Knowledge Graph Representation

From Recent Models towards a Theoretical Understanding

Ivana Balažević & Carl Allen

January 27, 2021

School of Informatics, University of Edinburgh
What are Knowledge Graphs?

Entities $\mathcal{E} = \{A, B, C, D\}$

Relations $\mathcal{R} = \{\text{married to}, \text{father of}, \text{uncle of}, \ldots\}$

Knowledge Graph $\mathcal{G} = \{(A, \text{father of}, B), (A, \text{married to}, C), \ldots\}$
Representing Entities and Relations

Subject and object entities $e_s, e_o$ are represented by vectors $e_s, e_o \in \mathbb{R}^d$ (embeddings).
Representing Entities and Relations

Subject and object entities \( e_s, e_o \) are represented by vectors \( e_s, e_o \in \mathbb{R}^d \) (embeddings).

Relations \( r \) are represented by transformations \( f_r, g_r : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} \) that transform the entity embeddings.
Representing Entities and Relations

Subject and object entities $e_s, e_o$ are represented by vectors $e_s, e_o \in \mathbb{R}^d$ (embeddings).

Relations $r$ are represented by transformations $f_r, g_r : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ that transform the entity embeddings.

A proximity measure, e.g. Euclidean distance, dot product, compares the transformed subject and object entities.
Subject and object entities $e_s, e_o$ are represented by vectors $e_s, e_o \in \mathbb{R}^d$ (embeddings).

Relations $r$ are represented by transformations $f_r, g_r : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ that transform the entity embeddings.

A proximity measure, e.g. Euclidean distance, dot product, compares the transformed subject and object entities.

(Edinburgh, capital_of, Scotland)
A **score function** \( \phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R} \) brings together entity, relation representations and proximity measure to assign a score \( \phi(e_s, r, e_o) \) to each triple, used to predict whether the triple is true or false.
A **score function** \( \phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R} \) brings together entity, relation representations and proximity measure to assign a score \( \phi(e_s, r, e_o) \) to each triple, used to predict whether the triple is true or false.

Representation parameters are optimised to improve prediction accuracy.
A **score function** \( \phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R} \) brings together entity, relation representations and proximity measure to assign a score \( \phi(e_s, r, e_o) \) to each triple, used to predict whether the triple is true or false.

Representation parameters are optimised to improve prediction accuracy.

Score functions can be broadly categorised by:

- relation representation type (additive, multiplicative or both); and
- proximity measure (e.g. dot product, Euclidean distance).
A score function \( \phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R} \) brings together entity, relation representations and proximity measure to assign a score \( \phi(e_s, r, e_o) \) to each triple, used to predict whether the triple is true or false.

Representation parameters are optimised to improve prediction accuracy.

Score functions can be broadly categorised by:

- relation representation type (additive, multiplicative or both); and
- proximity measure (e.g. dot product, Euclidean distance).

<table>
<thead>
<tr>
<th>Rel. Repr. Type</th>
<th>Example ( \phi(e_s, r, e_o) )</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicative</td>
<td>( e_s^T W_r e_o = \langle e_s^{(r)}, e_o \rangle )</td>
<td>DistMult (Yang et al., 2015) TuckER (Balažević et al., 2019b)</td>
</tr>
<tr>
<td>Additive</td>
<td>(-|e_s + r - e_o|^2)</td>
<td>TransE (Bordes et al., 2013)</td>
</tr>
<tr>
<td>Both</td>
<td>(-|e_s^T W_r e_o + r - e_o^T W_r^o|^2 + b_s + b_o)</td>
<td>MuRE (Balažević et al., 2019a)</td>
</tr>
</tbody>
</table>
Figure 1: Visualization of the TuckER architecture.

\[ \phi_{\text{TuckER}}(e_s, r, e_o) = ((\mathcal{W} \times_1 w_r) \times_2 e_s) \times_3 e_o = e_s^T W_r e_o \]
Figure 1: Visualization of the TuckER architecture.

\[ \phi_{\text{TuckER}}(e_s, r, e_o) = ((\mathcal{W} \times_1 w_r) \times_2 e_s) \times_3 e_o = e_s^\top \mathbf{W_r} e_o \]

**Multi-task learning:** Rather than learning distinct relation matrices \( \mathbf{W}_r \), the core tensor \( \mathcal{W} \) contains a shared pool of “prototype” relation matrices that are linearly combined using parameters of the relation embedding \( \mathbf{w}_r \).

(Balažević et al., 2019a)
Figure 2: MuRE spheres of influence.

\[ \phi_{\text{MuRE}} = -d(\text{Re}_s, \mathbf{e}_o + \mathbf{r})^2 + b_s + b_o \]

(Balažević et al., 2019b)
Recap

- KGs store facts: binary relations between entities ($e_s, r, e_o$).
Recap

- KGs store facts: binary relations between entities \((e_s, r, e_o)\).

- Enable computational reasoning over KGs, e.g. question answering and inferring new facts (link prediction).
Recap

- KGs store facts: binary \textit{relations} between \textit{entities} \((e_s, r, e_o)\).

- Enable computational reasoning over KGs, e.g. question answering and inferring new facts (\textit{link prediction}).

- Requires representation, typically:
  - each entity by a vector \textit{embedding} \(e \in \mathbb{R}^d\),
  - each relation by a \textit{transformation} from subject entity to object entity,

Many, many models with increasing success, but no principled rationale as to why they work, or how to improve (e.g. better prediction, incorporate logic, etc).
Recap

- KGs store facts: binary relations between entities \((e_s, r, e_o)\).

- Enable computational reasoning over KGs, e.g. question answering and inferring new facts (link prediction).

- Requires representation, typically:
  - each entity by a vector embedding \(e \in \mathbb{R}^d\),
  - each relation by a transformation from subject entity to object entity,

Many, many models with increasing success, but no principled rationale as to why they work, or how to improve (e.g. better prediction, incorporate logic, etc).
Recap

- KGs store facts: binary relations between entities \((e_s, r, e_o)\).

- Enable computational reasoning over KGs, e.g. question answering and inferring new facts (link prediction).

- Requires representation, typically:
  - each entity by a vector embedding \(e \in \mathbb{R}^d\),
  - each relation by a transformation from subject entity to object entity,

Many, many models with increasing success, but no principled rationale as to why they work, or how to improve (e.g. better prediction, incorporate logic, etc).
Recap

- KGs store facts: binary relations between entities \((e_s, r, e_o)\).

- Enable computational reasoning over KGs, e.g. question answering and inferring new facts (link prediction).

- Requires representation, typically:
  - each entity by a vector embedding \(e \in \mathbb{R}^d\),
  - each relation by a transformation from subject entity to object entity,

- Many, many models with gradually increasing success, but no principled rationale for why they work, or how to improve them (e.g. more accurate prediction, incorporate logic, etc).
Simplify: consider Word Embeddings

- Word embeddings, e.g.
  - **Word2Vec** (W2V, Mikolov et al., 2013)
  - **GloVe** (Pennington et al., 2014)

---

### Observation:
- Semantic relations between words $\Rightarrow$ geometric relationships between embeddings
- Similar words $\Rightarrow$ close embeddings
- Analogies (often) $\Rightarrow$ e.g., $\text{king} - \text{man} + \text{woman} \approx \text{queen}$

### Aim:
Relate the understanding of this to knowledge graph relations
Simplify: consider Word Embeddings

- Word embeddings, e.g.
  - **Word2Vec** (W2V, Mikolov et al., 2013)
  - **GloVe** (Pennington et al., 2014)

- Observation: **semantic** relations between words $\implies$ **geometric** relationships between embeddings
Simplify: consider Word Embeddings

- Word embeddings, e.g.
  - **Word2Vec** (W2V, Mikolov et al., 2013)
  - **GloVe** (Pennington et al., 2014)

- Observation: **semantic** relations between words \( \Rightarrow \) **geometric** relationships between embeddings
  - similar words \( \Rightarrow \) close embeddings
Simplify: consider Word Embeddings

➤ Word embeddings, e.g.
  • **Word2Vec** (W2V, Mikolov et al., 2013)
  • **GloVe** (Pennington et al., 2014)

➤ Observation: **semantic** relations between words $\Rightarrow$ **geometric** relationships between embeddings
  - **similar** words $\Rightarrow$ close embeddings
  - **analogies** (often) $\Rightarrow$

\[
\begin{align*}
\mathbf{w}_{\text{king}} + \mathbf{w}_{\text{man}} & \approx \mathbf{w}_{\text{queen}} \\
\mathbf{w}_{\text{woman}} \approx \mathbf{w}_{\text{king}} - \mathbf{w}_{\text{man}}
\end{align*}
\]
Simplify: consider Word Embeddings

- **Word embeddings**, e.g.
  - **Word2Vec** (W2V, Mikolov et al., 2013)
  - **GloVe** (Pennington et al., 2014)

- **Observation:** semantic relations between words $\Rightarrow$ geometric relationships between embeddings
  - similar words $\Rightarrow$ close embeddings
  - analogies (often) $\Rightarrow$

- **Aim:** relate the understanding of this to knowledge graph relations
Understanding word embeddings: the W2V Loss Function

\[-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(w_i^\top c_j) + \frac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-w_i^\top c_j))\]
Understanding word embeddings: the W2V Loss Function

\[-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(w_i^\top c_j) + \frac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-w_i^\top c_j))\]

\[\nabla_{w_i} \ell_{W2V} \propto \sum_j \left\{ p(w_i, c_j) + kp(w_i)p(c_j) \right\} \left\{ \sigma(S_{i,j}) - \sigma(w_i^\top c_j) \right\} c_j = C \text{ diag}(d^{(i)}) e^{(i)} \]
Understanding word embeddings: the W2V Loss Function

\[-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(w_i^\top c_j) + \frac{k \#(w_i) \#(c_j)}{D} \log(\sigma(-w_i^\top c_j))\]

\[\nabla_{w_i} \ell_{W2V} \propto \sum_j \left\{ p(w_i, c_j) + kp(w_i) p(c_j) \right\} \left\{ \sigma(S_{i,j}) - \sigma(w_i^\top c_j) \right\} c_j = C \text{diag}(d^{(i)}) e^{(i)}\]

- \(\ell_{W2V}\) minimised when:

**low-rank case:**  
\[
w_i^\top c_j = \log \frac{p(c_j|w_i)}{p(c_j)} - \log k \doteq S_{i,j}\]

\((\text{Levy and Goldberg, 2014})\)
Understanding word embeddings: the W2V Loss Function

\[-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log (w_i^\top c_j) + \frac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-w_i^\top c_j))\]

\[\nabla_{w_i} \ell_{W2V} \propto \sum_j \{ p(w_i, c_j) + kp(w_i)p(c_j) \} \{ \sigma(S_{i,j}) - \sigma(w_i^\top c_j) \} c_j = C \text{diag}(d^{(i)}_j)e^{(i)} \]

- \(\ell_{W2V}\) minimised when:

  **low-rank case:** \(w_i^\top c_j = \log \frac{p(c_j|w_i)}{p(c_j)} - \log k \doteq S_{i,j}\)  
  (Levy and Goldberg, 2014)

  **general case:** error vectors \(\text{diag}(d^{(i)}_j)e^{(i)}\) orthogonal to rows of \(C\)
- $\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(w_i^T c_j) + \frac{k \#(w_i) \#(c_j)}{D} \log(\sigma(-w_i^T c_j))$

\[ \nabla_{w_i} \ell_{W2V} \propto \sum_j \{p(w_i, c_j) + kp(w_i)p(c_j)\} \{\sigma(S_{i,j}) - \sigma(w_i^T c_j)\} c_j = C \text{ diag}(d^{(i)})e^{(i)} \]

- $\ell_{W2V}$ minimised when:
  - **low-rank case:** $w_i^T c_j = \log \frac{p(c_j | w_i)}{p(c_j)} - \log k = S_{i,j}$ (Levy and Goldberg, 2014)
    - PMI($w_i, c_j$)
  - **general case:** error vectors $\text{diag}(d^{(i)})e^{(i)}$ orthogonal to rows of $C$

$\Rightarrow$ Embedding $w_i$ is a (non-linear) projection of row $i$ of the PMI matrix*, a **PMI vector** $p^i$.

(* drop $k$ term as artefact of the W2V algorithm.)
$p^i = \{ \log \frac{p(c_j|w_i)}{p(c_j)} \}_{c_j \in \mathcal{E}} = \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E})}$  \hspace{1cm} (\mathcal{E} = \text{dictionary of all words})$

**Figure 3:** The PMI surface $\mathcal{S}$ with example PMI vectors of words (red dots)
**Similarity:** similar words, e.g. synonyms, induce similar distributions, $p(\mathcal{E}|w)$, over context words.
**Similarity:** similar words, e.g. synonyms, induce similar distributions, \( p(\mathcal{E}|w) \), over context words.

Identified by **subtraction** of PMI vectors:

\[
p^i - p^j = \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E}|w_j)} = \rho_{i,j}
\]
**Similarity:** similar words, e.g. synonyms, induce similar distributions, $p(E|w)$, over context words.

Identified by **subtraction** of PMI vectors:

$$p_i - p_j = \log \frac{p(E|w_i)}{p(E|w_j)} = \rho_{i,j}$$
**Paraphrases:** word sets with similar aggregate semantic meaning, e.g. \{man, royal\} \approx king.
Paraphrases: word sets with similar aggregate semantic meaning, e.g. \{man, royal\} \approx \text{king}.

Identified by \textit{addition} of PMI vectors:

\[
p^i + p^j = \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E})} + \log \frac{p(\mathcal{E}|w_j)}{p(\mathcal{E})} = p^k + \log \frac{p(\mathcal{E}|w_i,w_j)}{p(\mathcal{E}|w_k)} - \log \frac{p(w_i,w_j|\mathcal{E})}{p(w_i|\mathcal{E})p(w_j|\mathcal{E})} + \log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}
\]

\[\rho_{\{i,j\},k},\sigma_{i,j},\tau_{i,j}\]

\text{paraphrase error} \quad \text{independence error}
**PMI Vector Interactions = Semantics (Paraphrase)**

**Paraphrases:** word sets with similar aggregate semantic meaning, e.g. \{man, royal\} \(\approx\) king.

Identified by **addition** of PMI vectors:

\[
p^i + p^j = \log \frac{p(\varepsilon|w_i)}{p(\varepsilon)} + \log \frac{p(\varepsilon|w_j)}{p(\varepsilon)} = p^k + \log \frac{p(\varepsilon|w_i, w_j)}{p(\varepsilon|w_k)} - \log \frac{p(w_i, w_j|\varepsilon)}{p(w_i|\varepsilon)p(w_j|\varepsilon)} + \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}
\]

- **Paraphrase error**
- **Independence error**

\(E\) \(\Rightarrow\) \(\{\text{man, royal}\}\) \(\Rightarrow\) \(\text{king}\)
Analogies: word pairs that share a similar semantic difference, e.g. \{man, king\} and \{woman, queen\}. 
**Analogy**: word pairs that share a similar semantic difference, e.g. \{man, king\} and \{woman, queen\}.

Identified by a **linear combination** of PMI vectors:

\[ p_{king} - p_{man} \approx p_{queen} - p_{woman} \]
**Analogies:** word pairs that share a similar semantic difference, e.g. \{man, king\} and \{woman, queen\}.

Identified by a **linear combination** of PMI vectors:

\[ p_{king} - p_{man} \approx p_{queen} - p_{woman} \]

(Allen and Hospedales, 2019; Allen et al., 2019)
From Analogies to Relations

Analogy

Relation

Analogy contains common binary word relations, similar to KGs. For certain analogies (“specialisations”), the associated “vector offset” gives a transformation that represents the relation. Not all relations fit this semantic pattern, but we have insight to consider geometric aspects (relation conditions) of other relation types.
Analogy

Relation

Analogies contain common binary word relations, similar to KGs.
Analogy

Analogy contains common **binary word relations**, similar to KGs.

For certain analogies ("specialisations"), the associated "vector offset" gives a **transformation that represents the relation**.
Analogy contains common binary word relations, similar to KGs.

For certain analogies ("specialisations"), the associated "vector offset" gives a transformation that represents the relation.

Not all relations fit this semantic pattern, but we have insight to consider geometric aspects (relation conditions) of other relation types.
Categorising Relations: semantics → relation requirements

Similarity  Relatedness  Specialisation  Context-shift  Gen. context-shift

Relationships between PMI vectors for different relation types.
blue/green = strong word association (PMI > 0); red = relatedness; black = context sets
Categorising Relations: semantics → relation requirements

- **Similarity**
- **Relatedness**
- **Specialisation**
- **Context-shift**
- **Gen. context-shift**

Relationships between PMI vectors for different relation types.
blue/green = strong word association (PMI > 0); red = relatedness; black = context sets

Categorisation of WN18RR relations.

<table>
<thead>
<tr>
<th>Type</th>
<th>Relation</th>
<th>Examples (subject entity, object entity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>verb_group</td>
<td>(trim_down_VB_1, cut_VB_35), (hatch_VB_1, incubate_VB_2)</td>
</tr>
<tr>
<td></td>
<td>derivationally_related_form</td>
<td>(lodge_VB_4, accommodation_NN_4), (question_NN_1, inquire_VB_1)</td>
</tr>
<tr>
<td></td>
<td>also_see</td>
<td>(clean JJ_1, tidy JJ_1), (ram_VB_2, screw_VB_3)</td>
</tr>
<tr>
<td>S</td>
<td>hypernym</td>
<td>(land_reform_NN_1, reform_NN_1), (prickle-weed_NN_1, herbaceous_plant_NN_1)</td>
</tr>
<tr>
<td></td>
<td>instance_hypernym</td>
<td>(yellowstone_river_NN_1, river_NN_1), (leipzig_NN_1, urban_center_NN_1)</td>
</tr>
<tr>
<td>C</td>
<td>member_of_domain_usage</td>
<td>(colloquialism_NN_1, figure_VB_5), (plural_form_NN_1, authority_NN_2)</td>
</tr>
<tr>
<td></td>
<td>member_of_domain_region</td>
<td>(rome_NN_1, gladiator_NN_1), (usa_NN_1, multiple_voting_NN_1)</td>
</tr>
<tr>
<td></td>
<td>member_meronym</td>
<td>(south_NN_2, sunshine_state_NN_1), (genus_carya_NN_1, pecan_tree_NN_1)</td>
</tr>
<tr>
<td></td>
<td>has_part</td>
<td>(aircraft_NN_1, cabin_NN_3), (morocco_NN_1, atlas_mountains_NN_1)</td>
</tr>
<tr>
<td></td>
<td>synset_domain_topic_of</td>
<td>(quark_NN_1, physics_NN_1), (harmonize_VB_3, music_NN_4)</td>
</tr>
</tbody>
</table>
View PMI vectors as *sets of word features* and *relation types as set operations*:

- similarity ⇒ set equality
- relatedness ⇒ subset equality (relation-specific)
- context-shift ⇒ set difference (relation-specific)
View PMI vectors as *sets of word features* and *relation types as set operations*:

- similarity \(\Rightarrow\) set equality
- relatedness \(\Rightarrow\) subset equality (relation-specific)
- context-shift \(\Rightarrow\) set difference (relation-specific)

For any relation, each feature is either

- necessarily unchanged (relatedness),
- necessarily/potentially changed (context shift), or
- irrelevant.
View PMI vectors as *sets of word features* and *relation types as set operations*:

- similarity \( \Rightarrow \) set equality
- relatedness \( \Rightarrow \) subset equality (relation-specific)
- context-shift \( \Rightarrow \) set difference (relation-specific)

For any relation, each feature is either

- necessarily unchanged (relatedness),
- necessarily/potentially changed (context shift), or
- irrelevant.

**Conjecture:** the relation types identified partition the set of semantic relations.
Relations as mappings between embeddings

**R:**  $S$-relatedness requires both entity embeddings $\mathbf{e}_s, \mathbf{e}_o$ to share a common subspace component $\mathbf{V}_S$

- project onto $\mathbf{V}_S$ (multiply by matrix $\mathbf{P}_r \in \mathbb{R}^{d \times d}$) and compare.
- Dot product: $\mathbf{(P}_r\mathbf{e}_s)^\top\mathbf{(P}_r\mathbf{e}_o) = \mathbf{e}_s^\top \mathbf{P}_r^\top \mathbf{P}_r\mathbf{e}_o = \mathbf{e}_s^\top \mathbf{M}_r\mathbf{e}_o$
- Euclidean distance: $\|\mathbf{P}_r\mathbf{e}_s - \mathbf{P}_r\mathbf{e}_o\|^2 = \|\mathbf{P}_r\mathbf{e}_s\|^2 - 2\mathbf{e}_s^\top \mathbf{M}_r\mathbf{e}_o + \|\mathbf{P}_r\mathbf{e}_o\|^2$

**S/C:** requires $S$-relatedness and relation-specific component(s) ($\mathbf{v}_r^s$, $\mathbf{v}_r^o$).

- project onto a subspace (by $\mathbf{P}_r \in \mathbb{R}^{d \times d}$) corresponding to $S$, $\mathbf{v}_r^s$ and $\mathbf{v}_r^o$ (i.e. test $S$-relatedness while preserving relation-specific components);
- add relation-specific $\mathbf{r} = \mathbf{v}_r^o - \mathbf{v}_r^s \in \mathbb{R}^d$ to transformed embeddings.
- Dot product: $\mathbf{(P}_r\mathbf{e}_s + \mathbf{r})^\top \mathbf{P}_r\mathbf{e}_o$
- Euclidean distance: $\|\mathbf{P}_r\mathbf{e}_s + \mathbf{r} - \mathbf{P}_r\mathbf{e}_o\|^2$ (cf **MuRE**: $\|\mathbf{R}\mathbf{e}_s + \mathbf{r} - \mathbf{e}_o\|^2$)
Theoretic: a derivation of geometric components of relation representations from word co-occurrence statistics.
Summary

- **Theoretic**: a derivation of geometric components of relation representations from word co-occurrence statistics.

- **Interpretability**: associates geometric model components with semantic aspects of relations.

*Note: MuRE was inspired by the vector offset of analogies. Work to appear in ICLR 2021 (Allen et al., 2021).*
Summary

- **Theoretic**: a derivation of geometric components of relation representations from word co-occurrence statistics.

- **Interpretability**: associates geometric model components with semantic aspects of relations.

- **Empirically supported**: justifies relative link-prediction performance of a range of models on real datasets:
Summary

- **Theoretic**: a derivation of geometric components of relation representations from word co-occurrence statistics.

- **Interpretability**: associates geometric model components with semantic aspects of relations.

- **Empirically supported**: justifies relative link-prediction performance of a range of models on real datasets:

  \[ \text{additive & multiplicative} \quad > \quad \text{multiplicative} \quad \text{or} \quad \text{additive} \]

  - MuRE* (Balažević et al., 2019a)
  - TuckER (Balažević et al., 2019b)
  - DistMult (Yang et al., 2015)
  - TransE (Bordes et al., 2013)

*Note: MuRE was inspired by the vector offset of analogies.*
Any questions?


Ivana Balažević, Carl Allen, and Timothy M Hospedales. TuckER: Tensor Factorization for Knowledge Graph Completion. In EMNLP, 2019b.


