Knowledge Graph Representation

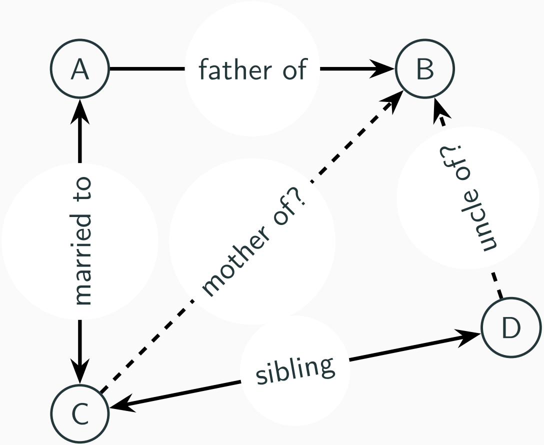
From Recent Models towards a Theoretical Understanding

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	$\mathcal{R} = \{\textit{married to}, \textit{father of}, \textit{uncle of}, ...\}$
Knowledge Graph	$\mathcal{G} = \{(A, \textit{father of}, B), (A, \textit{married to}, C), ...\}$

Representing Entities and Relations

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

Representing Entities and Relations

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

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

Representing Entities and Relations

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

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

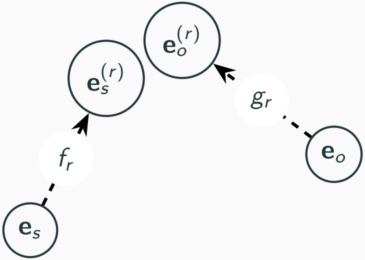
A **proximity measure**, e.g. Euclidean distance, dot product, compares the transformed subject and object entities.

Representing Entities and Relations

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

Relations r are represented by **transformations** f_r, $g_r : \mathbb{R}^d \to \mathbb{R}^{d'}$ 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)

Score Function

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.

Score Function

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 Function

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).

Score Function

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).

Rel. Repr. Type	**Example $\phi(e_s, r, e_o)$**	**Model**
Multiplicative	$\mathbf{e}_s^\top \mathbf{W}_r \mathbf{e}_o = \langle \mathbf{e}_s^{(r)}, \mathbf{e}_o \rangle$	DistMult (Yang et al., 2015) TuckER (Balažević et al., 2019b)
Additive	$-\|\mathbf{e}_s + \mathbf{r} - \mathbf{e}_o\|^2$	TransE (Bordes et al., 2013)
Both	$-\|\mathbf{e}_s^\top \mathbf{W}_r^s + \mathbf{r} - \mathbf{e}_o^\top \mathbf{W}_r^o\|^2 + b_s + b_o$	MuRE (Balažević et al., 2019a)

TuckER: Tensor Factorization for Knowledge Graph Completion

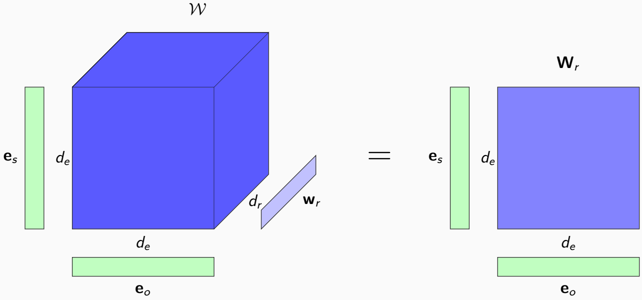


Figure 1: Visualization of the TuckER architecture.

$$\phi_{\text{TuckER}}(e_s, r, e_o) = ((\mathcal{W} \times_1 \mathbf{w}_r) \times_2 \mathbf{e}_s) \times_3 \mathbf{e}_o = \mathbf{e}_s^\top \mathbf{W}_r \mathbf{e}_o$$

TuckER: Tensor Factorization for Knowledge Graph Completion

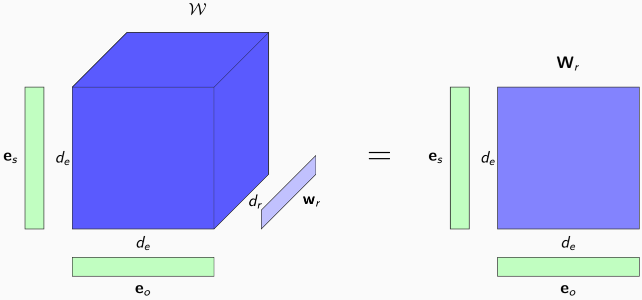


Figure 1: Visualization of the TuckER architecture.

$$\phi_{\text{TuckER}}(e_s, r, e_o) = ((\mathcal{W} \times_1 \mathbf{w}_r) \times_2 \mathbf{e}_s) \times_3 \mathbf{e}_o = \mathbf{e}_s^\top \mathbf{W}_r \mathbf{e}_o$$

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)

MuRE: Multi-relational Euclidean Graph Embeddings

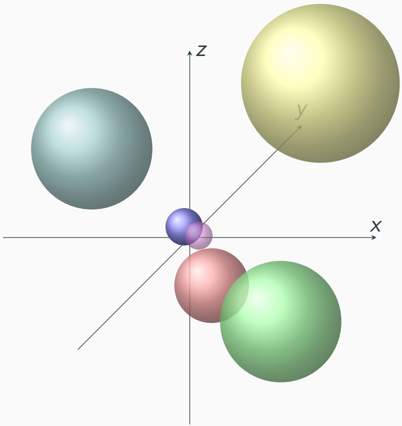


Figure 2: MuRE spheres of influence.

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

(Balažević et al., 2019b)

Recap

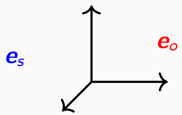
- KGs store facts: binary **relations** between **entities** (e_s, r, e_o).

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

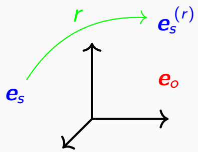
- KGs store facts: binary **relations** between **entities** (e_s, r, e_o).
- Enable computational reasoning over KGs,
  e.g. question answering and inferring new facts (**link prediction**)
- Requires representation, typically:
  - each entity by a vector **embedding** $\boldsymbol{e} \in \mathbb{R}^d$,
  - 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

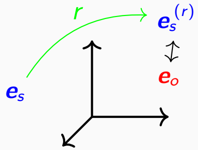
- KGs store facts: binary **relations** between **entities** (e_s, r, e_o).
- Enable computational reasoning over KGs,
  e.g. question answering and inferring new facts (**link prediction**)
- Requires representation, typically:
  - each entity by a vector **embedding** $\boldsymbol{e} \in \mathbb{R}^d$,
  - 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

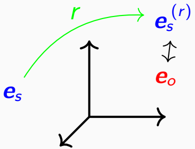
- KGs store facts: binary **relations** between **entities** (e_s, r, e_o).
- Enable computational reasoning over KGs,
  e.g. question answering and inferring new facts (**link prediction**)
- Requires representation, typically:
  - each entity by a vector **embedding** $\boldsymbol{e} \in \mathbb{R}^d$,
  - 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

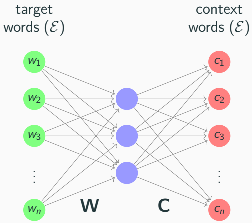
- KGs store facts: binary **relations** between **entities** (e_s, r, e_o).
- Enable computational reasoning over KGs,
  e.g. question answering and inferring new facts (**link prediction**)
- Requires representation, typically:
  - each entity by a vector **embedding** $\boldsymbol{e} \in \mathbb{R}^d$,
  - 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).

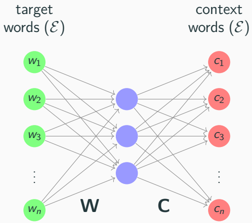
Simplify: consider Word Embeddings

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



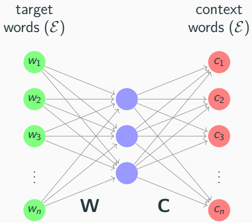
Simplify: consider Word Embeddings

- 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

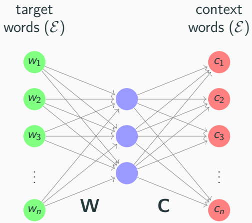


- Word embeddings, e.g.
  - **Word2Vec** (W2V, Mikolov et al., 2013)
  - **GloVe** (Pennington et al., 2014)

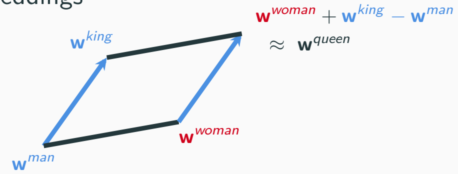
- Observation: **semantic** relations between words $\Longrightarrow$ **geometric** relationships between embeddings
  - **similar** words $\Rightarrow$ close embeddings

Simplify: consider Word Embeddings

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

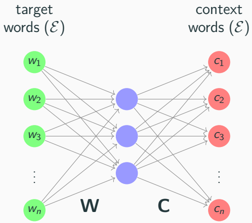


- Observation: **semantic** relations between words $\implies$ **geometric** relationships between embeddings
  - **similar** words $\Rightarrow$ close embeddings
  - **analogies** (often) $\Rightarrow$

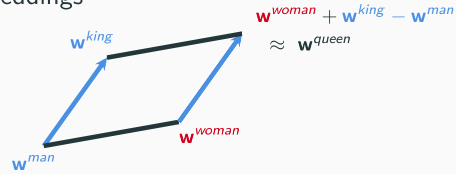


Simplify: consider Word Embeddings

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



- Observation: **semantic** relations between words $\Longrightarrow$ **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

$$-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(\mathbf{w}_i^\top \mathbf{c}_j) + \frac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-\mathbf{w}_i^\top \mathbf{c}_j))$$

Understanding word embeddings: the W2V Loss Function

$$-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(\mathbf{w}_i^\top \mathbf{c}_j) + \tfrac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-\mathbf{w}_i^\top \mathbf{c}_j))$$

$$\nabla_{\mathbf{w}_i} \ell_{W2V} \propto \sum_j \underbrace{\{p(w_i, c_j) + kp(w_i)p(c_j)\}}_{\mathbf{d}_j^{(i)}} \underbrace{\{\sigma(\mathbf{S}_{i,j}) - \sigma(\mathbf{w}_i^\top \mathbf{c}_j)\}}_{\mathbf{e}_j^{(i)}} \mathbf{c}_j = \mathbf{C}\,\mathrm{diag}(\mathbf{d}^{(i)})\mathbf{e}^{(i)}$$

Understanding word embeddings: the W2V Loss Function

$$-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(\mathbf{w}_i^\top \mathbf{c}_j) + \tfrac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-\mathbf{w}_i^\top \mathbf{c}_j))$$

$$\nabla_{\mathbf{w}_i} \ell_{W2V} \propto \sum_j \underbrace{\{p(w_i, c_j) + kp(w_i)p(c_j)\}}_{\mathbf{d}_j^{(i)}} \underbrace{\{\sigma(\mathbf{S}_{i,j}) - \sigma(\mathbf{w}_i^\top \mathbf{c}_j)\}}_{\mathbf{e}_j^{(i)}} \mathbf{c}_j = \mathbf{C}\,\mathrm{diag}(\mathbf{d}^{(i)})\mathbf{e}^{(i)}$$

- ℓ_{W2V} minimised when:

low-rank case: $\mathbf{w}_i^\top \mathbf{c}_j = \underbrace{\log \frac{p(c_j|w_i)}{p(c_j)}}_{\mathbf{PMI}(w_i,\, c_j)} - \log k \doteq \mathbf{S}_{i,j}$ (Levy and Goldberg, 2014)

Understanding word embeddings: the W2V Loss Function

$$-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(\mathbf{w}_i^\top \mathbf{c}_j) + \frac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-\mathbf{w}_i^\top \mathbf{c}_j))$$

$$\nabla_{\mathbf{w}_i} \ell_{W2V} \propto \sum_j \underbrace{\{p(w_i, c_j) + kp(w_i)p(c_j)\}}_{\mathbf{d}_j^{(i)}} \underbrace{\{\sigma(\mathbf{S}_{i,j}) - \sigma(\mathbf{w}_i^\top \mathbf{c}_j)\}}_{\mathbf{e}_j^{(i)}} \mathbf{c}_j = \mathbf{C}\,\mathrm{diag}(\mathbf{d}^{(i)})\mathbf{e}^{(i)}$$

- ℓ_{W2V} minimised when:

low-rank case: $\mathbf{w}_i^\top \mathbf{c}_j = \underbrace{\log \frac{p(c_j|w_i)}{p(c_j)}}_{\mathbf{PMI}(w_i, c_j)} - \log k \doteq \mathbf{S}_{i,j}$ (Levy and Goldberg, 2014)

general case: error vectors $\mathrm{diag}(\mathbf{d}^{(i)})\mathbf{e}^{(i)}$ orthogonal to rows of $\mathbf{C}$

Understanding word embeddings: the W2V Loss Function

$$-\ell_{W2V} = \sum_{i,j} \#(w_i, c_j) \log \sigma(\mathbf{w}_i^\top \mathbf{c}_j) + \frac{k\#(w_i)\#(c_j)}{D} \log(\sigma(-\mathbf{w}_i^\top \mathbf{c}_j))$$

$$\nabla_{\mathbf{w}_i} \ell_{W2V} \propto \sum_j \underbrace{\{p(w_i, c_j) + kp(w_i)p(c_j)\}}_{\mathbf{d}_j^{(i)}} \underbrace{\{\sigma(\mathbf{S}_{i,j}) - \sigma(\mathbf{w}_i^\top \mathbf{c}_j)\}}_{\mathbf{e}_j^{(i)}} \mathbf{c}_j = \mathbf{C}\,\mathrm{diag}(\mathbf{d}^{(i)})\mathbf{e}^{(i)}$$

- ℓ_{W2V} minimised when:

low-rank case: $\mathbf{w}_i^\top \mathbf{c}_j = \underbrace{\log \frac{p(c_j|w_i)}{p(c_j)}}_{\mathbf{PMI}(w_i, c_j)} - \log k \doteq \mathbf{S}_{i,j}$ (Levy and Goldberg, 2014)

general case: error vectors $\mathrm{diag}(\mathbf{d}^{(i)})\mathbf{e}^{(i)}$ orthogonal to rows of $\mathbf{C}$

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

(* drop k term as artefact of the W2V algorithm.)

PMI Vectors

$$\mathbf{p}^i \;=\; \left\{\log \frac{p(c_j|w_i)}{p(c_j)}\right\}_{c_j \in \mathcal{E}} \;=\; \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E})} \qquad (\mathcal{E} = \text{dictionary of all words})$$

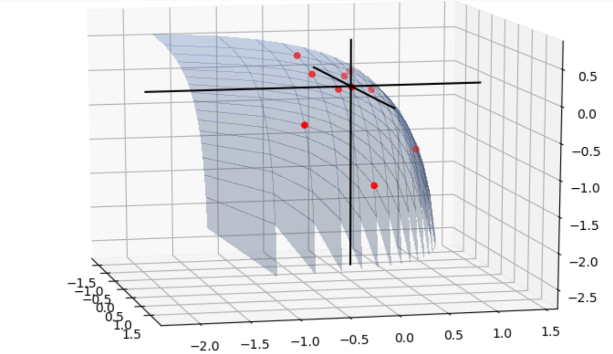


Figure 3: The PMI surface $\mathcal{S}$ with example PMI vectors of words (red dots)

PMI Vector Interactions = Semantics (Similarity)

Similarity: similar words, e.g. synonyms, induce similar distributions, $p(\mathcal{E}|w)$, over context words.

PMI Vector Interactions = Semantics (Similarity)

Similarity: similar words, e.g. synonyms, induce similar distributions, $p(\mathcal{E}|w)$, over context words.

Identified by **subtraction** of PMI vectors:

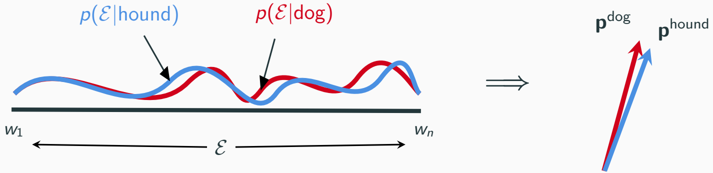
$$\mathbf{p}^i - \mathbf{p}^j \;=\; \log \tfrac{p(\mathcal{E}|w_i)}{p(\mathcal{E}|w_j)} \;=\; \boldsymbol{\rho}^{i,j}$$

PMI Vector Interactions = Semantics (Similarity)

Similarity: similar words, e.g. synonyms, induce similar distributions, $p(\mathcal{E}|w)$, over context words.

Identified by **subtraction** of PMI vectors:

$$\mathbf{p}^i - \mathbf{p}^j \;=\; \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E}|w_j)} \;=\; \boldsymbol{\rho}^{i,j}$$



PMI Vector Interactions = Semantics (Paraphrase)

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

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:

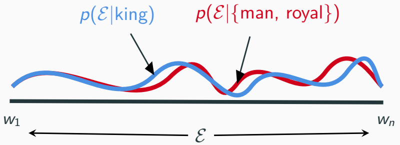
$$\begin{aligned}
\mathbf{p}^i + \mathbf{p}^j &= \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E})} + \log \frac{p(\mathcal{E}|w_j)}{p(\mathcal{E})} \\
&= \mathbf{p}^k + \underbrace{\underbrace{\log \frac{p(\mathcal{E}|w_i,w_j)}{p(\mathcal{E}|w_k)}}_{\rho^{\{i,j\},k}}}_{\text{paraphrase error}} - \underbrace{\underbrace{\log \frac{p(w_i,w_j|\mathcal{E})}{p(w_i|\mathcal{E})p(w_j|\mathcal{E})}}_{\boldsymbol{\sigma}^{i,j}} + \underbrace{\log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}}_{\tau^{i,j}}}_{\text{independence error}}
\end{aligned}$$

PMI Vector Interactions = Semantics (Paraphrase)

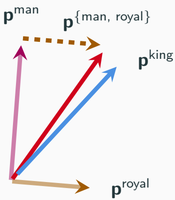
Paraphrases: word sets with similar aggregate semantic meaning, e.g. $\{$man, royal$\} \approx$ king.

Identified by **addition** of PMI vectors:

$$
\begin{aligned}
\mathbf{p}^i + \mathbf{p}^j &= \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E})} + \log \frac{p(\mathcal{E}|w_j)}{p(\mathcal{E})} \\
&= \mathbf{p}^k + \underbrace{\underbrace{\log \frac{p(\mathcal{E}|w_i, w_j)}{p(\mathcal{E}|w_k)}}_{\rho^{\{i,j\},k}}}_{\text{paraphrase error}} \underbrace{- \underbrace{\log \frac{p(w_i, w_j|\mathcal{E})}{p(w_i|\mathcal{E})p(w_j|\mathcal{E})}}_{\boldsymbol{\sigma}^{i,j}} + \underbrace{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}_{\tau^{i,j}}}_{\text{independence error}}
\end{aligned}
$$



$\Longrightarrow$



PMI Vector Interactions = Semantics (Analogy)

Analogies: word pairs that share a similar semantic difference, e.g. $\{$man, king$\}$ and $\{$woman, queen$\}$.

PMI Vector Interactions = Semantics (Analogy)

Analogies: word pairs that share a similar semantic difference, e.g. {man, king} and {woman, queen}.

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

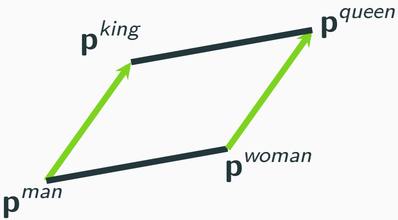
$$\mathbf{p}^{king} - \mathbf{p}^{man} \approx \mathbf{p}^{queen} - \mathbf{p}^{woman}$$

PMI Vector Interactions = Semantics (Analogy)

Analogies: word pairs that share a similar semantic difference, e.g. {man, king} and {woman, queen}.

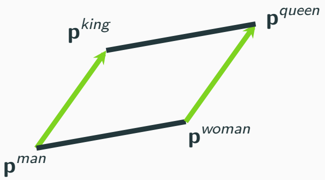
Identified by a **linear combination** of PMI vectors:

$$\mathbf{p}^{king} - \mathbf{p}^{man} \approx \mathbf{p}^{queen} - \mathbf{p}^{woman}$$



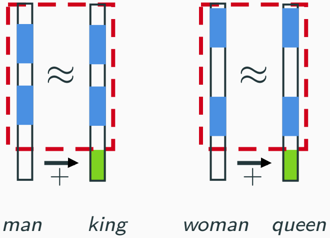
(Allen and Hospedales, 2019; Allen et al., 2019)

From Analogies to Relations



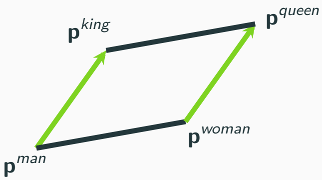
Analogy

⇔



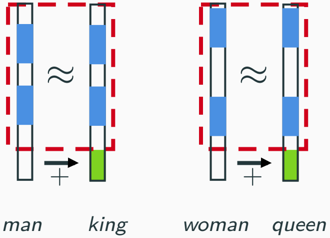
Relation

From Analogies to Relations



Analogy

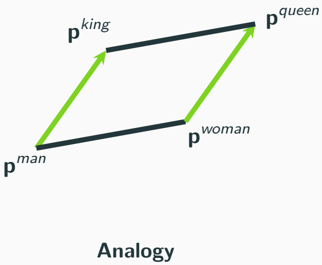
$\Leftrightarrow$



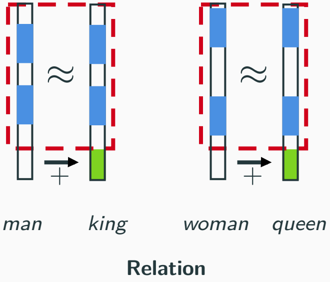
Relation

- Analogies contain common **binary word relations**, similar to KGs.

From Analogies to Relations

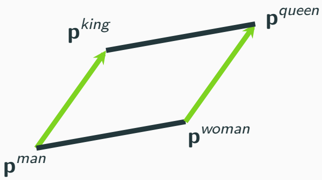


$\Leftrightarrow$



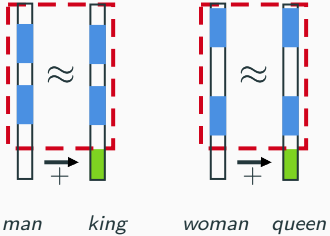
- Analogies contain common **binary word relations**, similar to KGs.
- For certain analogies ("specialisations"), the associated "vector offset" gives a **transformation that represents the relation**.

From Analogies to Relations



Analogy

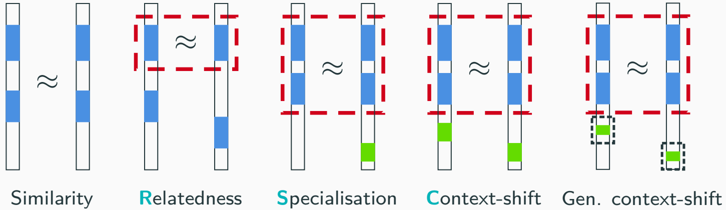
$\Leftrightarrow$



Relation

- Analogies contain 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.

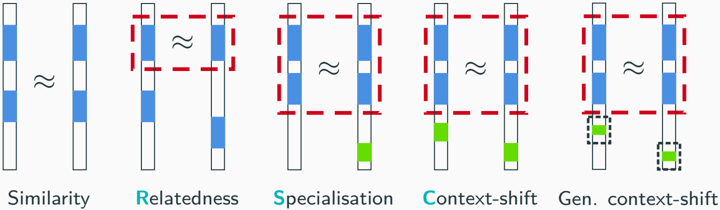
Categorising Relations: semantics → relation requirements



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



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

Categorisation of WN18RR relations.

Type	Relation	Examples *(subject entity, object entity)*
R	verb_group	*(trim_down_VB_1, cut_VB_35)*, *(hatch_VB_1, incubate_VB_2)*
	derivationally_related_form	***(lodge_VB_4, accommodation_NN_4)***, *(question_NN_1, inquire_VB_1)*
	also_see	*(clean_JJ_1, tidy_JJ_1)*, *(ram_VB_2, screw_VB_3)*
S	hypernym	*(land_reform_NN_1, reform_NN_1)*, *(prickle-weed_NN_1, herbaceous_plant_NN_1)*
	instance_hypernym	***(yellowstone_river_NN_1, river_NN_1)***, *(leipzig_NN_1, urban_center_NN_1)*
C	member_of_domain_usage	*(colloquialism_NN_1, figure_VB_5)*, *(plural_form_NN_1, authority_NN_2)*
	member_of_domain_region	***(rome_NN_1, gladiator_NN_1), (usa_NN_1, multiple_voting_NN_1)***
	member_meronym	*(south_NN_2, sunshine_state_NN_1)*, *(genus_carya_NN_1, pecan_tree_NN_1)*
	has_part	*(aircraft_NN_1, cabin_NN_3)*, *(morocco_NN_1, atlas_mountains_NN_1)*
	synset_domain_topic_of	*(quark_NN_1, physics_NN_1)*, *(harmonize_VB_3, music_NN_4)*

Categorical completeness: are all relations covered?

- 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)

Categorical completeness: are all relations covered?

- 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.

Categorical completeness: are all relations covered?

- 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: $\mathcal{S}$-relatedness requires both entity embeddings $\boldsymbol{e}_s, \boldsymbol{e}_o$ to share a common subspace component $\mathbb{V}_{\mathcal{S}}$

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

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

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

Summary

- **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.

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:

$$\underbrace{\textbf{additive \& multiplicative}}_{\text{MuRE* (Balažević et al., 2019a)}} \quad > \quad \underbrace{\textbf{multiplicative}}_{\substack{\text{TuckER (Balažević et al., 2019b)} \\ \text{DistMult (Yang et al., 2015)}}} \quad \textit{or} \quad \underbrace{\textbf{additive}}_{\text{TransE (Bordes et al., 2013)}}$$

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

Thanks!

Any questions?

References i

Carl Allen and Timothy Hospedales. Analogies Explained: Towards Understanding Word Embeddings. In *ICML*, 2019.

Carl Allen, Ivana Balažević, and Timothy Hospedales. What the Vec? Towards Probabilistically Grounded Embeddings. In *NeurIPS*, 2019.

Carl Allen, Ivana Balažević, and Timothy Hospedales. Interpreting Knowledge Graph Relation Representation from Word Embeddings. In *ICLR*, 2021.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Multi-relational Poincaré Graph Embeddings. In *NeurIPS*, 2019a.

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*, 2013.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NeurIPS*, 2014.

References ii

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, 2014.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*, 2015.