク(URI)を含めることで、多くの事物を発見できるように支援



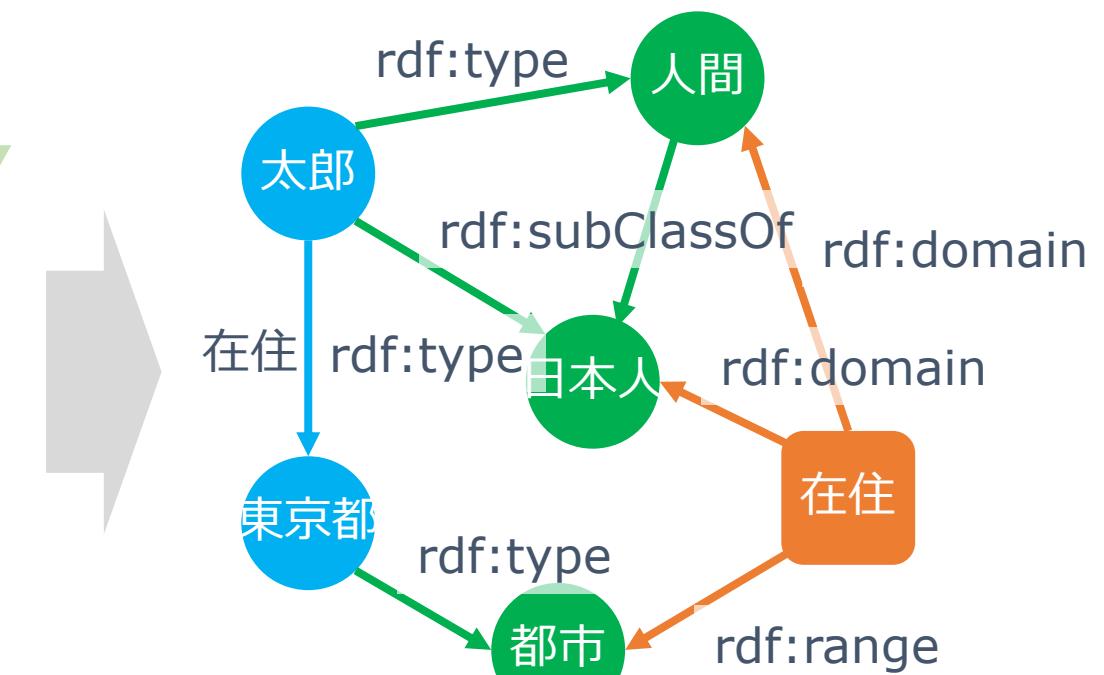
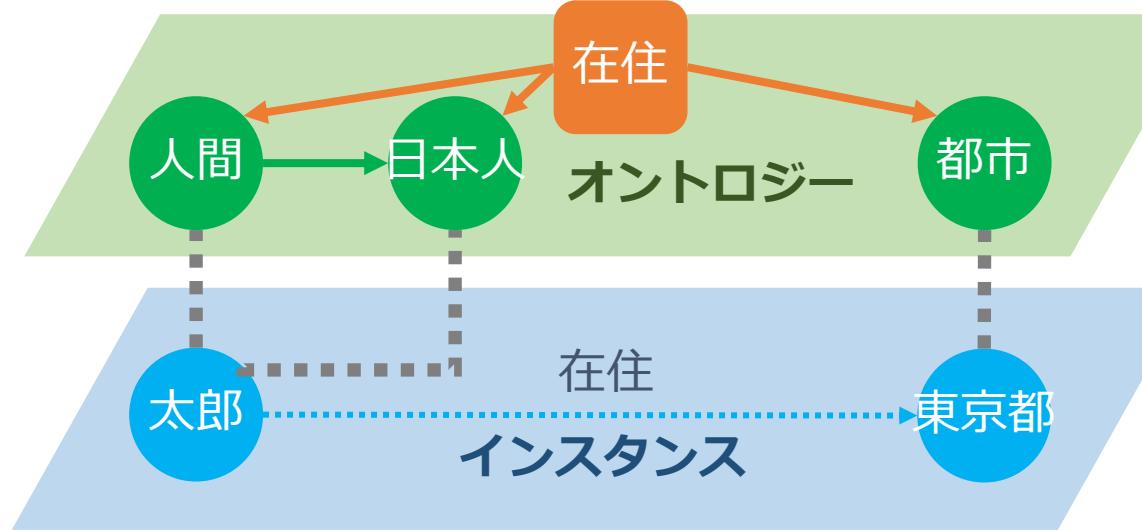
<http://linkeddata.org/>

# ナレッジグラフ

- ・ナレッジグラフ = オントロジー + インスタンス

オントロジー: 一般化されたレベルの知識 (クラス)

インスタンス: 具体例レベルの知識



# 代表的なナレッジグラフ

- 半構造化データから抽出, クラウドソーシングで構築

	インスタンス	ファクト	クラス	関係
DBpedia	4,806,150	176,043,129	735	2,813
YAGO	4,595,906	25,946,870	488,469	77
Freebase	49,947,845	3,041,722,635	26,507	37,781
Wikidata	15,602,060	65,993,797	23,157	1,673
Google KG	570,000,000	18,000,000,000	1,500	35,000

文献[4]からの引用

# 森羅プロジェクト

<http://liat-aip.sakura.ne.jp/森羅/> 森羅 wikipedia 構造化プロジェクト 2019/



- DBpedia や YAGO など,多くのナレッジグラフが構築 → 汚い
  - 情報の設計が指針なしでクラウドで作られている  
例) DBPedia はスポーツの下に4つカテゴリしかない  
→ ボトムアップで定義するのは無理
- モチベーション: きれいなナレッジグラフを作りたい
  - 知識の定義はトップダウン → 拡張固有表現(200カテゴリ)
  - 知識の内容はボトムアップにまかせる

文献[6]をもとに作成

# 森羅プロジェクト



- Wikipedia項目と拡張固有表現による構造化

部分ブロックに関する方針改訂が6月1日に行われました（[詳細](#)）。

## 小松飛行場

出典: フリー百科事典『[ウィキペディア \(Wikipedia\)](#)』

**小松飛行場**（こまつひこうじょう）は、[石川県小松市](#)にある共用飛行場である。

防衛省が管理しており、[航空自衛隊小松基地](#)（英: JASDF Komatsu Airbase）と民間航空（民航）が滑走路を共用する飛行場で、特に後者においてはターミナルビルなどの施設の通称として小松空港（こまつこうじょう、英: Komatsu Airport）と呼ばれている<sup>[1]</sup>。航空交通管制は航空自衛隊が行なっている。

**ターミナルビル**

**IATA: KMQ - ICAO: RJNK**

**概要**

- 1 概要
- 2 歴史
  - 2.1 年表
  - 2.2 旅客数
- 3 旅客・貨物施設
  - 3.1 空港内に施設をもつ行政機関・企業
- 4 定期就航路線
  - 4.1 国内線
  - 4.2 國際線乗継便



```
{
    "Name": "小松飛行場",
    "WikidataID": "100192",
    "ENE": "空港名",
    "Attributes": {
        "ふりがな": ["こまつひこうじょう"],
        "IATA (空港コード)": ["KMQ"],
        "ICAO (空港コード)": ["RJNK"],
        "別名": ["Komatsu Airbase", "Komatsu Airport", "小松空港", "FAC4017小松補助飛行場"]
    },
    "NameBy": [
        "名称由来": [],
        "名称由來人物の地位職業名": [],
        "国": ["日本"],
        "年間利用客数": [],
        "年間利用者数データの年": [],
        "年間発着回数": [],
        "年間発着回数データの年": [],
        "座標": {"經度": "東経136度24分27秒", "緯度": "北緯36度23分38秒", "高さ": "36.39389度"},
        "所在地": ["石川県小松市", "むじなが浜"],
        "旧称": [],
        "標高": ["6 m", "18 ft"],
        "母都市": ["福井市", "金沢市"],
        "滑走路数": ["2本"],
        "滑走路の長さ": ["2,700"],
        "総面積": ["241ha"],
        "近隣空港": [],
        "運営者": ["航空自衛隊"],
        "運用時間": [],
        "開港年": ["1953年 (昭和28年) 4月3日"]
    ]
}
```

文献[5][6]からの引用

# 森羅プロジェクト



- 構造化のアプローチ
    1. Wikipedia 100万ページを分類 (自動+アノテータ) →済み
    2. 78万項目を構造化する (クラウドソーシング+アノテータ)
      - ・クラウドでやると3億円以上かかる →RbCC
  - RbCC (Resource by Collaborative Contribution)
    - 評価型ワークショップを活用
      - 例えば,10中,8チームが正しければリソースを作ってしまう (Ensemble Learning)
      - 適切な人手チェックを入れてデータを拡張 (Active Learning)
      - 拡張したデータで再度タスクを実施 (Bootstrapping)
- Wikipedia構造化プロジェクト2018～2019

文献[6]をもとに作成

# まとめ

- ・ナレッジグラフは、知識を計算機上で表現したもの
  - ・ナレッジグラフ = オントロジー(クラス) + インスタンス(実体)
- ・セマンティック Web 技術を応用し、Linked Data として公開
  - ・Google や Yahoo!, Facebookなどの大手も独自のナレッジグラフを構築
- ・ボトムアップで情報を設計するのは難しい（森羅の示唆）
  - ・知識の形式 = トップダウン・・・オントロジー作成 → 人手
  - ・知識の内容 = ボトムアップ・・・構造化→ 人手 + 自動化

# 文献

- [1] 市瀬龍太郎, et al. "レクチャーシリーズ:「人工知能の今」[第3回] 知識表現—オントロジー,知識グラフー." 人工知能 34.4 (2019): 556-565.
- [2] 加藤文彦. "DBpedia の現在: リンクトデータ・プロジェクト." 情報管理 60.5 (2017): 307-315.
- [3] 林克彦. "知識グラフと分散表現." 言語処理学会第25回年次大会チュートリアル資料, 2019 (2019).
- [4] Paulheim, Heiko. "Knowledge graph refinement: A survey of approaches and evaluation methods." Semantic web 8.3 (2017): 489-508.
- [5] 「森羅2019」プロジェクトについての説明資料  
<https://drive.google.com/open?id=18SIQT2k6GcAB-xolsNZlg6KrKWvMHsij>
- [6] 関根聰,小林暁雄,安藤まや, and 乾健太郎. "拡張固有表現に基づく Wikipedia 項目の分類と構造化." 第43回SWO研究会, 2017.

# オントロジー構築の技術

# オントロジー構築の方法

オントロジー構築の方法は,2つに大別される

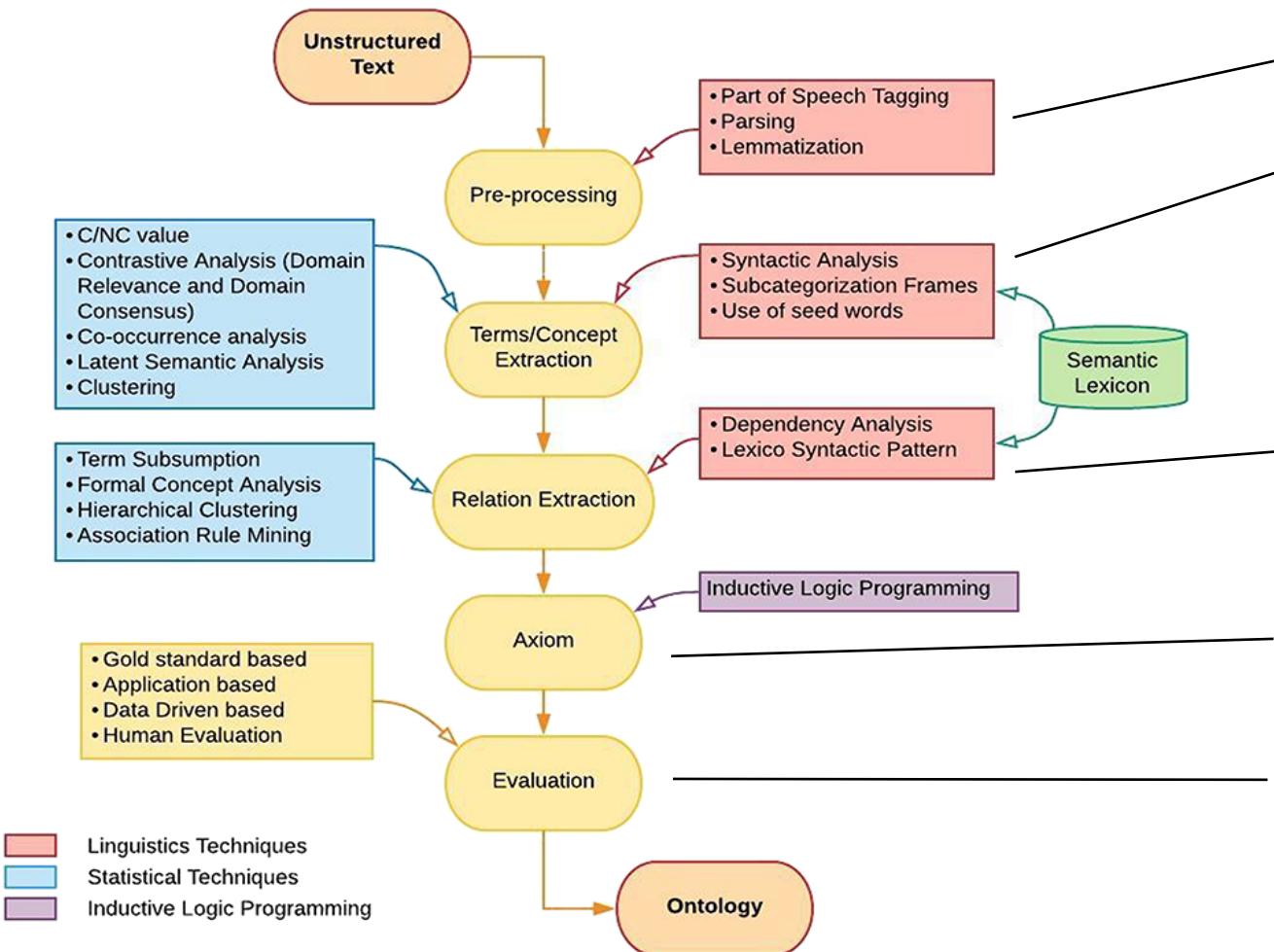
- 手動による構築 →この話はしません
  - オントロジーエディタ(Protégé等)を用いて構築
- 半自動による構築 →以降で簡単に紹介しています
  - **オントロジー学習**の手法を実装したツールを用いて構築
  - 各タスクの出力は,専門家またはオントロジーエンジニアが検証

文献[14]をもとに作成

# (参考) 手動による構築の例

- ・タスクの明確化, 分析対象の選定
- ・RDFデータ分析
  - ・ルートノードの作成 (root)
  - ・ルートノードと項目の紐付け (root → リテラルノード)
  - ・リテラルノードの分割 (\_brank → リテラルノード)
  - ・マスタノードとの紐付け (\_brank → マスタノード)
- ・オントロジー作成
  - ・クラス・プロパティ変換 (ノード⇒class, 矢印⇒{object property, data property} )
  - ・カーディナリティ, データタイプの付与
  - ・OWL等の定義済み語彙との紐付け

# オントロジー学習



**データ前処理:** 品詞タグ付け, 構文解析, 見出し化など自然言語処理技術によるテキストへの前処理を行う

**用語/概念および概念階層抽出:** 構文解析や格フレーム解析, シードワード(右), 専門用語抽出や関連性評価, 共起分析, LSA, クラスタリング(左)を用いて用語および概念を抽出する

**関係および関係階層抽出:** 係り受け解析や語彙統語パタン(右), 形式概念解析, 階層的クラスタリング, 相関ルール(左)を用いて用語/概念間の意味的関係および関係階層を抽出する

**公理の導出:** 背景知識と論理プログラミングを使用してテキストから推論規則を抽出する

**評価:** グラフの類似度, 検索結果の比較などを行い, 生成したオントロジーを評価する

文献[8]からの引用

# 主な手法

主な手法			主なシステム
自然言語処理	データ前処理	Berkley Parser	Text2Onto [Cimiano+,2005], CRCTOL [Jiang+,2010]
		Stanford Parser	
		構文解析（主要語-修飾語解析）	
	関係抽出	語彙統語解析	Text2Onto [Cimiano+,2005], CRCTOL [Jiang+,2010], ASIUM [Faure+,2000], TextStorm/Clouds [Oliveira+,2001]
		依存構造解析	
統計的手法	用語/概念抽出	専門用語抽出 (C/NC-Value)	OntoGain [Drymonas+2010], CValue-TermExtraction [Frantzi+,2000]
		対照分析	OntoLearn [Navigli+,2003], CRCTOL [Jiang+,2010], OntoGain [Drymonas+2010]
		共起分析	Text2Onto [Cimiano+,2005], Sematch [Zhu+,2017]
		クラスタリング	ASIUM [Faure+,2000], Text2Onto [Cimiano+,2005]
	関係抽出	形式概念解析 (FCA)	OntoGain [Drymonas+2010]
		階層的クラスタリング	Text2Onto [Cimiano+,2005]
		相関ルールマイニング (ARM)	Text2Onto [Cimiano+,2005]
論理的手法		帰納的論理プログラミング	TextStorm/Clouds [Oliveira+,2001] , Syndikate [Hahn+,2000]

※ 2000年代の前半頃から多く研究されている一方で、最近の実装が少ない

文献[8]からの引用

# Text2Onto

- A Framework for Ontology Learning and Data-driven Change Discovery
- Can Text2Onto **automatically build** an ontology by **learning** on a corpus of texts?  
→ **No**
- Can Text2Onto **help** a user to build an ontology?  
→ **Yes, but it needs improvement**

文献[9]から引用

# まとめ

- ・オントロジー学習は、オントロジー構築を半自動化するための支援ツールで、自然言語処理や統計解析など、複数の技術で構成されている
- ・オントロジー学習のためのシステムやツールが存在したが、多くが2010年までのもので、最近の実装が少ない
- ・オントロジー学習は、各タスクが改善を必要とする、膨大な研究であり、発展途上分野

# 文献

- [7] 森田武史, and 山口高平. "オントロジー学習の現状と動向 (<特集> オントロジーの進化と普及 (前編))." 人工知能学会誌 25.3 (2010): 354-365.
- [8] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. A survey of ontology learning techniques and applications. Database, 2018:bay101, 2018.
- [9] Cimiano, Philipp, and Johanna Völker. "text2onto." International conference on application of natural language to information systems. Springer, Berlin, Heidelberg, 2005.

# ナレッジグラフ構築の技術

# 技術マップ

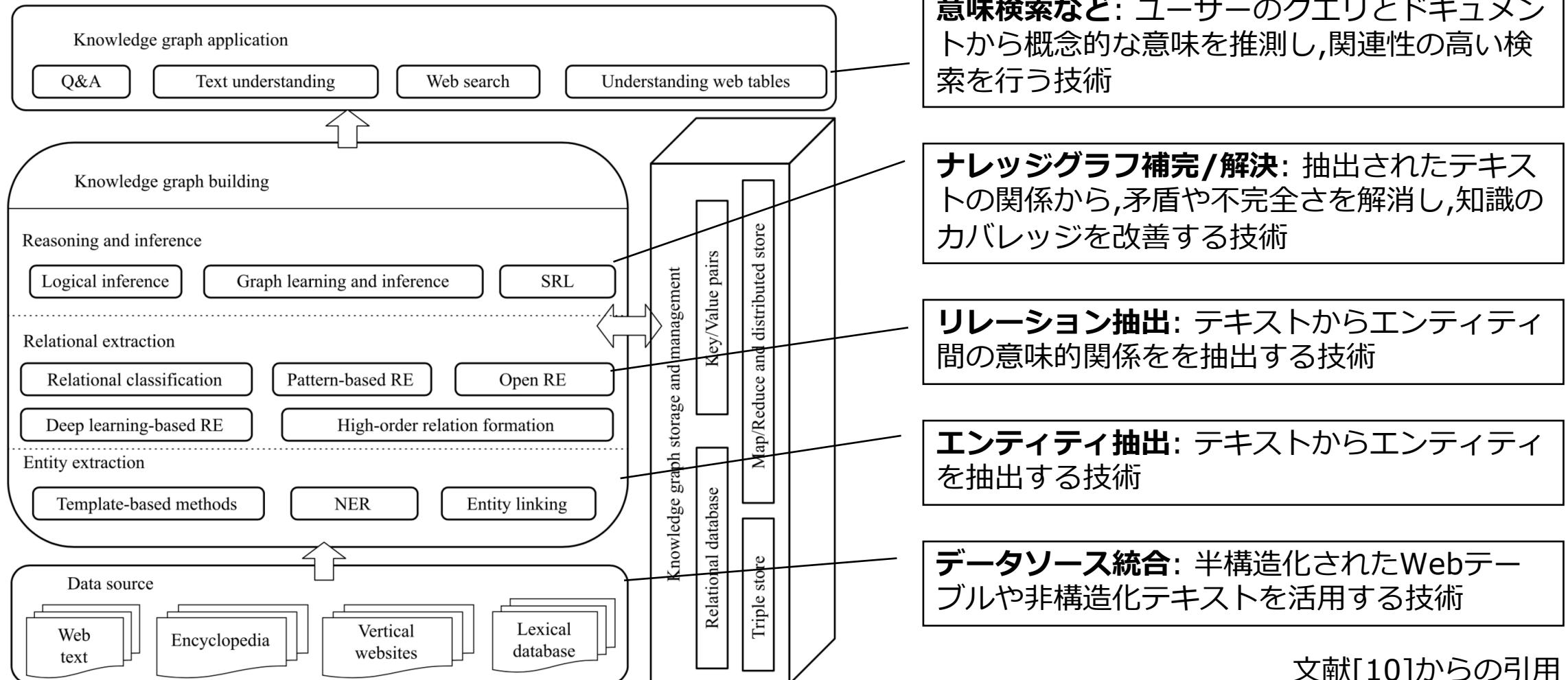
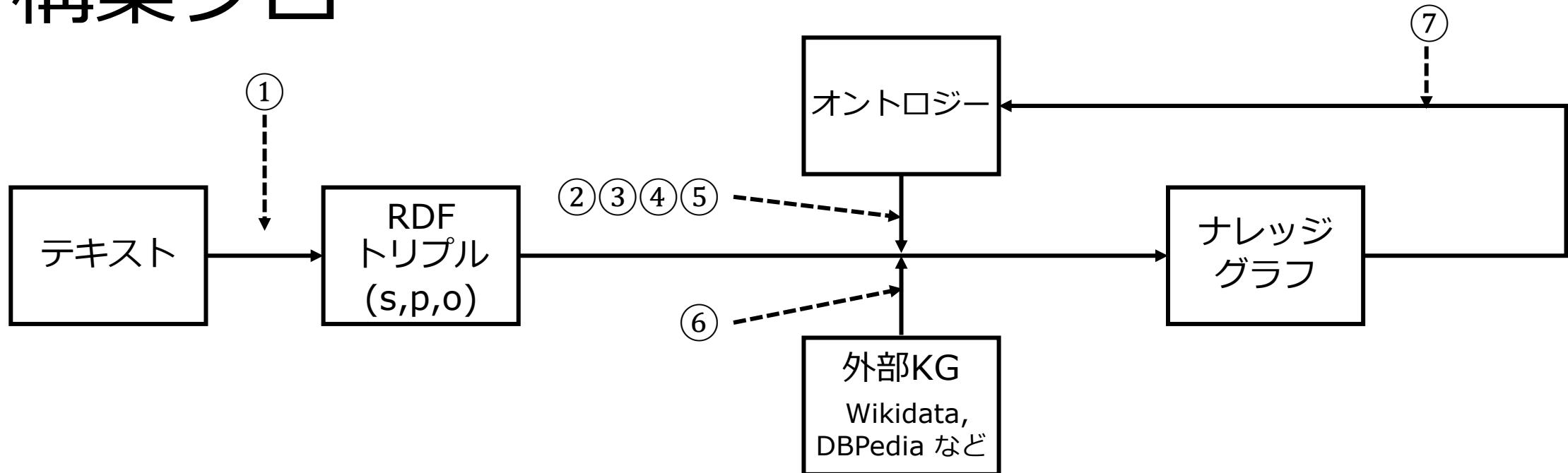


Fig. 1 The framework of KG

# 構築フロー



**抽出対象(項目や関係)を限定できる:** 難易度低, 高品質

辞書ベース

パターンベース

機械学習(固有表現抽出)

**抽出対象(項目や関係)を限定できない:** 難易度高, ノイズが多い

① トリプル抽出

形態素解析, 述語項構造解析,  
照応解析, 共参照解析など

② エンティティ抽出

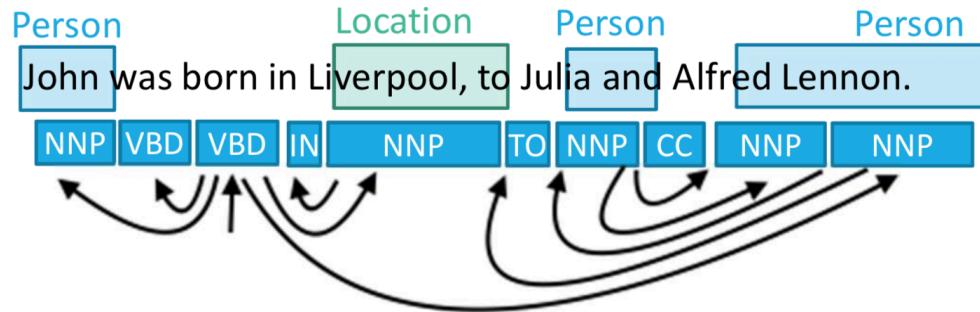
③ リレーション抽出

④ ナレッジグラフ補完, ⑤ ナレッジグラフ解決

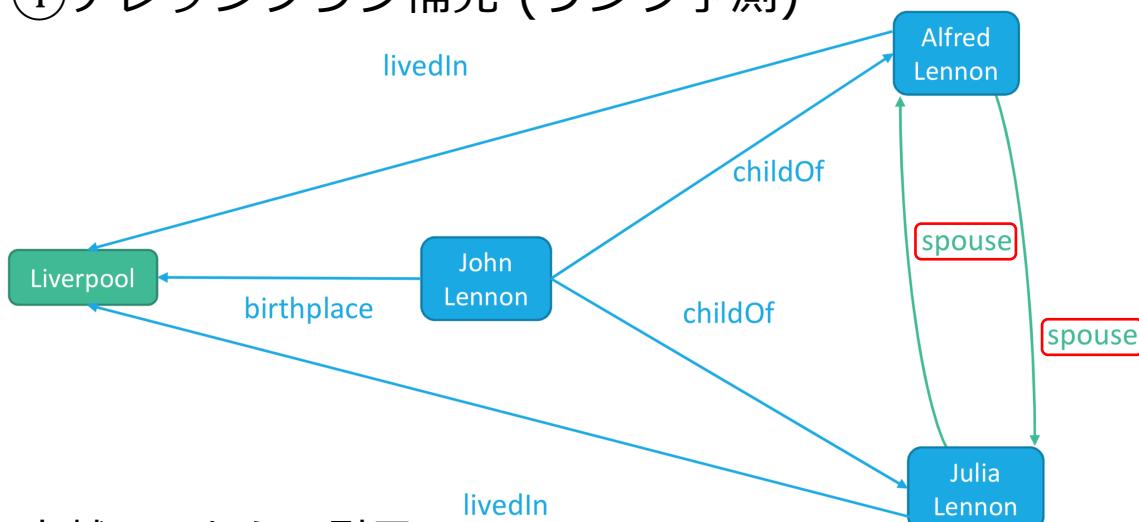
⑥ エンティティリンク

⑦ オントロジー学習

## ①トリプル抽出, ②エンティティ抽出

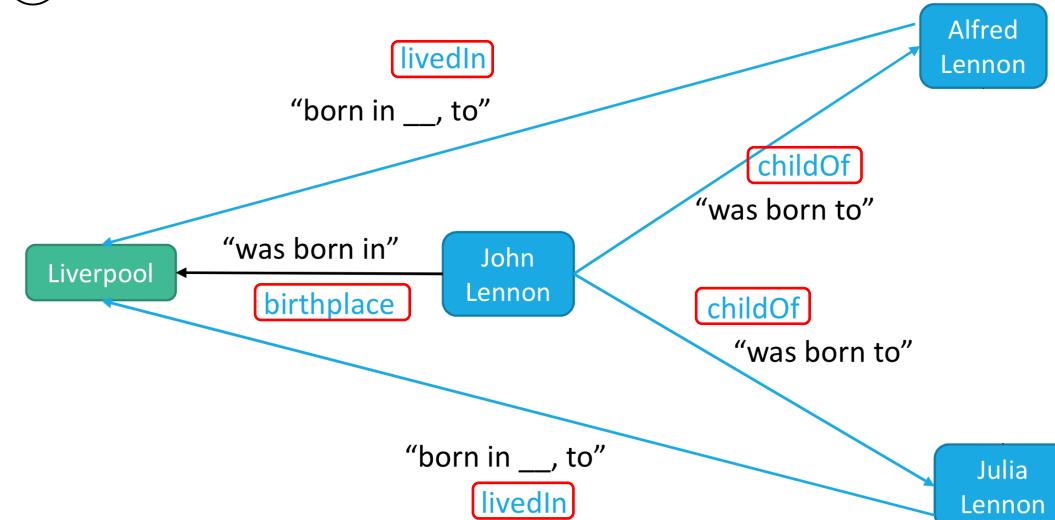


## ④ナレッジグラフ補完 (リンク予測)



文献[11]からの引用

## ③リレーション抽出



## ⑤ナレッジグラフ解決 (PSL-KGIの例)

### Uncertain Extractions:

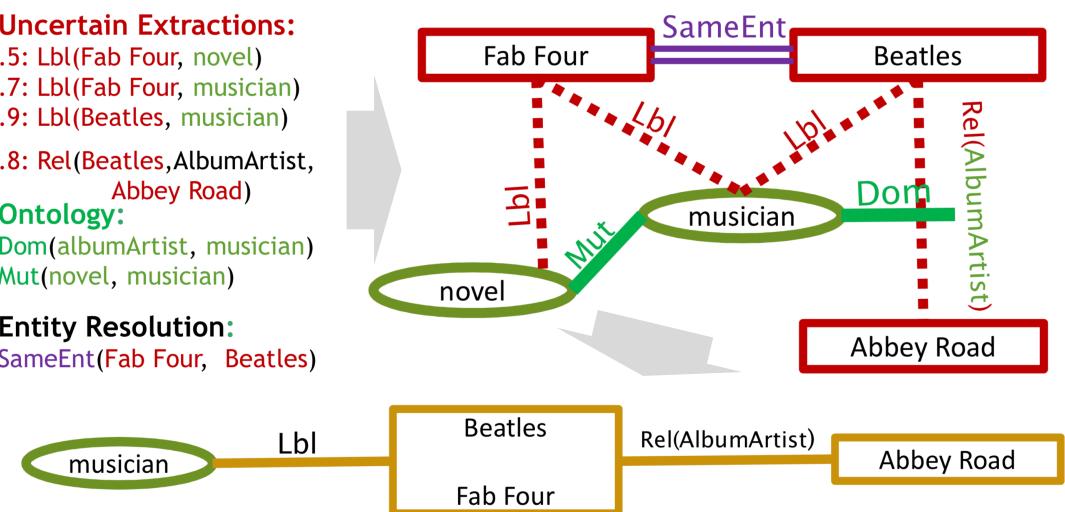
- .5: Lbl(Fab Four, novel)
- .7: Lbl(Fab Four, musician)
- .9: Lbl(Beatles, musician)
- .8: Rel(Beatles, AlbumArtist, Abbey Road)

### Ontology:

Dom(albumArtist, musician)  
Mut(novel, musician)

### Entity Resolution:

SameEnt(Fab Four, Beatles)



# 主な手法

抽出対象	アプローチ	主なNLPタスク	主なデータセット	主な実装	必要な準備
限定できる ・難易度低 ・高品質	辞書ベース	キーワードマッチング	—	不要	辞書作成
	パタンベース（ルールベース）	パタンマッチング	—	不要	ルール作成
	機械学習	固有表現抽出	CoNLL-2003	NCRF++	学習データ作成
限定できない ・難易度高 ・ノイズ多い	①トリプル抽出	形態素解析,述語項構造解析,照応解析,共参照解析などの複合的なタスク	Penn Tree Bank, CoNLL-2012, OntoNotes	OpenIE (w/ CoreNLP)	対象言語のモデル調達
	②エンティティ抽出	Entity Typing	Open Entity, FIGER	ERNIE	学習データ作成
	③リレーション抽出	Relation Extraction	TACRED,FewRel	ERNIE	学習データ作成
		Relationship Extraction (Distant Supervised)	NYT-Corpus	RESIDE	類似ドメインのナレッジベース調達
	④ナレッジグラフ補完	Link Prediction	WN18RR, FB15k-237	TuckER	シードにするナレッジグラフ
	⑤ナレッジグラフ解決	KG Identification	—	PSL-KGI	オントロジー作成
共通	⑥エンティティ・リンク	Entity Linking	CoNLL-YAGO	Wikipedia2vec	学習データ作成

※ 数多のタスクや実装が存在するが、日本語に対応した学習済みモデルやデータセットが少ない

# 技術トレンド

## 古典的手法の限界

- 推論規則や特徴量設計など  
マニュアル設計による表現の制限
- 存在しないエンティティや関係の  
一般化が難しい

## 計算量の問題

- 複雑さは明示的な次元数に依存
- ルールが多くなると推論コストが高
- データサイズによっては NP困難
- クエリはしばしばNP困難
- 一般的に並列化やGPUが使われない

## 埋め込みモデル

- 密ベクトル
- 多くの関係を学習データから獲得
- 複雑さは潜在的な次元に依存
- 勾配法や逆伝搬が利用できる
- クエリはしばしば低コスト
- GPU並列化にフレンドリー

文献[11]からの引用

# 埋め込みモデルー平行移動型

- word2vec のアナロジー操作にヒントを得たモデル

- TransE<sup>[Nickel+,11]</sup>, TransR<sup>[Lin+,15]</sup>, ManifoldE<sup>[Xiao+,16]</sup>, TorusE<sup>[Ebisu+,18]</sup>, RotatE<sup>[Sun+,18]</sup> など

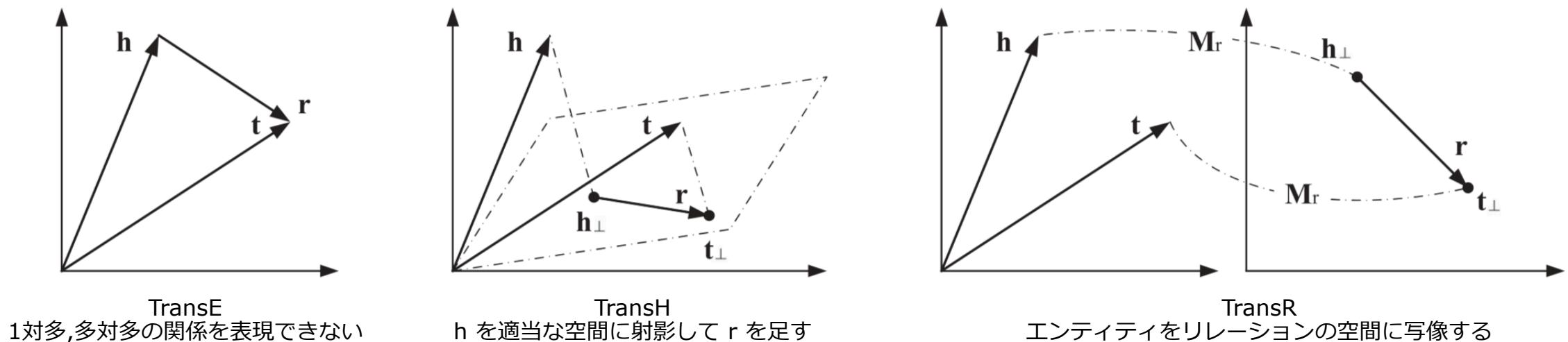


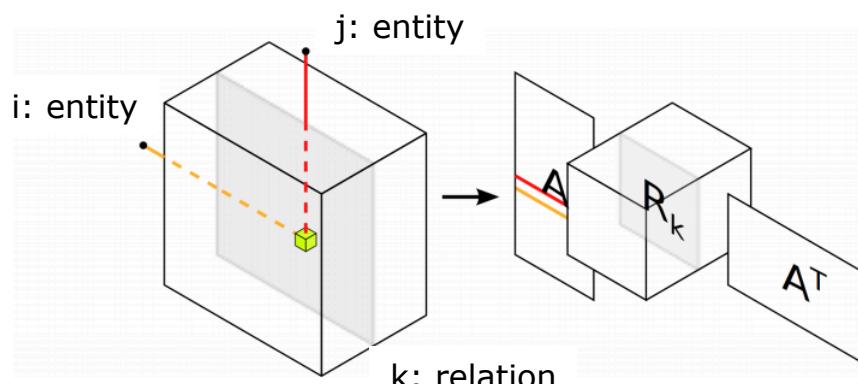
Fig. 1. Simple illustrations of TransE, TransH, and TransR. The figures are adapted from [15], [16].

文献[14]からの引用

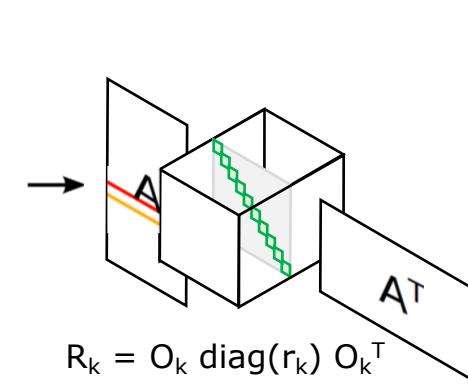
# 埋め込みモデル — 双線形型

- 行列・テンソル分解から発展したモデル

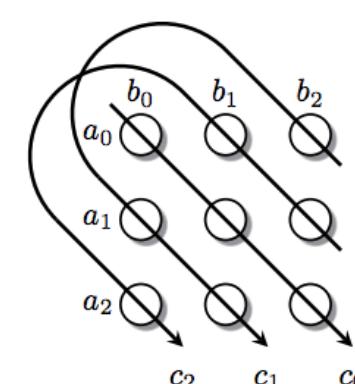
- RESCAL[Nickel+,11], DistMult[Yang+,14], HolE[Nickel+,16], ComplEx[Trouillon+,16],  
Analogy[Liu+17], SimplE[Kazemi+,18], TuckER[Balazević+,19] など



RESCAL  
シンプル, 計算効率が悪い:  $O(d^2)$



DistMult  
直行対角化:  $O(d)$ , 非対称を表現できない



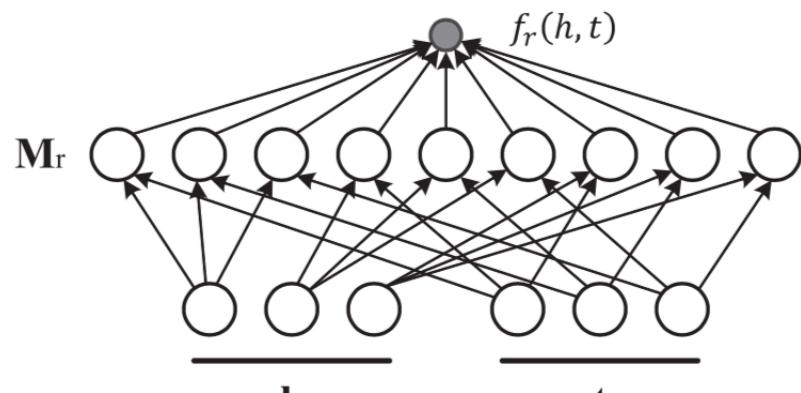
HolE  
巡回相互関係:  $O(d \log d)$ , 非対称を表現

文献[12][13]をもとに作成

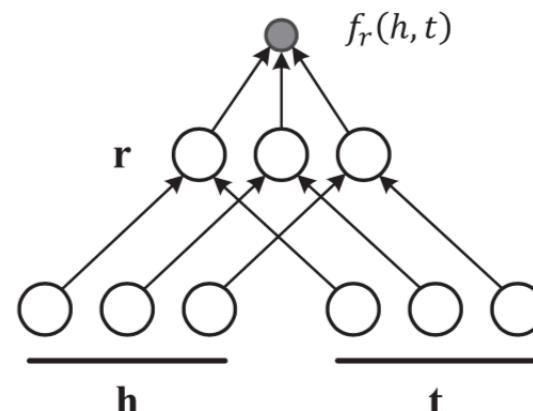
# 埋め込みモデル — 双線形型

- 行列・テンソル分解から発展したモデル

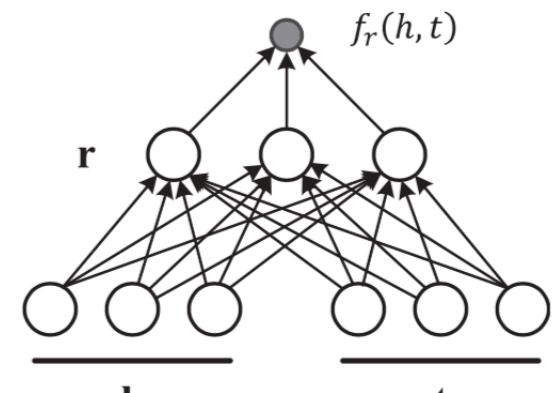
- RESCAL[Nickel+,11], DistMult[Yang+,14], HolE[Nickel+,16], ComplEx[Trouillon+,16],  
Analogy[Liu+17], SimplE[Kazemi+,18], TuckER[Balazević+,19] など



RESCAL  
シンプル, 計算効率が悪い: $O(d^2)$



DistMult  
直行対角化: $O(d)$ , 非対称を表現できない



HolE  
巡回相互相関: $O(d \log d)$ , 非対称を表現

Fig. 2. Simple illustrations of RESCAL, DistMulti, and HolE. The figures are adapted from [62].

文献[14]からの引用

# 埋め込みモデル — ニューラルネット型

- ニューラルネットワークから発展したモデル

- SME[Bordes+,14], NTN[Socher+,13], MLP[Dong+,14], MAM[Liu+,16], ConvE[Dettmers+,18]など

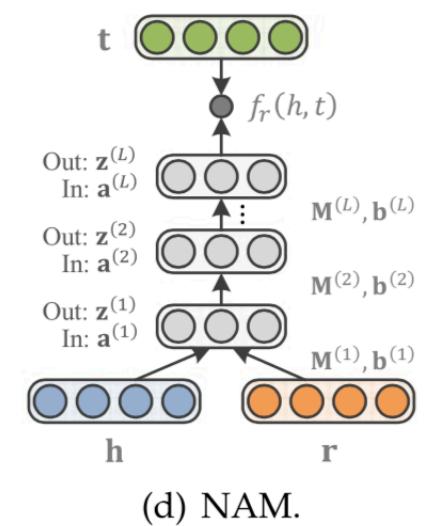
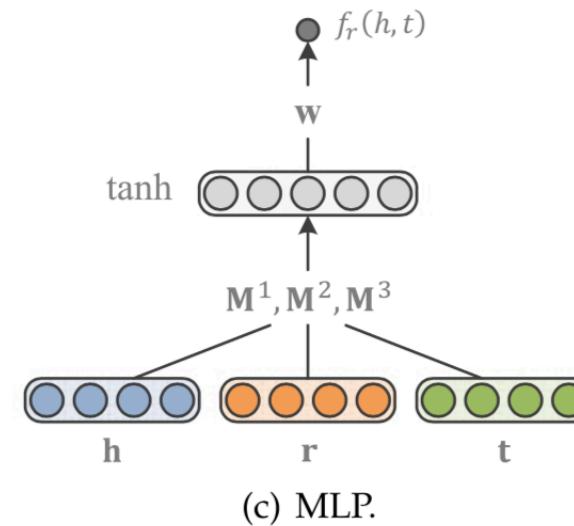
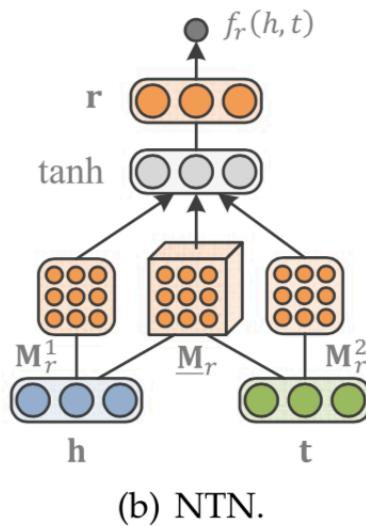
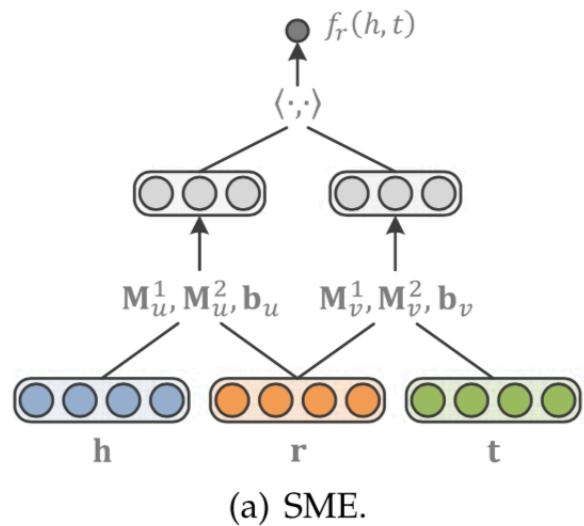


Fig. 3. Neural network architectures of SME, NTN, MLP, and NAM. The figures are adapted from [18], [19], [63].

文献[14]からの引用

# 埋め込みモデル — グラフ畳み込み

文献[15]からの引用

- ・グラフ構造の畳み込みを適用したモデル



Represent knowledge base as

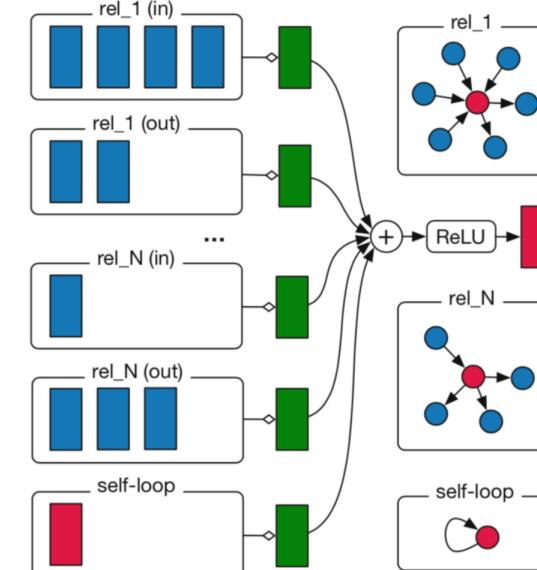
**directed, labeled graph**  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$

with edges:

and relation types:

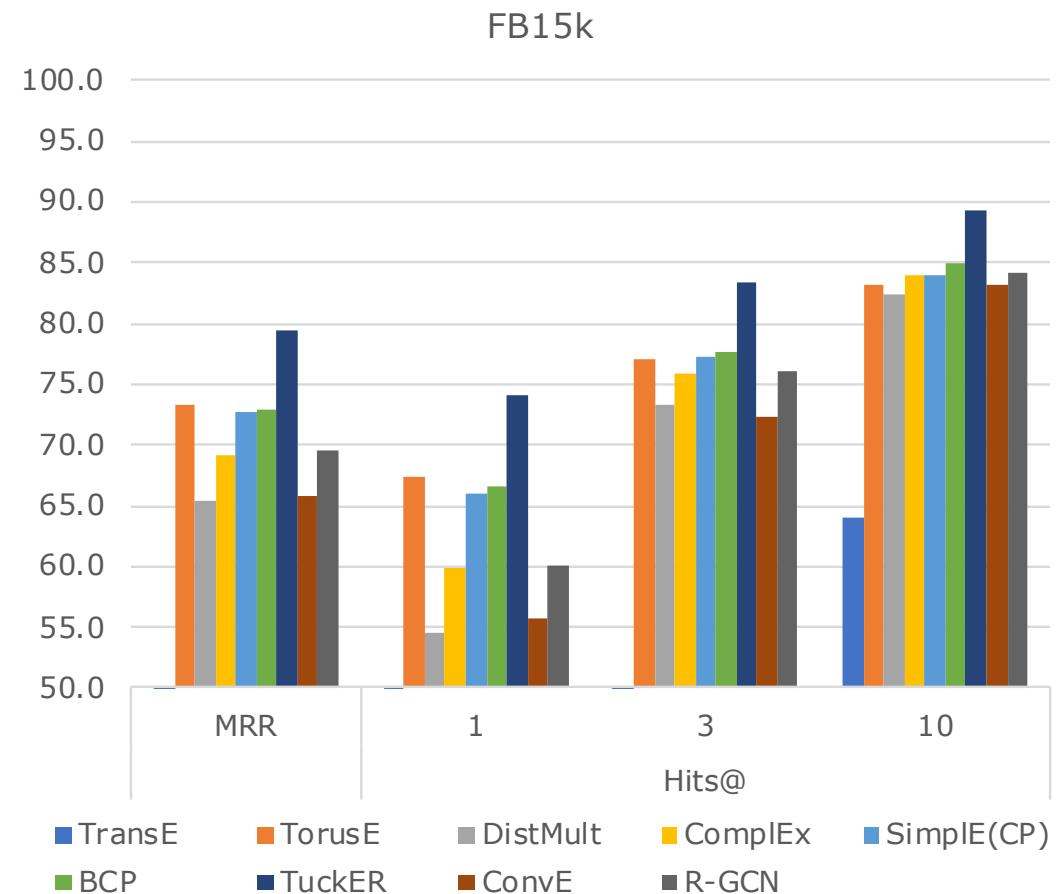
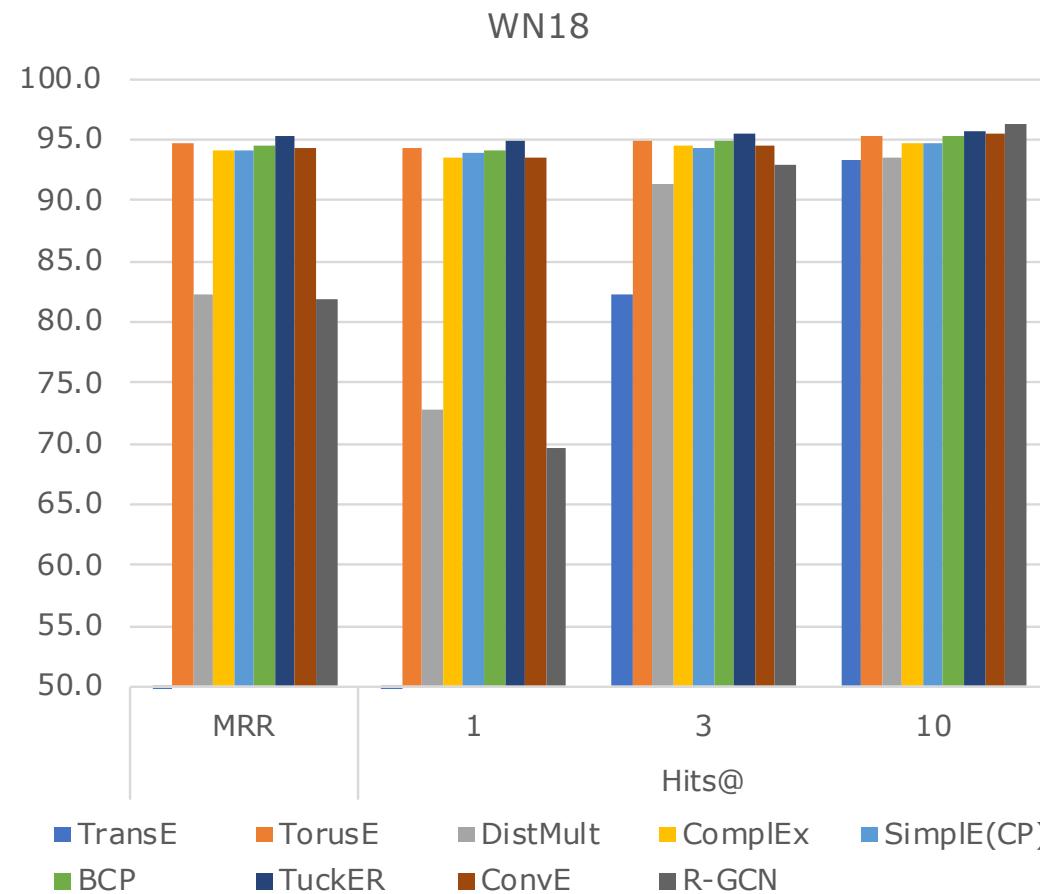
$$(v_i, r, v_j) \in \mathcal{E} \quad r \in \mathcal{R}$$

e.g. (Obama, born\_in, U.S.A.)



$$\mathbf{h}_i^{(l+1)} = \sigma \left( \mathbf{W}_0^{(l)} \mathbf{h}_i^{(l)} + \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{W}_r^{(l)} \mathbf{h}_j^{(l)} \right)$$

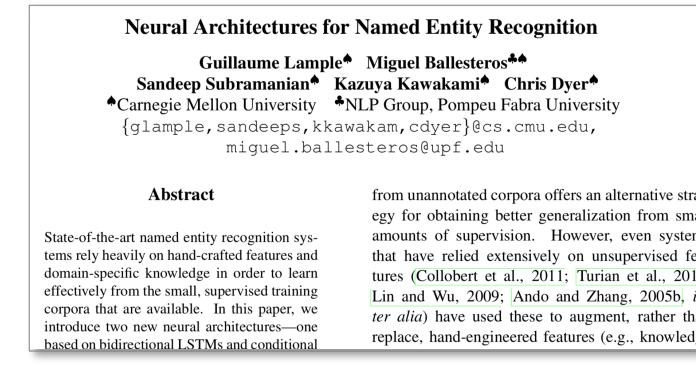
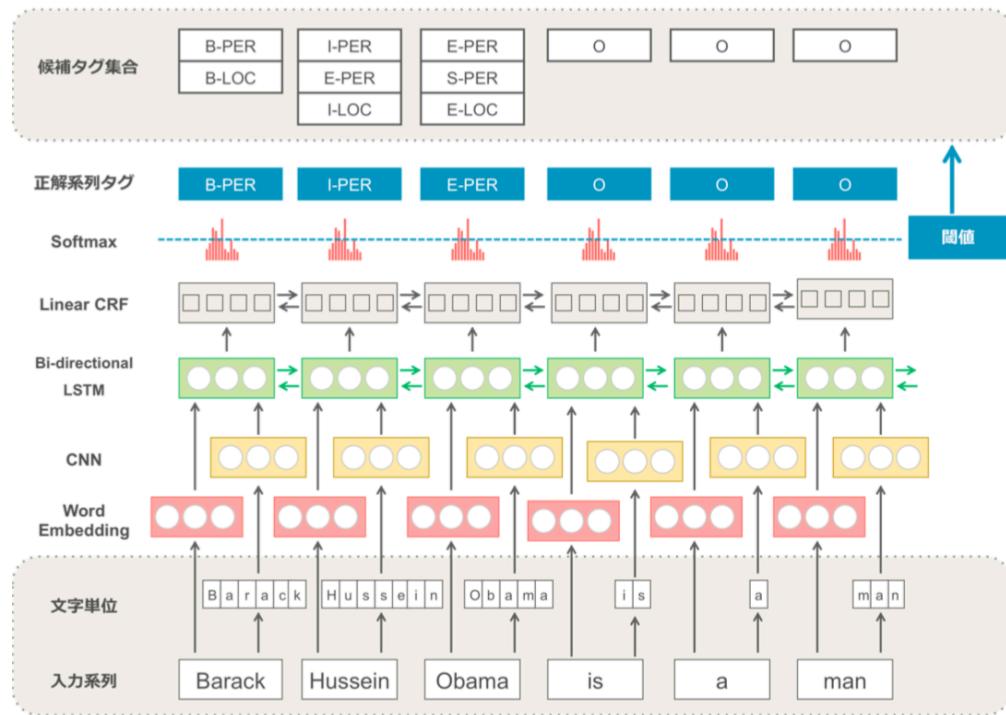
# Link Prediction Results



文献[16]をもとに作成

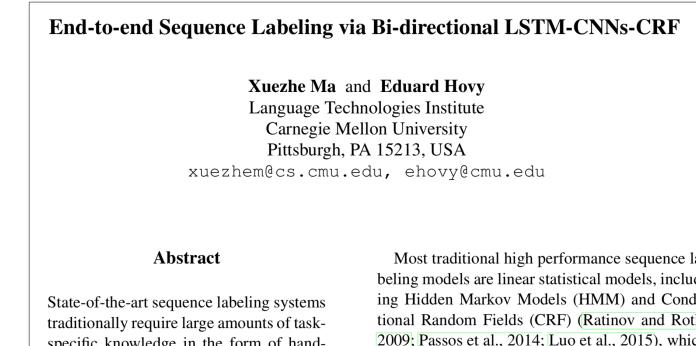
# ニューラル固有表現抽出

- ・単語綴りと分布情報を利用した単語表現による系列学習



[Lample+, 2016]

文字: LSTM  
単語: LSTM



[Ma+, 2016]

文字: CNN  
単語: LSTM

文献[18]をもとに作成

# エンティティリンク

- リンクグラフとword2vecの同時学習で, 単語とエンティティを同一ベクトル空間上にマップ

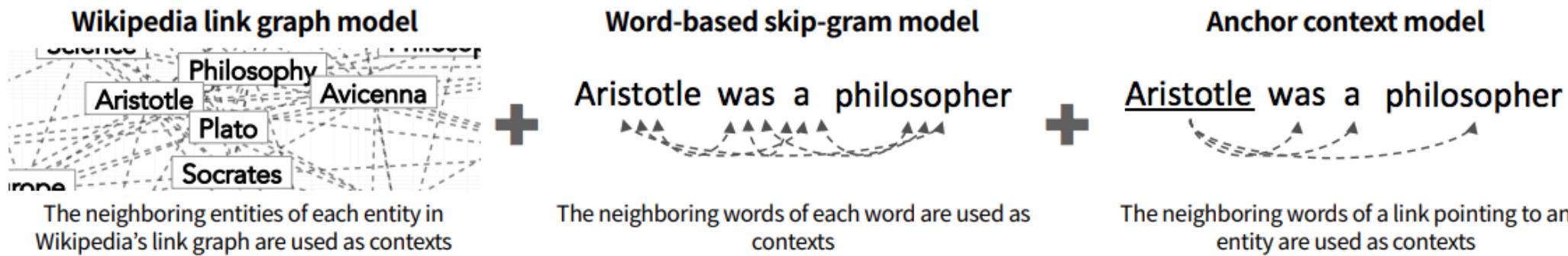


Figure 1: Wikipedia2Vec learns embeddings by jointly optimizing three submodels.

文献[19]をもとに作成

# まとめ

- ・ナレッジグラフ構築は、自然言語処理や機械学習など、複数の技術で構成されたシステム
- ・抽出対象が限定できれば、比較的難易度は低く、品質の高いグラフを作成可能、抽出対象が限定できないと、難易度が高く、必要でノイズも多くなる
- ・技術のトレンドは、論理的な手法から、深層学習の技術を適用したグラフ埋め込みの手法へシフト

# 文献

- [10] Yan, Jihong, et al. "A retrospective of knowledge graphs." *Frontiers of Computer Science* 12.1 (2018): 55-74.
- [11] Pujara, Jay, and Sameer Singh. "Mining Knowledge Graphs From Text." *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018.
- [12] Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel. "Factorizing yago: scalable machine learning for linked data." *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012.
- [13] Nickel, Maximilian, Lorenzo Rosasco, and Tomaso Poggio. "Holographic embeddings of knowledge graphs." *Thirtieth Aaai conference on artificial intelligence*. 2016.
- [14] Q. Wang, Z. Mao, B. Wang and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724-2743, 1 Dec. 2017.

# 文献

- [15] Schlichtkrull, Michael, et al. "Modeling relational data with graph convolutional networks." European Semantic Web Conference. Springer, Cham, 2018.
- [16] Balažević, Ivana, Carl Allen, and Timothy M. Hospedales. "TuckER: Tensor Factorization for Knowledge Graph Completion." arXiv preprint arXiv:1901.09590 (2019).
- [17] Ma, Xuezhe, and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." arXiv preprint arXiv:1603.01354 (2016).
- [18] 佐藤元紀, et al. "ラティス構造とニューラルネットワークに基づく系列ラベリング." 言語処理学会第23回年次大会発表論文集 (2017): 254-257.
- [19] Yamada, Ikuya, et al. "Wikipedia2Vec: An Optimized Tool for Learning Embeddings of Words and Entities from Wikipedia." CoRR (2018).

# 参考

# 転移学習

- ・あるドメイン(データセット)で学習した識別器(特徴抽出器)を他ドメインでの識別器作成に役立てる
- ・2つのアプローチ
  - ・特徴抽出器として利用 (Pre-trained feature)
    - ・学習済NWを特徴抽出器とし,中間層の出力を利用して識別器を作成
  - ・Fine-tuning
    - ・学習済NWを初期値とし,適用先データセットでさらに学習
    - ・所望のタスクを内包するものでなければ効果が薄い (むしろ悪化)

# 2018年10月: BERT の衝撃

BERT: 文献[20]

- タスク特化のNN構造を持たずに入間のスコアを大きく超えた

## SQuAD1.1 Leaderboard

Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of SQuAD v1.1.

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar et al. '16)	82.304	91.221
1	BERT (ensemble) Google AI Language <a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a>	87.433	93.160
2	BERT (single model) Google AI Language <a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a>	85.083	91.835
2	nlnet (ensemble) Microsoft Research Asia	85.356	91.202

<https://rajpurkar.github.io/SQuAD-explorer/>

- 特徴

- 双方向Transformerモデルを大規模コーパスで事前学習し、出力層をタスク毎に1層のみ追加してfine-tuning
- マスク単語予測と次文章判定で事前学習

- 評価

- 11タスクでSOTA (含意、言い換え、文の分類など)
- 機械読解タスク(左)でも、完全一致と部分一致の両指標で最高精度

(2018/10/5)

# 2019年5月: **XLNet** に注目

XLNet: 文献[21]

- BERT の弱点を修正し, 20以上のタスクで BERT を超えた

## SQuAD1.1 Leaderboard

Here are the ExactMatch (EM) and F1 scores evaluated on the test set of SQuAD v1.1.

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar et al. '16)	82.304	91.221
1 May 21, 2019	XLNet (single model) XLNet Team	89.898 <b>XLNet</b>	95.080 87.433 BERT
2 Oct 05, 2018	BERT (ensemble) Google AI Language <a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a>	93.160	

<https://rajpurkar.github.io/SQuAD-explorer/>

- 特徴

- BERTではマスク単語を予測するが,マスクは通常発生しないためノイズにもなる問題を克服

- 評価

- 20タスクでBERTを超えた  
(2018/5/21)
- 機械読解タスク(左)では,  
single モデルで BERT の  
ensemble モデルを超えている

# 文献

- [20] Jacob Devlin, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." (2018)
- [21] Zhilin Yang, et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". (2019)