# Text-enhanced Representation Learning for Knowledge Graph

**Reporter**: Zhigang WANG
**Authors**: Zhigang WANG, Juanzi LI, Zhiyuan LIU, Jie TANG
**Tsinghua University**

To appear in IJCAI 2016
2016-04-17

# Outline

- **Introduction**
- Problem Definition
- Our Proposed Approach
- Experiments and Analysis
- Conclusion

# Representation Learning for KG

## Input

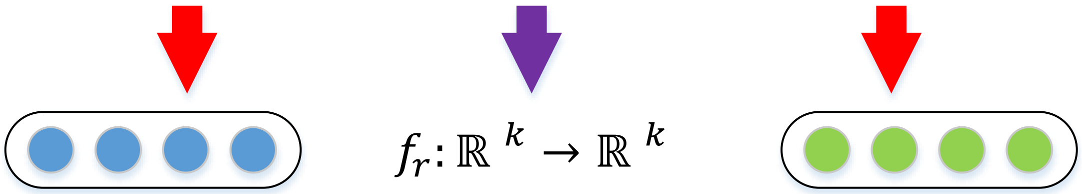
a knowledge graph $\mathcal{KG} = \{(h, r, t)\}$

## Target

- To learn one embedding (a $k$-dimensional vector) for each entity: $h \rightarrow \vec{h}$ and $t \rightarrow \vec{t}$, where $\vec{h}, \vec{t} \in \mathbb{R}^k$

$\boldsymbol{h}$ $\boldsymbol{r}$ $\boldsymbol{t}$

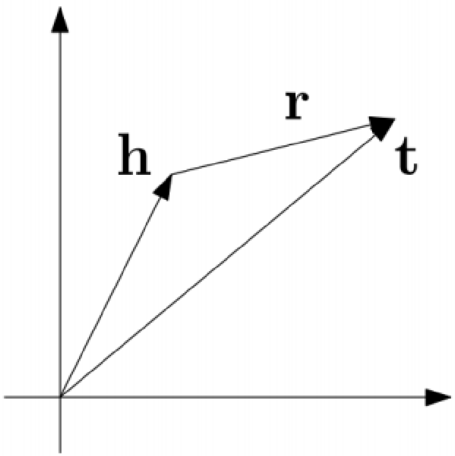
( ***Avatar***, */film/film/directed_by*, ***James Cameron*** )



# Translation-based Methods

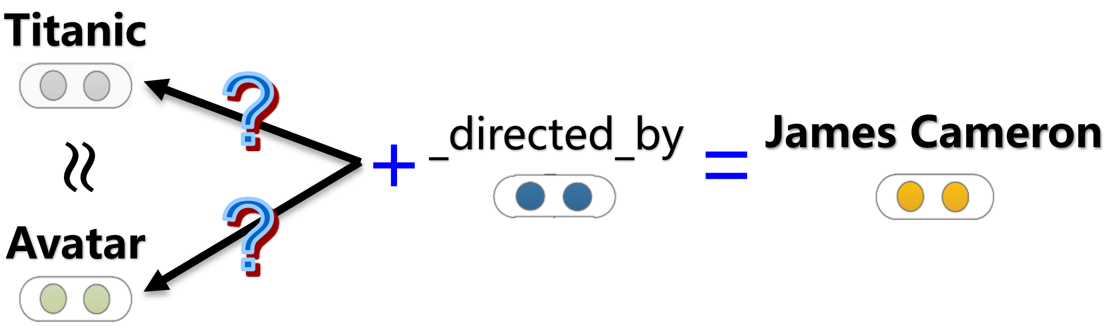
## TransE

- For each triple (head, relation, tail), treat *relation* as a **translation** from head to tail
- Simple, effective, and achieving the state-of-the-art performance




Idea

Issues when modelling **1-to-N**, **N-to-1**, **N-to-N** relations

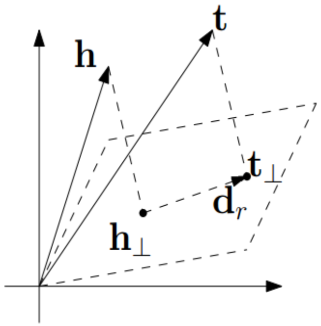



Bordes, et al. (2013). Translating embeddings for modeling multi-relational data. NIPS.

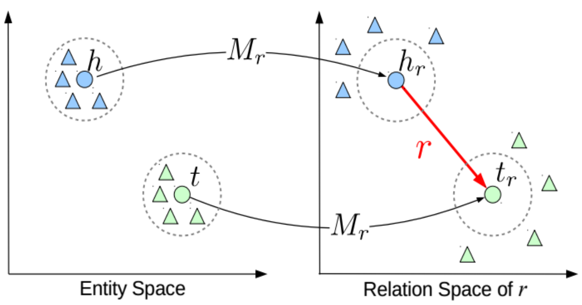
# Translation-based Methods

## TransH and TransR

- Build relation-specific entity embeddings




TransH




TransR

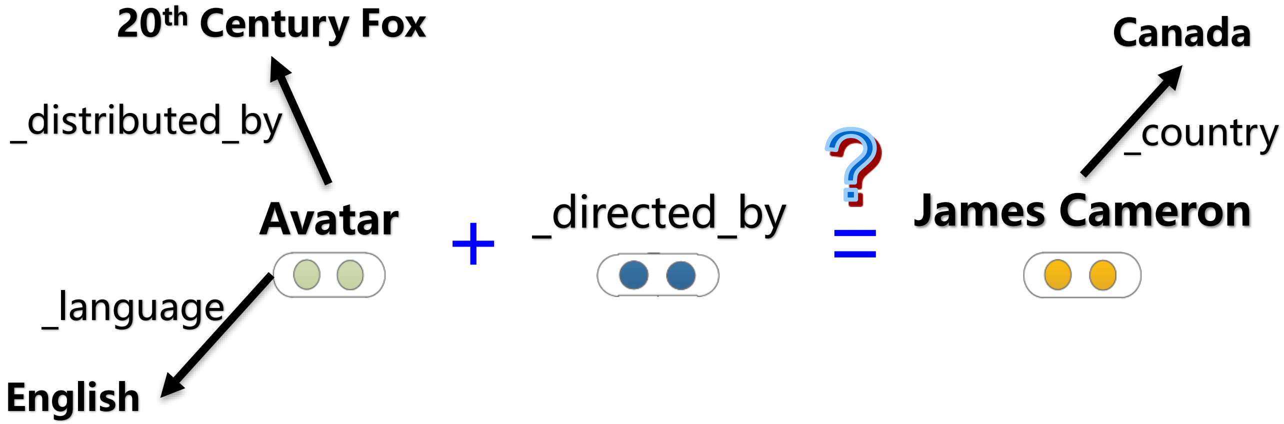
| Tasks | Predicting Head(Hits@10) | | | | Predicting Tail(Hits@10) | | | |
|---|---|---|---|---|---|---|---|---|
| Relation Category | 1-to-1 | 1-to-N | N-to-1 | N-to-N | 1-to-1 | 1-to-N | N-to-1 | N-to-N |
| TransE (Bordes et al. 2013) | 43.7 | 65.7 | 18.2 | 47.2 | 43.7 | 19.7 | 66.7 | 50.0 |
| TransH (unif) (Wang et al. 2014) | 66.7 | 81.7 | 30.2 | 57.4 | 63.7 | 30.1 | 83.2 | 60.8 |
| TransH (bern) (Wang et al. 2014) | 66.8 | 87.6 | 28.7 | 64.5 | 65.5 | 39.8 | 83.3 | 67.2 |
| TransR (unif) | 76.9 | 77.9 | **38.1** | 66.9 | 76.2 | 38.4 | 76.2 | 69.1 |
| TransR (bern) | 78.8 | **89.2** | 34.1 | 69.2 | 79.2 | 37.4 | **90.4** | 72.1 |

**Motivation 1**. low performance on 1-to-N, N-to-1 and N-to-N relations

Wang, et al. (2014). Knowledge graph embedding by translating on hyperplanes. AAAI.
Lin, et al. (2015). Learning entity and relation embeddings for knowledge graph completion. AAAI.

# Translation-based Methods

- **Learn embeddings directly from the graph structure**
  - Graph sparseness
  - In domain-specific and non-English situations




**Motivation 2**. limited performance by the structure sparseness of KG

# Our Idea

- **Text-enhanced Representation Learning for KG**
  - Go back to traditional relation extraction
  - Inspired by distant supervision

Triple: ( ***Avatar***, */film/film/directed_by*, ***James Cameron*** )

Context: {*film*, *movie*, *directed*, ...} ⟶ {*direct*} ⟵ {*director*, ...}

Text:
**James Francis Cameron**, the famous ***director*** of the ***movie*** **Avatar**, is an …
The fiction ***film*** **Avatar** ***directed*** by **J. Cameron** was nominated by …
In 1994 ***director*** **James Cameron** wrote an 80-page treatment for **Avatar**

**Contributions:**
[Motivation 1]. Enable each relation to own different representations for different head and tail entities.
[Motivation 2]. Incorporate the textual contexts to each entity and relation.

# Outline

- Introduction
- **Problem Definition**
- Our Proposed Approach
- Experiments and Analysis
- Conclusion

# Problem Definition

- **Input**
  - Knowledge Graph

$$\mathcal{KG} = \{(h, r, t)\}$$

  - Text Corpus

$$\mathcal{D} = \langle w_1 \dots w_i \dots w_m \rangle$$

- **Text-enhanced Knowledge Embedding (TEKE)**
  - learn the entity embeddings $h \to \vec{h} \in \mathbb{R}^k$ and $t \to \vec{t} \in \mathbb{R}^k$ for each triple $(h, r, t)$ by utilizing the rich text information in $\mathcal{D}$ to deal with
    - low performance on 1-to-N, N-to-1, N-to-N relations
    - knowledge graph sparseness
  - learn the relation embedding $r \to \vec{r} \in \mathbb{R}^k$

# Outline

Introduction

Problem Definition

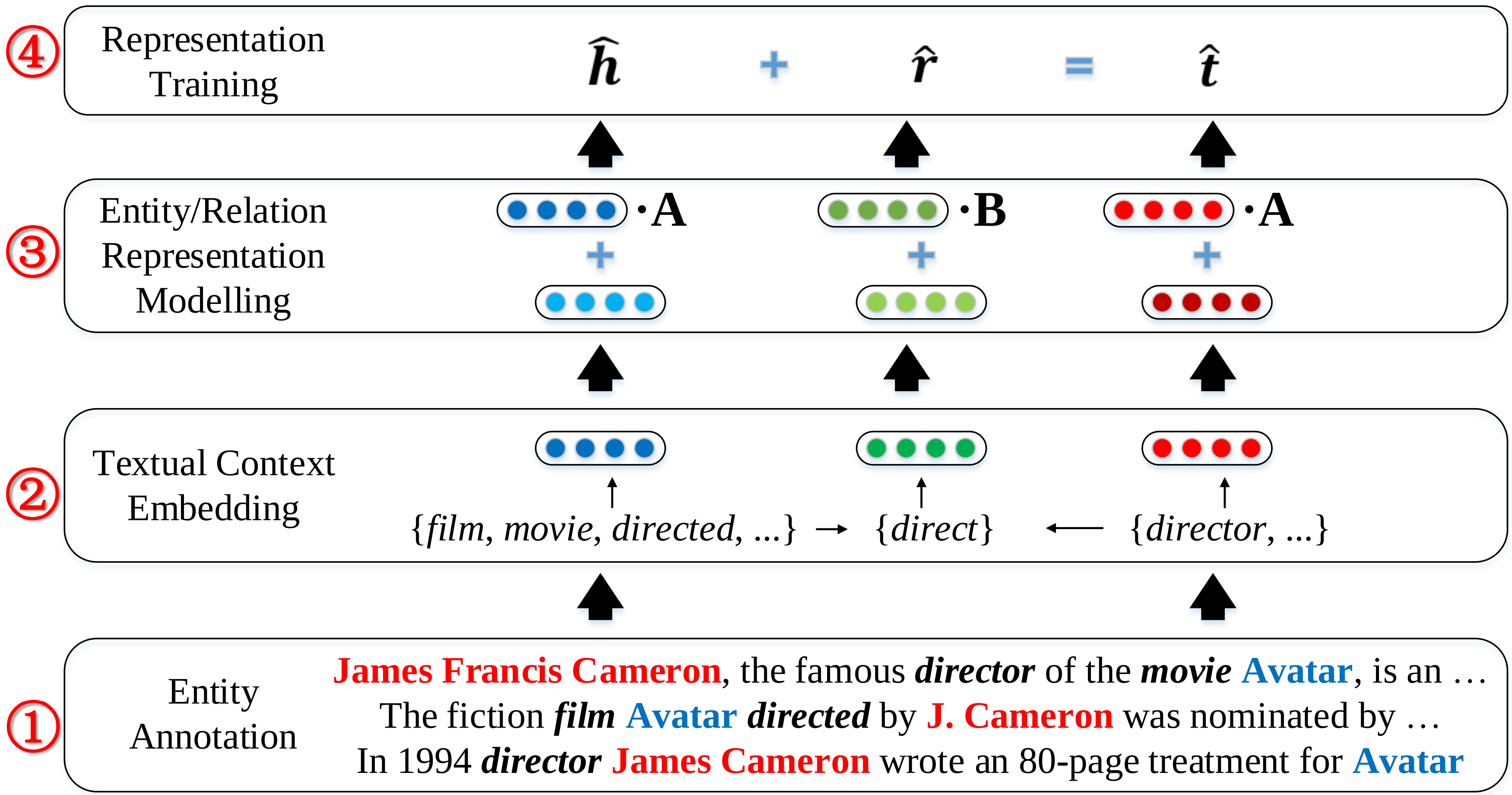
**Our Proposed Approach**

Experiments and Analysis

Conclusion

# The Proposed Approach

Triple: ( ***Avatar***, */film/film/directed_by*, ***James Cameron*** )



# The Proposed Approach

## Entity Annotation

- Given the text corpus $\mathcal{D} = \langle w_1 \dots w_i \dots w_m \rangle$, use an entity linking tool to automatically label the entities in $\mathcal{KG}$, and get an entity-annotated text corpus:

$$\mathcal{D}' = \langle X_1 \dots X_i \dots \mathrm{X}_{m'} \rangle$$

## Textual Context Embedding

- co-occurrence network $\mathcal{G} = (\mathcal{X}, \mathcal{Y})$

  $x_i \in \mathcal{X}$ : denotes to the node (a word or an entity)

  $y_{ij} \in \mathcal{Y}$ : co-occurrence frequency between $x_i$ and $x_j$

# The Proposed Approach

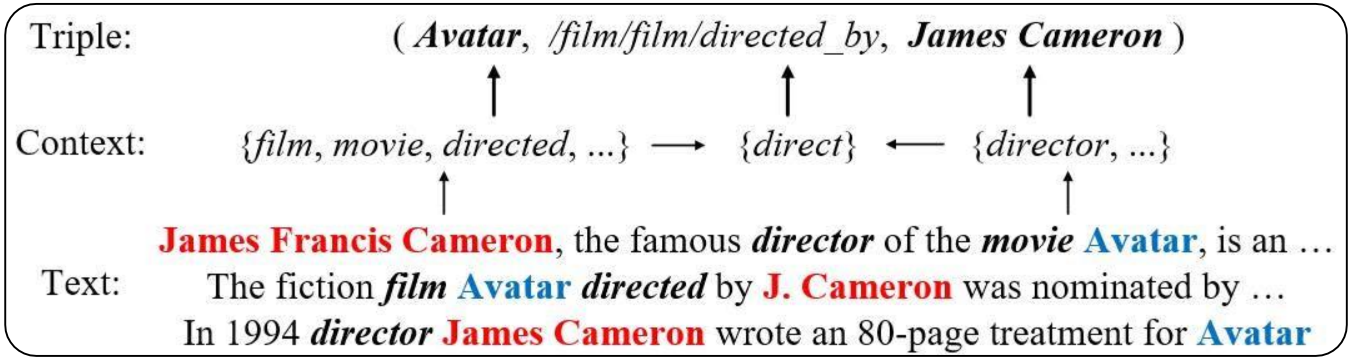
## Textual Context Embedding

- Pointwise textual context

$$\mathrm{n}(x_i) = \{x_j | y_{ij} > \theta\}$$
$$\mathrm{n}(\boldsymbol{Avatar}) = \{film, movie, directed \ldots\}$$
$$\mathrm{n}(\boldsymbol{James\_Cameron}) = \{director \ldots\}$$

- Pairwise textual context

$$\mathrm{n}(x_i, x_j) = \{x_k | x_k \in \mathrm{n}(x_i) \cap \mathrm{n}(x_j)\}$$
$$\mathrm{n}(\boldsymbol{Avatar}, \boldsymbol{James\_Cameron}) = \{direct \ldots\}$$



# The Proposed Approach

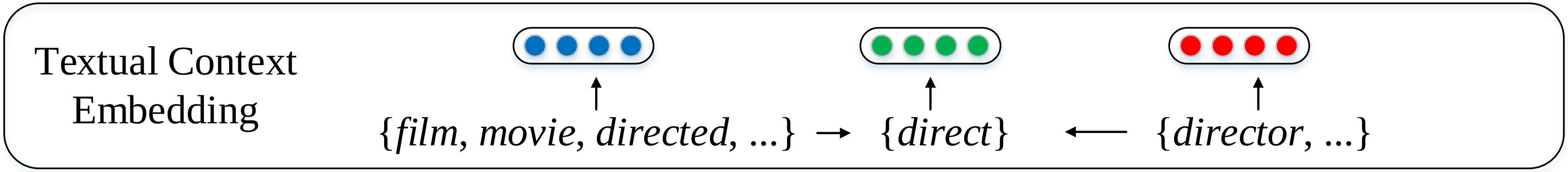
## Textual Context Embedding

- Word Embedding Learning $x_i \rightarrow \boldsymbol{x}_i$
- Pointwise textual context embedding of $x_i$:

$$\boldsymbol{n}(x_i) = \frac{1}{\sum_{x_j \in \mathrm{n}(x_i)} y_{ij}} \sum_{x_j \in \mathrm{n}(x_i)} y_{ij} \cdot \boldsymbol{x}_j$$

- Pairwise textual context embedding of $x_i$ and $x_j$:

$$\boldsymbol{n}(x_i, x_j) = \frac{1}{Z} \sum_{x_k \in \mathrm{n}(x_i, x_j)} min(y_{ik}, y_{jk}) \cdot \boldsymbol{x}_k$$



# The Proposed Approach

## Entity/Relation Representation Modeling

- Incorporate the textual context information to the representation learning on knowledge graph

- $\widehat{\boldsymbol{h}} = \boxed{\boldsymbol{n}(h)}\boldsymbol{A} + \boxed{\boldsymbol{h}}$
- $\hat{\boldsymbol{t}} = \boxed{\boldsymbol{n}(t)}\boldsymbol{A} + \boxed{\boldsymbol{t}}$
- $\hat{\boldsymbol{r}} = \boxed{\boldsymbol{n}(h,t)}\boldsymbol{B} + \boxed{\boldsymbol{r}}$

**Linear transformation of textual context information**

- given a relation, different textual context embeddings for different pairs of (head, tail) entities
- to better handle 1-to-N, N-to-1, and N-to-N relations

**Incorporate textual context information into the KG**

- more background information
- to deal with knowledge graph sparseness

- $f(h,r,t) = \left\|\widehat{\boldsymbol{h}} + \hat{\boldsymbol{r}} - \hat{\boldsymbol{t}}\right\|_2^2$

# The Proposed Approach

## Representation Training

- Margin-based score function

$$L = \sum_{(h,r,t)\in\mathcal{S}} \sum_{(h',r,t')\in\mathcal{S}'} \max(0, f(h,r,t) + \gamma - f(h',r,t'))$$

- Stochastic gradient descent (SGD)

# Outline

Introduction

Problem Definition

Our Proposed Approach

**Experiments and Analysis**

Conclusion

# Experiments and Analysis

## Datasets

- 4 benchmark knowledge graphs

Table 1: Statistics of the data sets.

| Dataset | #Relations | #Entities | #Triples(Train/Valid/Test) | | |
|---|---|---|---|---|---|
| WN18 | 18 | 40,943 | 141,442 | 5,000 | 5,000 |
| FB15K | 1,345 | 14,951 | 483,142 | 50,000 | 59,071 |
| WN11 | 11 | 38,696 | 112,581 | 2,609 | 10,544 |
| FB13 | 13 | 75,043 | 316,232 | 5,908 | 23,733 |

- Entity-annotated Wikipedia corpuses

Table 2: Statistics of entity-annotated Wikipedia corpuses.

| KG | #Entities | #Annotated Entities | #Word Stems |
|---|---|---|---|
| WN18 | 40,943 | 32249 | 1,529,251 |
| FB15K | 14,951 | 14,405 | 744,983 |
| WN11 | 38,696 | 30,937 | 1,526,467 |
| FB13 | 75,043 | 69,208 | 706,484 |

# Experiments and Analysis

## Evaluation

- (China, /location/location/adjoin, North_Korea)

- Link Prediction
  - Mean Rank: 11
  - Hits@10: 0%
  - Raw; Filter: 9; 100%

- Triple Classification
  - a binary classification task

| **Head** | **China** |
|---|---|
| **Relation** | **/location/location/adjoin** |
| 1 | Japan |
| 2 | Taiwan |
| 3 | Israel |
| 4 | South_Korea |
| 5 | Argentina |
| 6 | France |
| 7 | Philippines |
| 8 | Hungary |
| 9 | Germany |
| 10 | USA |
| 11 | **North_Korea** |

# Link Prediction

## TEKE compare with baselines

Table 3: Experimental Results on Link Prediction.

| Datasets | WN18 | | | | FB15K | | | |
|---|---|---|---|---|---|---|---|---|
| Metric | Mean Rank | | Hits@10 (%) | | Mean Rank | | Hits@10 (%) | |
| | Raw | Filter | Raw | Filter | Raw | Filter | Raw | Filter |
| TransE / TEKE_E | 263 / **140** | 251 / **127** | 75.4 / **80.0** | 89.2 / **93.8** | 243 / **233** | 125 / **79** | 34.9 / **43.5** | 47.1 / **67.6** |
| TransH / TEKE_H unif | 318 / **142** | 303 / **128** | 75.4 / **79.7** | 86.7 / **93.6** | **211** / 228 | 84 / **75** | 42.5 / **44.9** | 58.5 / **70.4** |
| TransH / TEKE_H bern | 401 / **127** | 388 / **114** | 73.0 / **80.3** | 82.3 / **92.9** | 212 / **212** | **87** / 108 | 45.7 / **51.2** | 64.4 /**73.0** |
| TransR / TEKE_R unif | 232 / **203** | 219 / **203** | 78.3 / **78.4** | 91.7 / **92.3** | **226** / 237 | **78** / 79 | 43.8 / **44.3** | 65.5 / **68.5** |
| TransR / TEKE_R bern | 238 / **197** | 225 / **193** | **79.8** / 79.4 | **92.0** / 91.8 | **198** / 218 | **77** / 109 | 48.2 / **49.7** | 68.7 / **71.9** |

- A lower Mean Rank is better while a higher Hits@10 is better
- Mean Rank
  - TEKE methods perform much better than the baselines on WN18.
  - No much improvement is observed on FB15K
- Hits@10
  - TEKE methods outperform other baselines significantly and consistently

# Link Prediction

- **Capability to handle 1-to-N, N-to-1 and N-to-N relations**
  - FB15K: 1-1, 1-N, N-1, N-N → 24.2%, 22.9%, 28.9%, 24.0%

Table 4: Experimental Results on FB15K by Mapping Properties of Relations. (%)

| Tasks | Prediction Head (Hits@10) | | | | Prediction Tail (Hits@10) | | | |
|---|---|---|---|---|---|---|---|---|
| Relation Category | 1-to-1 | 1-to-N | **N-to-1** | **N-to-N** | 1-to-1 | **1-to-N** | N-to-1 | **N-to-N** |
| TransE/TEKE_E | 43.7 / **48.9** | 65.7 / **72.1** | 18.2 / **52.3** | 47.2 / **76.8** | 43.7 / **46.3** | 19.7 / **50.2** | 66.7 / **75.3** | 50.0 / **76.1** |
| TransH/TEKE_H unif | **66.7** / 66.6 | **81.7** / 80.9 | 30.2 / **58.0** | 57.4 / **79.6** | **63.7** / 60.5 | 30.1 / **60.4** | **83.2** / 81.5 | 60.8 / **80.2** |
| TransH/TEKE_H bern | 66.8 / **69.3** | 87.6 / **90.8** | 28.7 / **54.1** | 64.5 / **82.0** | **65.5** / 60.7 | 39.8 / **61.5** | 83.3 / **88.3** | 67.2 / **82.1** |
| TransR/TEKE_R unif | **76.9** / 66.2 | 77.9 / **82.0** | 38.1 / **57.0** | 66.9 / **81.3** | **76.2** / 62.5 | 38.4 / **57.5** | 76.2 / **83.1** | 69.1 / **81.2** |
| TransR/TEKE_R bern | **78.8** / 70.1 | 89.2 / **89.3** | 34.1 / **54.0** | 69.2 / **81.7** | **79.2** / 69.6 | 37.4 / **59.2** | **90.4** / 89.2 | 72.1 / **83.5** |

  - TEKE methods significantly outperform the baselines when predicting the entity where multiple entities could be correct.
  - TEKE methods have not shown much advantage for predicting the entity where only one entity is correct.

# Link Prediction

## Capability to handle knowledge graph sparseness

Table 5: Datasets with different densities.

| Dataset | $\#\mathcal{E}$ | $\#\mathcal{R}$ | $\#\mathcal{T}$ | $\#\mathcal{T}/\#\mathcal{E}$ | $\#\mathcal{T}/\#\mathcal{R}$ |
|---|---|---|---|---|---|
| FB3K | 3,000 | 613 | 19,339 | 6.45 | 31.55 |
| FB6K | 6,000 | 913 | 75,347 | 12.56 | 82.53 |
| FB9K | 9,000 | 1,094 | 167,191 | 18.58 | 152.83 |

$\mathcal{T}$ represents the training triples.

- Rank 3,000 entities for 2,238 triples for all three datasets

Table 6: Mean Rank Comparison.

| Methods | TransE / TEKE_E | | | |
|---|---|---|---|---|
| Metric | Raw | | Filter | |
| FB3K | 102.7 | **94.9** | 41.7 | **34.8** |
| FB6K | 81.9 | **78.1** | 29.8 | **25.6** |
| FB9K | 79.5 | **77.0** | 27.6 | **24.7** |

- As the graph density gets higher, both TransE and TEKE_E perform better.
- TEKE_E achieves the highest improvement on the sparsest FB3K dataset.

# Triple Classification

- **TEKE compare with baselines**

Table 7: Evaluation results of triple classification. (%)

| Datasets | WN11 | FB13 |
|---|---|---|
| TransE / TEKE_E unif | 75.9 / **84.1** | 70.9 / **75.1** |
| TransE / TEKE_E bern | 75.9 / **84.5** | 81.5 / **82.1** |
| TransH / TEKE_H unif | 77.7 / **84.3** | 76.5 / **77.4** |
| TransH / TEKE_H bern | 78.8 / **84.8** | 83.3 / **84.2** |
| TransR / TEKE_R unif | **85.5** / 85.2 | 74.7 / **77.1** |
| TransR / TEKE_R bern | 85.9 / **86.1** | **82.5** / 81.6 |

- TEKE_E and TEKE_H consistently outperform the comparison methods, especially on WN11.
- TEKE_R (unif) on WN11 and TEKE_R (bern) on FB13 perform better than TransR, while others perform a bit worse.

# Outline

- Introduction
- Problem Definition
- Our Proposed Approach
- Experiments and Analysis
- **Conclusion**

# Conclusion and Future Work

- A novel text-enhanced knowledge embedding method named **TEKE** for knowledge graph representation learning to deal with
  - Low performance on 1-to-N, N-to-1 and N-to-N relations
  - Limited performance by structure sparseness of KG

- Future Work
  - Improve performance on 1-to-1 relations
  - Experimentally analyze the influence of entity annotation
  - Use different text corpus
  - Incorporate knowledge reasoning
  - ...

# Thanks!

**Zhigang WANG**
wangzigo@gmail.com
http://xlore.org/

# TransE

## Idea

vec('Paris') - vec('France') ≅ vec('Rome') - vec('Italy')

_has_capital _has_capital

- Treat each relation as one unique vector

vec('_has_capital') = vec('_ has_capital')

- and it would be reasonable that

vec('Paris') - vec('France') ≅ vec('Rome') - vec('Italy') = vec('_capital_of')

$\vec{t}$ $\vec{h}$ $\vec{t}$ $\vec{h}$ $\vec{r}$

**Assumption $\vec{t}$ - $\vec{h}$ = $\vec{r}$**