Knowledge Graph Embedding : An introduction

Thibaut SOULARD

Web Of Data,
Data-Science
Université Paris-Saclay

7 february 2025

Description of this course

In this course my goal is to offer a **non exhaustive** overview of the landscape of **Knowledge Graph embedding**. We will briefly see the math behind each and every of the following models but we will not take a dive into their optimisation and the challenges related to such formula.

Summary

What is an embedding ?

Definition

Embedding :

Given an object X and another object Y, it is said that X is embedded in Y iff $\mathcal{F} : X \to Y$ such that the map $\mathcal{F}$ is injective and structure-preserving.

What is an embedding ?

Definition

Embedding :
Given an object X and another object Y, it is said that X is embedded in Y iff $\mathcal{F} : X \to Y$ such that the map $\mathcal{F}$ is injective and structure-preserving.

- **Injective** : $\forall (x, x') \in X^2, (f(x) = f(x') \Rightarrow x = x')$
- **Structure-preserving**:
  - Domain dependant
  - Task dependant

An example : Word Embedding

Represent **words** in a way that captures their **meaning and relationships**, so that **similar words** have similar representations in a continuous space.

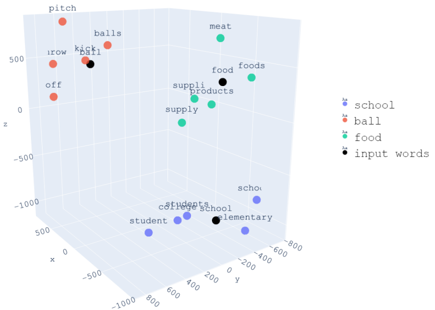


Figure: Word Embedding

An example : An example of representation with N dimensions

Represent **words** in a way that captures their **meaning and relationships**, so that **similar words** have similar representations in a continuous space.

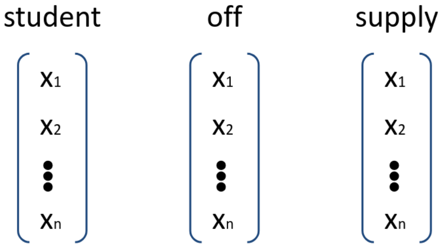


Figure: A representation of words with N dimensions

Another example : Graph Embedding

Represent **nodes** and **edges** in a way that captures **relationships** between the nodes in a continuous space.

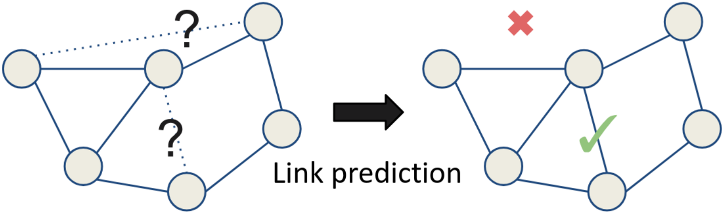


Figure: Link prediction

Another example : Graph Embedding

Represent **nodes** and **edges** in a way that captures **relationships** between the nodes in a continuous space.

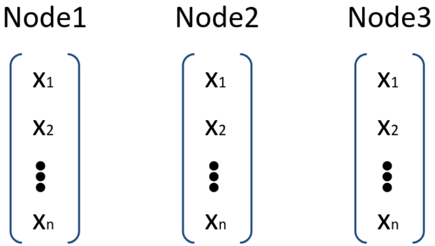


Figure: A representation of nodes with N dimensions

Summary

Knowledge Graph

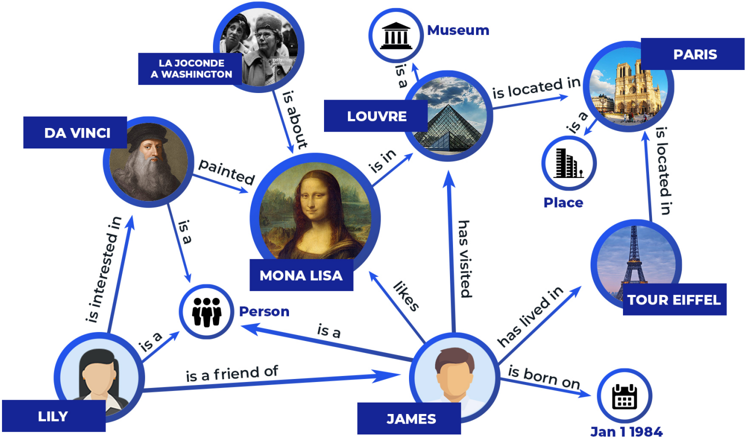


Figure: An example of a Knowledge Graph

Knowledge Graph

Definition

We consider a **knowledge graph** defined by a pair $(\mathcal{O}, \mathcal{G})$, where:

- $\mathcal{O} = (\mathcal{C}, \mathcal{P})$ is an **ontology** represented in OWL and composed of a set of **classes** $\mathcal{C}$ and **properties** $\mathcal{P}$ that can be either of type `owl:objectProperty` or `owl:dataTypeProperty`.
- $\mathcal{G}$ is a set of **RDF data graph**.

Definition

An **RDF data graph** $\mathcal{G}$ is a set of facts represented by triples of the form {(subject, predicate, object) | subject $\in \mathcal{I}$, property $\in \mathcal{P}$, object $\in \mathcal{I} \cup \mathcal{L}$}, where $\mathcal{I}$ is the set of **entities** designated IRIs, $\mathcal{P}$ is the set of **properties**, and $\mathcal{L}$ is the set of **literals**.

Knowledge Graph Embedding Tasks

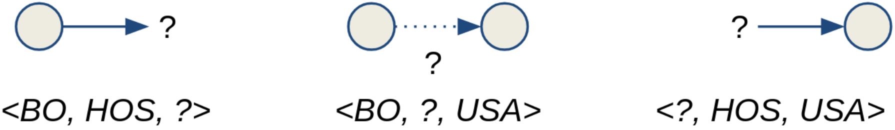
A knowledge graph can be used to perform :

- A completion of information based upon the data stored in it.
- An agglomeration of sources of information.
- A question answering tool.
- ...

Knowledge Graph Embedding Tasks

Knowledge Graph Completion (KGC)

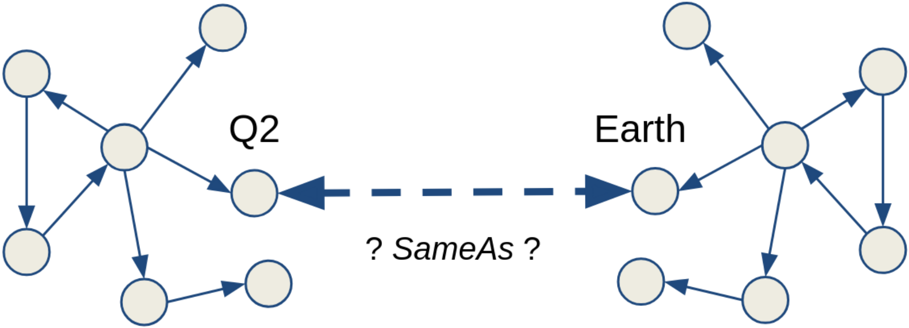
Given a **partial triple**, can the model find the **missing** component ?



Knowledge Graph Embedding Tasks

Entity Linking

Given two **Knowledge Graphs** (possibly **Heterogene**), can the model aligns the **entities** representing the **same real world entity** together ?

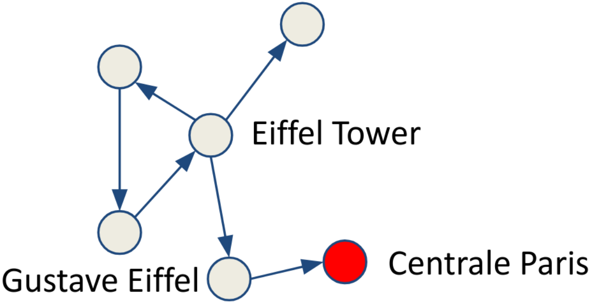


Knowledge Graph Embedding Tasks

Question answering

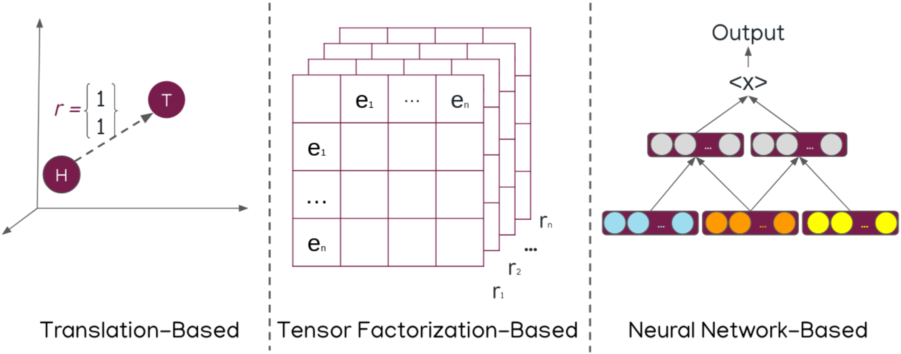
Given a **question** (given in a **naturel language** or **not**) can the model provide the **right answer** by relying on a KG only.

Q = Where did the architect of the Eiffel Tower study ?



Summary

Main approaches



Translations-Based Models - Original Idea

Word2Vec [Mikolov et al., 2013] :

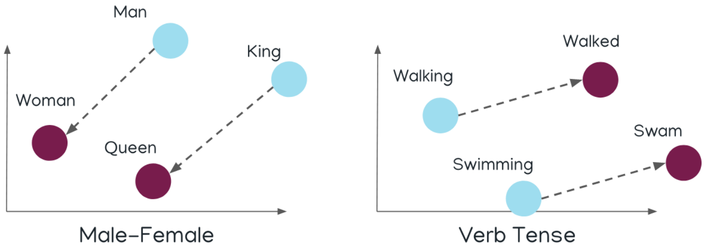


Figure: Word2Vec

$$\text{King} - \text{Man} + \text{Woman} \simeq \text{Queen}$$

$$\text{Walking} - \text{Swimming} + \text{Walked} \simeq \text{Swan}$$

One embedding per **type of relation** that we want to **represent**.

Translations-Based Models - Original Idea

Word2Vec [Mikolov et al., 2013] :

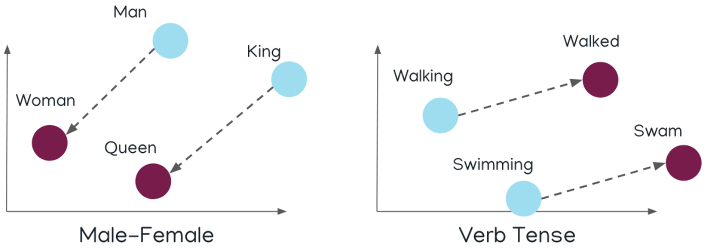


Figure: Word2Vec

One embedding per **type of relation** that we want to **represent**. However, our problem requires **multiple relations** and these relations needs to **interact with each others** thus this representation does not satisfy our problem and we need to come up with a new one.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

$$H + r \simeq T$$

Figure: An example of representation for TransE

Definition

Translating Embedding - TransE [Bordes et al., 2013]

A first try at the loss :

$$\mathcal{L} = \sum_{(h,r,t) \in S} d(h + r, t)$$

where S is the training set of triples.

If we only have this **Loss**, what is going to happen ?

Translations-Based Models - First Model : TransE

Definition

Translating Embedding - TransE [Bordes et al., 2013]
A first try at the loss :

$$\mathcal{L} = \sum_{(h,r,t)\in S} d(h+r,t)$$

where S is the training set of triples.

If we only have this **Loss**, what is going to happen ?

A **perfect Loss** $\mathcal{L}$ could be obtained through the use of **zero vector** for each embeddings.

$$\left\{ \begin{array}{r} h = (0,\cdots,0) \\ r = (0,\cdots,0) \\ t = (0,\cdots,0) \end{array} \right\}$$

Translations-Based Models - First Model : TransE

We introduce the **negative sampling** as a mean to avoid this issue.

Definition

Translating Embedding - TransE [Bordes et al., 2013]
A second try at the loss :

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [d(h+r,t) - d(h'+r,t')]_+$$

where S is the training set of triples and S' is the set of corrupted triples such that $[x]_+ = max(0,x)$ and
$S'_{(h,r,t)} = \{(h',r,t)|h' \in E, (h',r,t) \notin S\} \cup \{(h,l,t')|t' \in E,, (h,r,t') \notin S\}$.

$$< \text{Obama}, \text{HeadOfState}, \text{USA} >\in S$$
$$< \text{Macron}, \text{HeadOfState}, \text{USA} >\in S'$$

Translations-Based Models - First Model : TransE

We introduce the **negative sampling** as a mean to avoid this issue.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

A second try at the loss :

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [d(h+r,t) - d(h'+r,t')]_+$$

Is it enough ?

We introduce the negative sampling as a mean to avoid this issue.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

A second try at the loss :

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [d(h+r,t) - d(h'+r,t')]_+$$

Is it enough ?

Still not sadly. If we still have **zero vectors** as embeddings then the **Loss is still perfect**.

Translations-Based Models - First Model : TransE

Let's use the **Margin Based Ranking** to solve this issue.

Definition

Translating Embedding - TransE [Bordes et al., 2013]
The third try the charm :

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + d(h+r,t) - d(h'+r,t')]_+$$

where S is the training set of triples and S' is the set of corrupted triples such that $[x]_+ = max(0,x)$ and
$S'_{(h,r,t)} = \{(h',r,t)|h' \in E, (h',r,t) \notin S\} \cup \{(h,l,t')|t' \in E,, (h,r,t') \notin S\}$.

Is it enough ?

Let's use the **Margin Based Ranking** to solve this issue.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

The third try the charm :

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + d(h+r,t) - d(h'+r,t')]_+$$

Is it enough ?

Yes, but **actually no we still have an issue**.

Translations-Based Models - First Model : TransE

What if the model for a corruption that is **obviously** false, decide to put it **infinitly far** from the **translation** ?

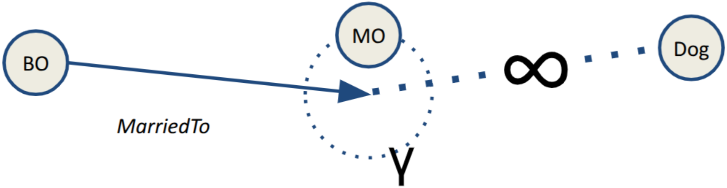


Figure: The final issue of the Loss of TransE

The loss will be again perfect and get to 0, however our embedding we do not get any assurance on the quality of the other embeddings.

To solve this isue our Margin Based Ranking receives an additional constraint.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + d(h+r,t) - d(h'+r,t')]_+$$

•$\forall e \in \mathcal{E}, ||e||_2 \leq 1$

In this constraint, the entity are softly constrained to be at a distance 1 of the orgin.

Translations-Based Models - First Model : TransE

Then we can use the learned embeddings for the final task of Graph **Completion** or **Validation**.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

$$f_r(h, t) = d(h + r, t)$$

The distance can be either defined by the **L1-norm** or the **L2-norm**.

Now that we have defined the scoring function how do we use this value ?

Translations-Based Models - First Model : TransE

Then we can use the learned embeddings for the final task of Graph **Completion** or **Validation**.

Definition

Translating Embedding - TransE [Bordes et al., 2013]

$$f_r(h, t) = d(h + r, t)$$

The distance can be either defined by the **L1-norm** or the **L2-norm**.

Now that we have defined the scoring function how do we use this value ?

- A global threshold
- A relation threshold
- A comparison between triples

Now that we have defined the scoring function how do we use this value to **Complete** or **Validate** a graph ?

A global threshold

$$f_r(h, t) \leq \theta \Rightarrow \text{Valid}$$

A relation threshold

$$f_r(h, t) \leq \theta_r \Rightarrow \text{Valid}$$

A comparison between triples

$$\text{Rank}(f_r(h, t)) \leq \text{Max}_{rank} \Rightarrow \text{Valid}$$

A global threshold

$$f_r(h, t) \leq \theta \Rightarrow \text{Valid}$$

Does not work because we do not force any constraint for the behavior between two relations. Hence for one relation the entities could have a small distance between them while the other might be greater.

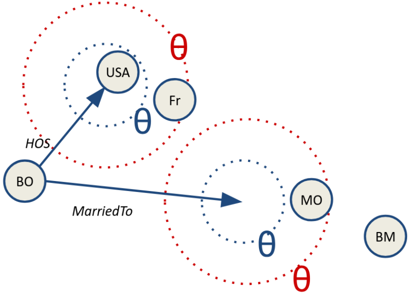


Figure: An example of two global thresholds that both does not fit the two relations

A relation threshold

$$f_r(h, t) \leq \theta_r \Rightarrow \text{Valid}$$

With this setting we are indeed able to work within relations and validate the right triple. However it does requires to learn a threshold per relation.

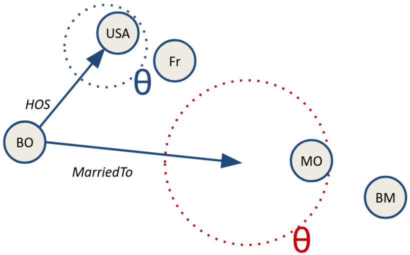


Figure: An example of two relation thresholds

A comparison between triples

$$\text{Rank}(f_r(h,t)) \leq \text{Max}_{rank} \Rightarrow \text{Valid}$$

This is classical method to assess the quality of an embedding model. It is named as the **Hit@X** metric but in a real world setting the **X** also has to be tuned to fit the relation.

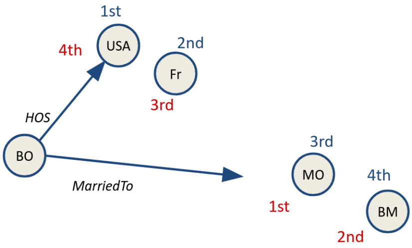


Figure: An example of two rankings

Now that we have defined the scoring function how do we use this value to **Complete** or **Validate** a graph ?

~~**A global threshold**~~ : Not allowed

$$f_r(h, t) \leq \theta \Rightarrow \text{Valid}$$

A relation threshold : Can be used

$$f_r(h, t) \leq \theta_r \Rightarrow \text{Valid}$$

A comparison between triples : Classical use

$$\text{Rank}(f_r(h, t)) \leq \text{Max}_{rank} \Rightarrow \text{Valid}$$

Summary of TransE

- Relation dependant translation
- Distance based scoring function
- Training through corruption
- Final decision with a comparison or a relation threshold

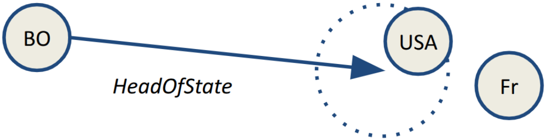


Figure: An example of training of TransE

- Is this model perfect ?
- Do we really need other models ?
- Do we really need other approaches ?
- Why there is still a lot of slides ?

Translations-Based Models - First Model : TransE

An issue Let's take the relation `citizenOf` and the tale `France`. How many head can we use to complete this triple ?
At least 66 millions today and many more if we take a look back.

Figure: N-1 relation

Thus while this approach can work greatly we have issues to represent some specific relations. [**1-n, n-n, n-1 relations**]

A N-M problem

To solve this issue, we would need to have the entity embedding to **be dependant of the relation**.

However we can not just re-use the definition of **Word2Vec** as we can not get an inter-relation embedding of the entities.

But what if we could **deviate** the embedding of an entity with respect to the relation it is used with ?

A N-M problem

Hence, I present to you the model **TransH**

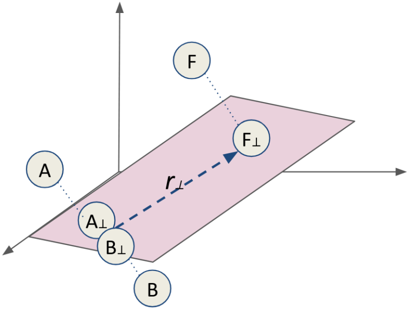


Figure: TransH

In this model, the authors proposed to first **project** entities to a **hyperplane** before applying any translation.

Translations-Based Models - TransH

Definition

Translating on Hyperplanes - TransH [Wang et al., 2014]

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + f_r(h,t) - f_r(h',t')]_+$$

$$f_r(h,t) = ||(h - w_r{}^\top h w_r) + r - (t - w_r{}^\top t w_r)||_2^2$$

by applying the 3 following constraints :

- $\forall e \in \mathcal{E}, ||e||_2 \leq 1,$ We limit the position of the entities
- $\forall r \in \mathcal{R}, \dfrac{|w_r^T d_r|}{||d_r||_2} \leq \epsilon,$ We force the orthogonality within ϵ
- $\forall r \in \mathcal{R}, ||w_r||_2 = 1,$ To obtain the previous def of f we need to force the unit-ness of the projection vector.

Definition

Translating on Hyperplanes - TransH [Wang et al., 2014]

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + f_r(h,t) - f_r(h',t')]_+$$
$$+ C\{\sum_{e\in\mathcal{E}} [||e||_2^2 - 1]_+ + \sum_{r\in\mathcal{R}} [\frac{(w_r^T d_r)^2}{||d_r||_2} - \epsilon]_+\}$$

All but the last constraint are represented in this loss but the latter is done by forcing w_r to follow the constraint when applying it using a normalisation application.

A digression - Negative Sampling

Classical negative sampling as in **TransE**: **UNIF**

First a set $S'_{(h,r,t)}$ is generated by the definition

$S'_{(h,r,t)} = \{(h', r, t) | h' \in E, (h', r, t) \notin S\} \cup \{(h, l, t') | t' \in E, , (h, r, t') \notin S\}$

Then **a set** of corrupted (negative) triples are sampled from this set to compute the loss.

A **new negative sampling**, described in **TransH**: **BERN**

We first compute two parameters dependant of the relation r of the triple (h, r, t).

- tph, the average number of tail entity per head entity.
- hpt, the average number of head entity per tail entity.

Then we define the probability that will corrupt the head by $\dfrac{tph}{tph + hpt}$.

Hence, when sampling for the **set** of currupted triples, we aim to follow the distribution of variability for the triple.

What did we do ?

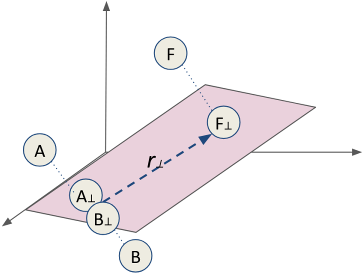


Figure: TransH

We started from a vectorial space of dimension *n* and reduced to a dimension *m* such that $m < n$. Thus, we solved the issue of **1-n** relations but we in a way reduced the expressivity in terms of dimension of the model.

Translations-Based Model - TransR

Instead of just projecting our entities to a smaller hyperplane in terms of dimension, we could transpose them into a **different vectorial space**. In this vectorial space we could variate the number of dimensions to represent the entities within the relation.

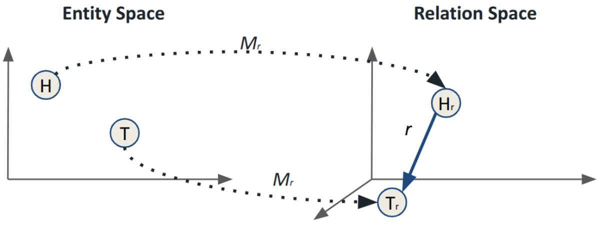


Figure: TransR

Translations-Based Model - TransR

Definition

TransR [Lin et al., 2015]

$$f_r(h,t) = ||h_r + r - t_r||_2^2$$

With $h_r = hM_r$ and $t_r = tM_r$

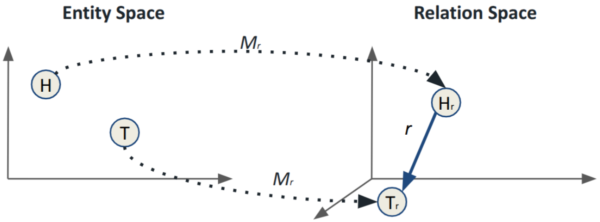


Figure: TransR

We have now described a model that can **diverge the representation** of an entity with respect to a **specific relation**. However are all entities connected by the same relation can be viewed as being part of the **same cluster** ?

In other words, can we consider them as really being similar ?

Translations-Based Model - CTransR : A digression

Definition

CTransR [Lin et al., 2015]

$$f_r(h,t) = ||h_{r,c} + r_c - t_{r,c}||_2^2 + \alpha||r_c - r||_2^2$$

With $h_{r,c} = hM_r$ and $t_{r,c} = tM_r$

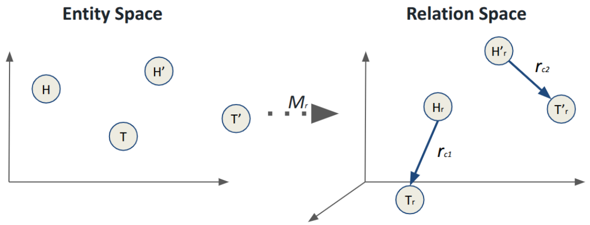


Figure: CTransR

From the previous approach we have been able to describe that not all entities represented by a relation *r* can be viewed as being part of the same cluster.

Hence, why do we represent the translation of the *Head* and the *Tale* as being entities of the same cluster ?

In TransD, we will see a more expressive model.

Definition

TransD - A position tailored model [Ji et al., 2015]

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma - f_r(h,t) + f_r(h',t'))]_+$$

$$f_r(h,t) = -||h_\perp + r_c - t_\perp||_2^2 + \alpha||r_c - r||_2^2$$

$$\begin{aligned} M_{rh} &= r_p h_p^t + I^{mxn} \quad & h_\perp &= M_{rh} h \\ M_{rt} &= r_p t_p^t + I^{mxn} \quad & t_\perp &= M_{rt} t \end{aligned}$$

Definition

TransD - A position tailored model [Ji et al., 2015]

$$f_r(h,t) = ||h_\perp + r_c - t_\perp||_2^2 + \alpha||r_c - r||_2^2$$

$$\begin{aligned} M_{rh} &= r_p h_p^t + I^{mxn} & h_\perp &= M_{rh} h \\ M_{rt} &= r_p t_p^t + I^{mxn} & t_\perp &= M_{rt} t \end{aligned}$$

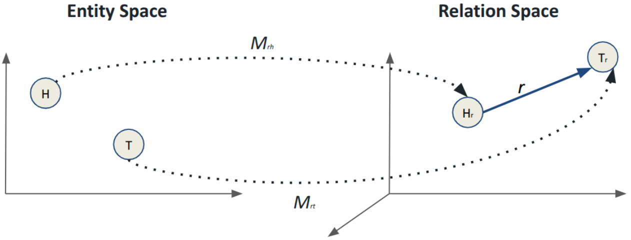


Figure: TransD

Translations-Based Model - TransA

So now we have the previous model that is able to express the transformation of an entity from an entity space to a **Relation Space** with a translation matrix that is dependant to its position in the triple.
While we have explored the **transformation of the representation**, do we have other parameters to play with ?

Yes, let's play with the **distance function**.

Definition

TransA [Xiao et al., 2015]

$$f_r(h,t) = (|h+r-t|)^\top W_r(|h+r-t|)$$

Does not look like much but what is hidden in W_r ?

Translations-Based Model - TransA

Definition

TransA [Xiao et al., 2015]

$$f_r(h,t) = (|h + r - t|)^\top W_r([h + r - t|)$$

Does not look like much but what is hidden in W_r ? The Mahalanobis distance !

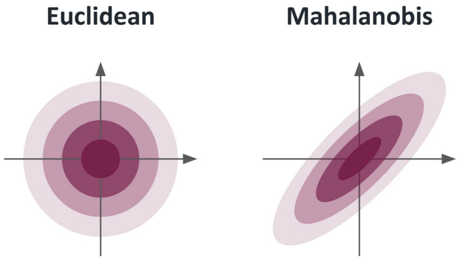


Figure: Euclidean VS Mahalanobis distance

Translations-Based Model - TransA

But how does one compute the Weights for the different dimensions ?

Definition

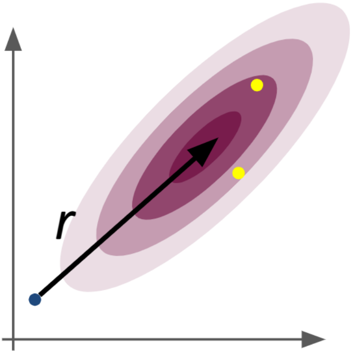
TransA [Xiao et al., 2015]

$$f_r(h,t) = (|h+r-t|)^\top W_r(|h+r-t|)$$

$$W_r = -\sum_{(h,r,t)\in S} (|h+r-t||h+r-t|^\top) + \sum_{(h',r,t')\in S'} (|h'+r-t'||h'+r-t'|^\top)$$

$$W_{r_{ij}} = [W_{r_{ij}}]_+$$

Let's try with an example :



Which of the two *Tale* (red dots) should be linked with the *Head* through the relation *r* ?

What about the ontology

Since the begining of this section we have not discussed the information displayed in the **ontology** of our KG.

With the rise of **Neuro-Symbolic** approaches it is important to introduce this information in our model but how do we model it ?

What about the ontology

Since the begining of this section we have not discussed the information displayed in the **ontology** of our KG.

With the rise of **Neuro-Symbolic** approaches it is important to introduce this information in our model but how do we model it ?

In the work of [d'Amato et al., 2021] they propose to re-use the scoring functions of the previous model while adding several soft-constraint into the loss $\mathcal{L}$ to infuse the knowledge coming from the ontology.

Translations-Based Model - TransOWL

Definition

TransOWL [d'Amato et al., 2021]

$$
\begin{aligned}
\mathcal{L} = & \sum_{\substack{<h,r,t>\in\Delta \\ <h',r,t'>\in\Delta'}} [\gamma + f_r(h,t) - f_r(h',t')]_+ + \sum_{\substack{<h,q,t>\in\Delta_{\text{inverseOf}} \\ <h',q,t'>\in\Delta'_{\text{inverseOf}}}} [\gamma + f_q(h,t) - f_q(h',t')]_+ \\
& + \sum_{\substack{<h,s,t>\in\Delta_{\text{equivProp}} \\ <h',s,t'>\in\Delta'_{\text{equivProp}}}} [\gamma + f_s(h,t) - f_s(h',t')]_+ \\
& + \sum_{\substack{<h,typeOf,p>\in\Delta\cup\Delta_{\text{equivClass}} \\ <h',typeOf,t'>\in\Delta\cup\Delta'_{\text{equivClass}}}} [\gamma + f_{\text{typeOf}}(h,t) - f_{\text{typeOf}}(h',t')]_+ \\
& + \sum_{\substack{<h,subClassOf,t>\in\Delta_{\text{subClass}} \\ <h',subClassOf,t'>\in\Delta'_{\text{subClass}}}} [\gamma - \beta + f(h,p) - f(h',p')]_+
\end{aligned}
$$

Translations-Based Model - TransROWL

Definition

TransROWL [d'Amato et al., 2021]

$$
\begin{aligned}
\mathcal{L} = & \sum_{\substack{<h,r,t>\in\Delta \\ <h',r,t'>\in\Delta'}} [\gamma + f_r(h,t) - f_r(h',t')]_+ + \lambda_1 \sum_{\substack{<h,q,t>\in\Delta_{\text{inverseOf}} \\ <h',q,t'>\in\Delta'_{\text{inverseOf}}}} [\gamma + f_q(h,t) - f_q(h',t')]_+ \\
+ & \quad \lambda_2 \sum_{\substack{<h,s,t>\in\Delta_{\text{equivProp}} \\ <h',s,t'>\in\Delta'_{\text{equivProp}}}} [\gamma + f_s(h,t) - f_s(h',t')]_+ \\
+ & \quad \lambda_3 \sum_{\substack{<h,typeOf,p>\in\Delta\cup\Delta_{\text{equivClass}} \\ <h',typeOf,t'>\in\Delta\cup\Delta'_{\text{equivClass}}}} [\gamma + f_{\text{typeOf}}(h,t) - f_{\text{typeOf}}(h',t')]_+ \\
+ & \quad \lambda_4 \sum_{\substack{<h,subClassOf,t>\in\Delta_{\text{subClass}} \\ <h',subClassOf,t'>\in\Delta'_{\text{subClass}}}} [\gamma - \beta + f(h,p) - f(h',p')]_+
\end{aligned}
$$

Translations-Based Model - Sum up

In this first section, we have seen 6 (8) models that all share the same idea of translation but a different levels of expressivity.

Model	Scoring function	Memory Complexity
TransE	$\|\|h + r - t\|\|_2$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
TransH	$\|\|(h - w_r^\top h w_r) + d_r - (t - w_r^\top t w_r)\|\|_2^2$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
TransR	$\|\|M_r h + r - M_r t\|\|_2^2$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r dk)$
CTransR	$\|\|h_{r,c} + r_c - t_{r,c}\|\|_2^2 + \alpha\|\|r_c - r\|\|_2^2$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r dkc)$
TransD	$\|\|(r_p h_p^\top + I)h + r - (r_p t_p^\top + I)t)\|\|_2^2$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)$
TransA	$(h + r - t)^\top W_r (h + r - t)$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k^2)(d = k)$

But this is not a full view of the field and other model exist in this type of approach such as **TransG**[Ou et al., 2016] or **KG2E**[He et al., 2015] that relies on a **probabilistic approach**.

Summary

Tensor factorization - An idea

A representation of a classical graph :

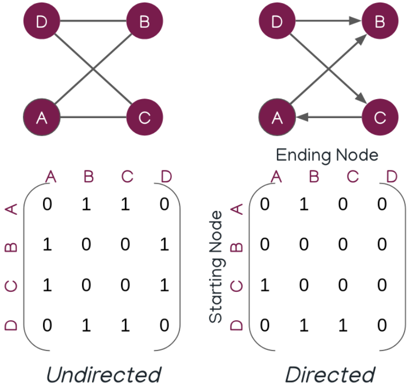


Figure: adjacency Matrix

With both representations we can not describe multiple relations.

Tensor factorization - An adaptation

Let's first define $\mathcal{X}$ as the three-way tensor where $\mathcal{X}_{ijk} = 1$ only if the triple `i-th entity, k-th relation, j-th entity` exists.

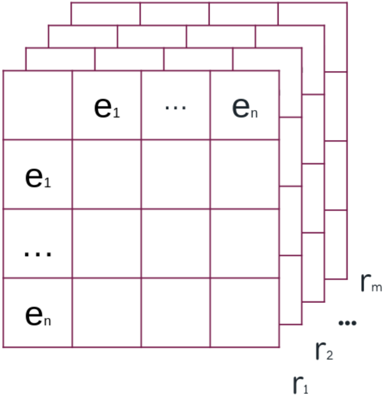
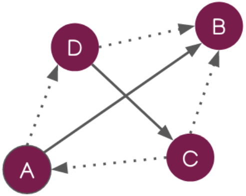


Figure: Tensor representation

Tensor factorization - An adaptation

An example of $\mathcal{X}$, with 4 *entities* and 2 *relations* :



$$
\begin{array}{c}
\begin{array}{cc}
 & \begin{array}{cccc} A & B & C & D \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}
\end{array} \\
R1
\end{array}
\quad
\begin{array}{c}
\begin{array}{cc}
 & \begin{array}{cccc} A & B & C & D \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}
\end{array} \\
R2
\end{array}
$$

Tensor factorization - An adaptation

Our goal :

For every following models, we want to design a scoring function such that for a triple $<\texttt{h,r,t}>$, we have $f_r(h,t) \simeq \mathcal{X}_{hrt}$.

In other word, for a triple $<\texttt{h,r,t}>$ the range of meaningful values that f can yield is between $[0,1]$. Thus we remove the distance dimension of all the following approaches.

Tensor factorization - Rescal

Definition

Rescal[Nickel et al., 2011]

$$f_r(h, t) = h^\top R_k t$$

$$\mathcal{L} = f(A, R_k) + g(A, R_k)$$

$$f(A, R_k) = \frac{1}{2} \sum_k ||\mathcal{X}_k - A R_k A^T||_F^2$$

$$g(A, R_k) = \frac{1}{2} \lambda (||A||_F^2 + \sum_k ||R_k||_F^2)$$

Frobenius Norm or the euclidean norm: $||M||_f = \sqrt{Tr(MM^T)}$

Tensor factorization - DistMult

Definition

DistMult [Yang et al., 2014]

$$f_r(h, t) = h^\top diag(r) t$$

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [1 - f_r(h, t) + f_r(h', t')]_+$$

Fewer parameters to optimize, but a drawback appears $h^\top rt = t^\top rh$. Thus through this approach we can only model **symmetrical** relations.

Tensor factorization - DistMult

Definition

DistMult [Yang et al., 2014]

$$f_r(h,t) = h^\top diag(r) t$$

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [1 - f_r(h,t) + f_r(h',t')]_+$$

Fewer parameters to optimize, but a main drawback appears $h^\top rt = t^\top rh$.

Let's keep the same type of approach and number of parameters while being able to model asymmetrical relations.

Tensor factorization - Hole

Definition

Circular correlation

$$[h \star t]_k = \sum_{i=0}^{d-1} h_i t_{(k+i) \bmod d}$$

$$h \star t = \begin{bmatrix} [h \star t]_0 \\ [h \star t]_{\dots} \\ [h \star t]_n \end{bmatrix}$$

Non commutative : $a \star b \neq b \star a$

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \star \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} (h_1 * t_1) + (h_2 * t_2) + (h_3 * t_3) \\ (h_1 * t_2) + (h_2 * t_3) + (h_3 * t_1) \\ (h_1 * t_3) + (h_2 * t_1) + (h_3 * t_2) \end{bmatrix}$$

Tensor factorization - Hole

Definition

Hole [Nickel et al., 2016]

$$f_r(h, t) = r^\top (h \star t)$$

We can now represent that

`<Paris, CapitalOf, France>` is True

while

`<France, CapitalOf, Paris>` is False.

Complex - Improve Rescal & DistMult

How can we do the same for the first operator ?

Can we do the same for the first operator ?

Let's introduce the Complex space

Definition

Complex [Trouillon et al., 2016]

$$\mathcal{X}_k \approx EW_k\overline{E}^T, \text{ for } k = 1, ..., m$$

with W_k a diagonal matrix.

$$f_r(h, t) = \mathrm{Re}(h^\top diag(r)\bar{t})$$

Complex - Improve Rescal & DistMult

Definition

Complex [Trouillon et al., 2016]

$$f_r(h,t) = \mathrm{Re}(h^\top diag(r)\bar{t})$$

$$Re\left(\begin{bmatrix} a_{h1}+b_{h1}i \\ a_{h2}+b_{h2}i \\ a_{h3}+b_{h3}i \end{bmatrix}^\top \begin{bmatrix} a_{r1}+b_{r1}i & 0 & 0 \\ 0 & a_{r2}+b_{r2}i & 0 \\ 0 & 0 & a_{r3}+b_{r3}i \end{bmatrix} \begin{bmatrix} a_{t1}-b_{t1}i \\ a_{t2}-b_{t2}i \\ a_{t3}-b_{t3}i \end{bmatrix}\right)$$

$$\begin{aligned} f_r(h,t) = \quad & \langle Re(w_r), Re(h), Re(t)\rangle \\ & + \langle Re(w_r), Im(h), Im(t)\rangle \\ & + \langle Im(w_r), Re(h), Im(t)\rangle \\ & + \langle Im(w_r), Re(h), Re(t)\rangle \end{aligned}$$

Tensor Factorization - Sum up

In this second section, we have seen 4 models.

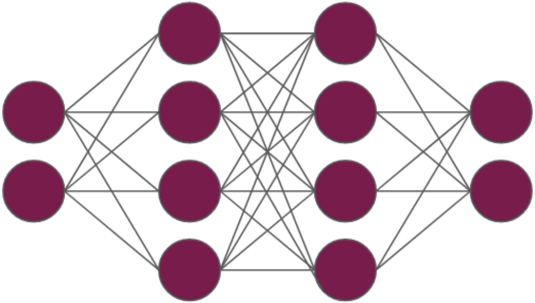
Model	Scoring function	Memory Complexity
RESCAL	$h^\top M_r t$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k^2)(d = k)$
DistMult	$h^\top diag(r) t$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
HolE	$r^\top (h \star t)$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
ComplEx	$Re(h^\top diag(r) \bar{t})$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$

We have seen that depending on the space that we considere and the size of the parameters declared the models can be less or more expressive and obtain different results.

Summary

Neural Network-Based Models - Main Idea

Let's now use Neural Network approaches to let the model more freedom in the representation of the information.



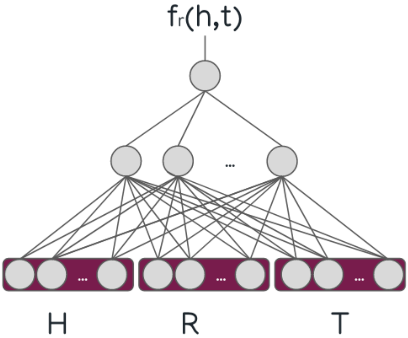
Neural Network

MLP

Definition

MLP [Dong et al., 2014]

$$
\begin{aligned}
f_r(h,t) &= \sigma(m^\top tanH(M_1 h + M_2 r + M_3 t)) \\
&= \sigma(\beta^\top tanH(M[h,r,t]))
\end{aligned}
$$



NAM

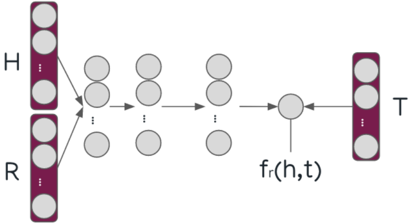
Definition

NAM [Liu et al., 2016]

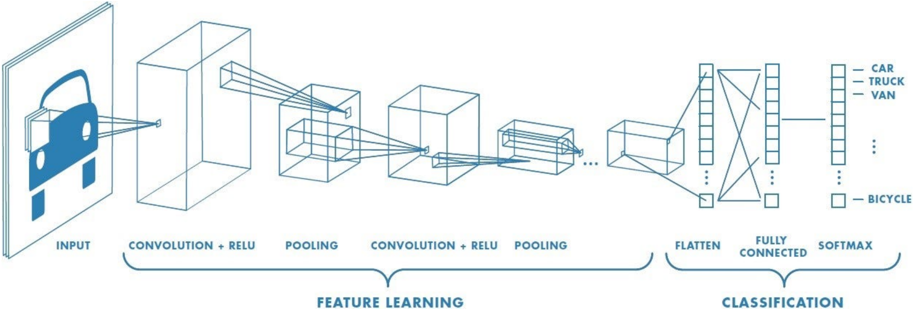
$$z^0 = [h; r]$$

$$\begin{cases} a^l = & M^l z^{l-1} + b^l \\ z^l = & ReLU(a^l) \end{cases}$$

$$f_r(h, t) = \sigma(z^L t)$$



Convulation Approach



Input Kernel Output

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 8 & 4 \\ 9 & 19 & 25 & 10 \\ 21 & 37 & 43 & 16 \\ 6 & 7 & 8 & 0 \end{bmatrix}$$

ConvKB

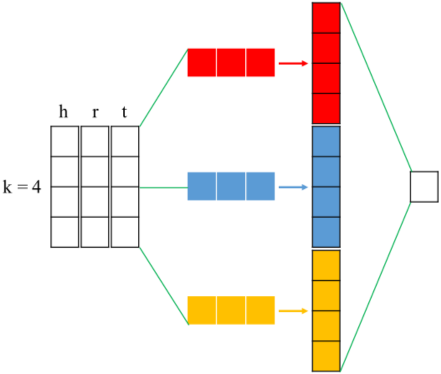
Definition

ConvKB [Nguyen et al., 2017]
Input : $A = [h, r, t]$ a matrix of dimension $3 * k$

ConvKB

Definition

ConvKB [Nguyen et al., 2017]
Input : $A = [h, r, t]$ a matrix of dimension $(3 * k)$
Kernel : τ kernels of dimension $(1 * 3)$



ConvKB

Definition

ConvKB [Nguyen et al., 2017]
Input : $A = [h, r, t]$ a matrix of dimension $(3 * k)$
Kernel ω : τ kernels of dimension $(1 * 3)$
Application of the kernels to obtain τ feature maps of dimension $(1 * k)$:

$$v = \begin{bmatrix} v_0 & = g(\omega . A_{0,:} + b) \\ & \cdots \\ v_k & = g(\omega . A_{k,:} + b) \end{bmatrix}$$

Where g is the activation function *ReLU* and b the bias parameter.

ConvKB

Definition

ConvKB [Nguyen et al., 2017]
Input : $A = [h, r, t]$ a matrix $\mathbb{R}^{(3\times k)}$
Kernel ω : τ kernels $\mathbb{R}^{(1\times 3)}$
Application of the kernels to obtain τ feature maps $\mathbb{R}^{(1\times k)}$:

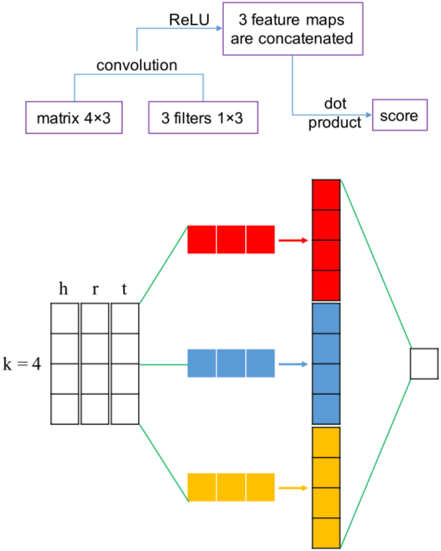
$$v = \begin{bmatrix} v_0 & = g(\omega.A_{0,:} + b) \\ & \cdots \\ v_k & = g(\omega.A_{k,:} + b) \end{bmatrix}$$

Where g is the activation function *ReLU* and b the bias parameter.
Final scoring function :

$$f_r(h, t) = concat(g([h, r, t] * \Omega)).w$$

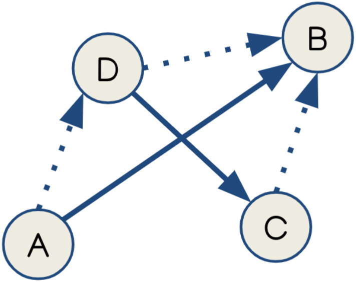
Where Ω is the set of kernels, *concat* the function that will transform τ feature maps v to a vector $\mathbb{R}^{(1\times(\tau * k))}$ and w a weight vector $\mathbb{R}^{(1\times(\tau * k))}$.

ConvKB - Schema



A message passing model : RGCN

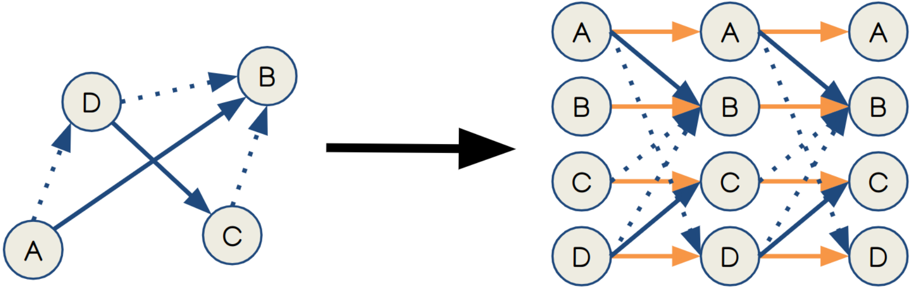
In the previous models, we have seen that we could take advantage of multiple layers however these methods do not take into consideration the neighborhood or the structure of the graph.



So now we will see a model that fully take advantage of the structure and re-use previous models.

A message passing model : RGCN

We will transform our Knowledge Graph into a layered stack of relations (**Blue full and dotted**) and we will add self loops (**Orange**).



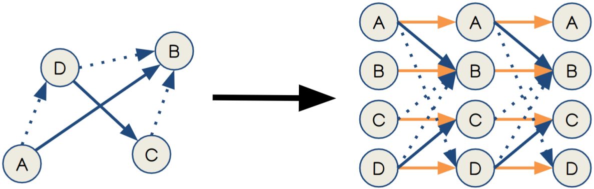
A message passing model : RGCN

Definition

RGCN [Schlichtkrull et al., 2018]

$$h_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}\right)$$

We have now represented our entities within their structure but we still lack our scoring function for the link prediction task.



A message passing model : RGCN

Definition

RGCN [Schlichtkrull et al., 2018]

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

$$f(s,r,o) = e_s^T R_r e_o$$

A message passing model : RGCN

Definition

RGCN [Schlichtkrull et al., 2018]

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

$$f(s,r,o) = e_s^T R_r e_o$$

DistMult

$$f_r(h,t) = h^\top diag(r)t$$

Which you might recognize as being ...

Neural Network - Sum up

In this section, we have seen 3 examples of how we can transform a neural network to fit our problem.

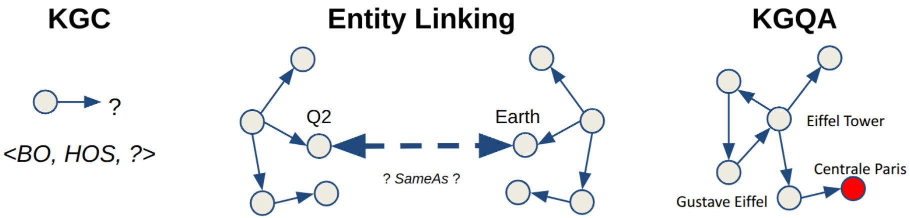
Model	Scoring function	Memory Complexity
MLP	$m^\top tanh(M_1 h + M_2 r + M3_t)$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
NAM	$\sigma(z^L t); z^l = ReLU(M^l z^{l-1} + b^l)$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
ConvKB	$Concat(g(A * \Omega)).w$	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$
RGCN	DistMult(MessagePassing)	$\mathcal{O}(\mathcal{N}_e d + \mathcal{N}_r k)(d = k)$

As you have seen, we are free to adapt any other Deep Learning approaches to our problem.

Summary

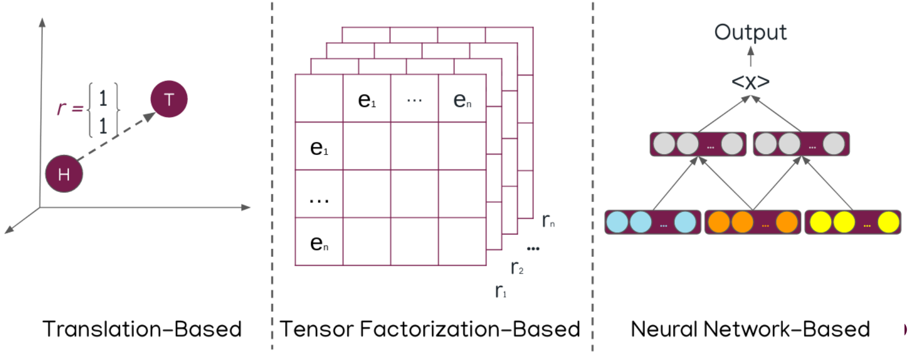
Conclusion

In this course, we have seen the different tasks that are common in the Knowledge Graph Embedding community



Conclusion

To solve these tasks, we were able to see some of the different family of models. However, if you are interested in this field do not limit yourself to them and explore the other type of models.



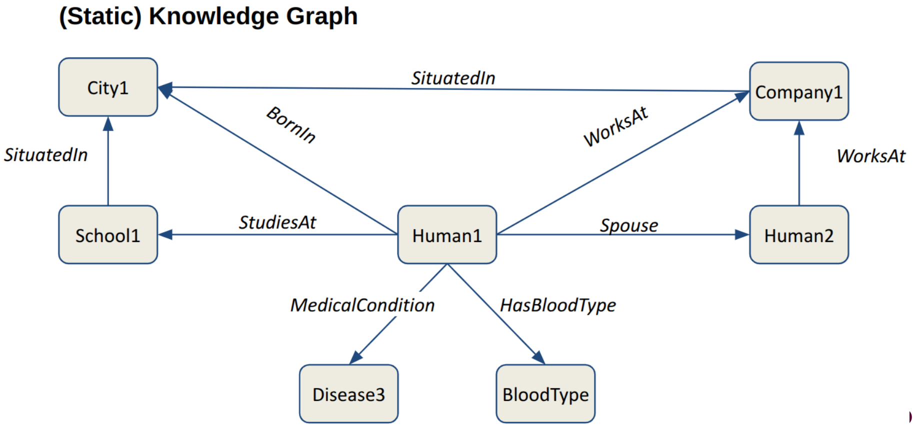
TP

Next week we will do a TP on **AmpliGraph** to learn a Knowledge Graph embedding package. Our goals will be to :

- Load a Knowledge Graph.
- Select a model and its hyper-parameters.
- Test such model.
- Augment the information available in a Knowledge Graph.

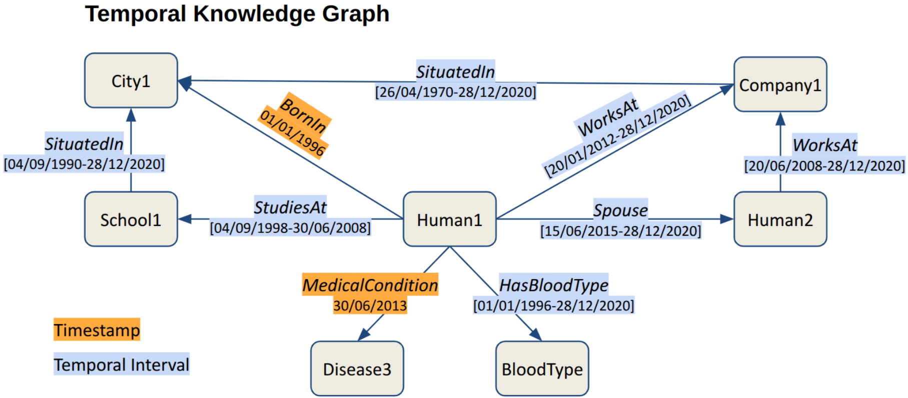
Is it enough ?

Until now we have only see **Static Knowledge Graph** or in the other words the information is always true regardless of a temporal component.



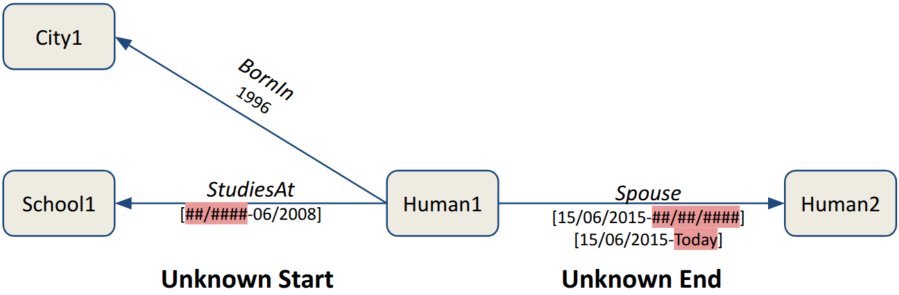
Is it enough ?

But to accuratly represent our world we need to become **Temporal** :



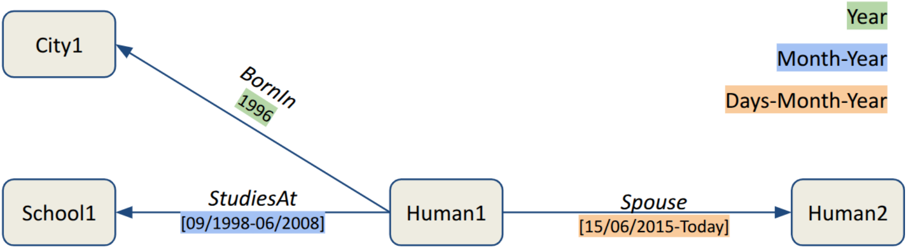
Is it enough ?

However we can represent time in different ways :



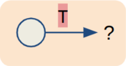
Is it enough ?

And with different precisions :

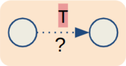


Is it enough ?

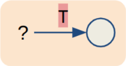
But our goal remains the same with the only added difficulty of a temporal context :



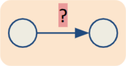
<*BO, HOS, ?, ['07,'11]*>



<*BO, ?, USA, ['07,'11]*>



<*?, HOS, USA, ['07,'11]*>



<*BO, HOS, USA, ?*>

References I

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013).
Translating embeddings for modeling multi-relational data.
In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Cai, L. and Wang, W. Y. (2017).
Kbgan: Adversarial learning for knowledge graph embeddings.
arXiv preprint arXiv:1711.04071.

Dai, Y., Wang, S., Xiong, N. N., and Guo, W. (2020).
A survey on knowledge graph embedding: Approaches, applications and benchmarks.
Electronics, 9(5):750.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014).
Knowledge vault: A web-scale approach to probabilistic knowledge fusion.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610.

d'Amato, C., Quatraro, N. F., and Fanizzi, N. (2021).
Injecting background knowledge into embedding models for predictive tasks on knowledge graphs.
In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*, pages 441–457. Springer.

Glorot, X., Bordes, A., Weston, J., and Bengio, Y. (2013).
A semantic matching energy function for learning with multi-relational data.

He, S., Liu, K., Ji, G., and Zhao, J. (2015).
Learning to represent knowledge graphs with gaussian embedding.
In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 623–632.

References II

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015).
Knowledge graph embedding via dynamic mapping matrix.
In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 687–696.

Kazemi, S. M. and Poole, D. (2018).
Simple embedding for link prediction in knowledge graphs.
Advances in neural information processing systems, 31.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015).
Learning entity and relation embeddings for knowledge graph completion.
In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Liu, Q., Jiang, H., Evdokimov, A., Ling, Z.-H., Zhu, X., Wei, S., and Hu, Y. (2016).
Probabilistic reasoning via deep learning: Neural association models.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013).
Efficient estimation of word representations in vector space.

Nguyen, D. Q., Nguyen, T. D., Nguyen, D. Q., and Phung, D. (2017).
A novel embedding model for knowledge base completion based on convolutional neural network.
arXiv preprint arXiv:1712.02121.

Nickel, M., Rosasco, L., and Poggio, T. (2016).
Holographic embeddings of knowledge graphs.
In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Nickel, M., Tresp, V., Kriegel, H.-P., et al. (2011).
A three-way model for collective learning on multi-relational data.
In *Icml*, volume 11, pages 3104482–3104584.

Ou, M., Cui, P., Pei, J., Zhang, Z., and Zhu, W. (2016).
Asymmetric transitivity preserving graph embedding.
In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018).
Modeling relational data with graph convolutional networks.
In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019).
Rotate: Knowledge graph embedding by relational rotation in complex space.
arXiv preprint arXiv:1902.10197.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016).
Complex embeddings for simple link prediction.
In *International conference on machine learning*, pages 2071–2080. PMLR.

Wang, J., Wang, B., Qiu, M., Pan, S., Xiong, B., Liu, H., Luo, L., Liu, T., Hu, Y., Yin, B., et al. (2023).
A survey on temporal knowledge graph completion: Taxonomy, progress, and prospects.
arXiv preprint arXiv:2308.02457.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014).
Knowledge graph embedding by translating on hyperplanes.
In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

References IV

Xiao, H., Huang, M., Hao, Y., and Zhu, X. (2015).
Transa: An adaptive approach for knowledge graph embedding.
arXiv preprint arXiv:1509.05490.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014).
Embedding entities and relations for learning and inference in knowledge bases.
arXiv preprint arXiv:1412.6575.

Zhang, S., Tay, Y., Yao, L., and Liu, Q. (2019).
Quaternion knowledge graph embeddings.