# TransG : A Generative Mixture Model for Knowledge Graph Embedding

**Han Xiao**[1], **Minlie Huang**[1], **Hao Yu**[1], **Xiaoyan Zhu**[1]
[1]Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China



**Abstract**

Recently, knowledge graph embedding, which projects symbolic entities and relations into continuous vector space, has become a new, hot topic in artificial intelligence. This paper addresses a new issue of **multiple relation semantics** that a relation may have multiple meanings revealed by the entity pairs associated with the corresponding triples, and proposes a novel Gaussian mixture model for embedding, **TransG**. The new model can discover latent semantics for a relation and leverage a mixture of relation component vectors for embedding a fact triple. To the best of our knowledge, this is the first generative model for knowledge graph embedding, which is able to deal with multiple relation semantics. Extensive experiments show that the proposed model achieves substantial improvements against the state-of-the-art baselines.

## Introduction

Abstract or real-world knowledge is always a major topic in Artificial Intelligence. Knowledge bases such as Wordnet (Miller 1995) and Freebase (Bollacker et al. 2008) have been shown very useful to AI tasks including question answering, knowledge inference, and so on. However, traditional knowledge bases are symbolic and logic, thus numerical machine learning methods cannot be leveraged to support the computation over the knowledge bases. To this end, knowledge graph embedding, which projects entities and relations into continuous vector spaces, has been proposed. Among various embedding models, there is a line of translation-based models such as TransE (Bordes et al. 2013), TransH (Wang et al. 2014) and TransR (Lin et al. 2015).

A fact of knowledge base can usually be represented by a triple $(h, r, t)$ where $h, r, t$ indicates a head entity, a relation, and a tail entity, respectively. All translation-based models are almost following the same principle $\mathbf{h_r} + \mathbf{r} \approx \mathbf{t_r}$ where $\mathbf{h_r}, \mathbf{r}, \mathbf{t_r}$ indicate the embedding vectors of triple $(h, r, t)$, with the head and tail entity vector projected with respect to the relation space.

In spite of the success of these models, none of the previous models can deal with the ***multiple relation semantics*** that a relation may have multiple meanings revealed by the entity pairs associated with the corresponding triples. As can be seen from Fig. 1, clustering results on embedding vectors obtained from TransE (Bordes et al. 2013) show that, there are different clusters for a specific relation, and different clusters indicate different latent semantics. For example, the relation $\mathrm{HasPart}$ has at least two latent semantics: composition-related as $(\mathrm{Table}, \mathrm{HasPart}, \mathrm{Leg})$ and location-related as $(\mathrm{Atlantics}, \mathrm{HasPart}, \mathrm{NewYorkBay})$. This phenomenon is quite common in knowledge bases for two reasons: artificial simplification and nature of knowledge. On one hand, knowledge base curators could not involve too many similar relations, so abstracting multiple similar relations into one specific relation is a common trick. On the other hand, both language and knowledge representations often involve ambiguous information. The ambiguity of knowledge means a semantic mixture. For example, when we mention "Expert", we may refer to scientist, businessman or writer, so the concept "Expert" may be ambiguous in a specific situation, or generally a semantic mixture of these cases.

However, since previous translation-based models adopt $\mathbf{h_r} + \mathbf{r} \approx \mathbf{t_r}$, they assign only one translation vector for one relation, and these models cannot alleviate the issue of multiple relation semantics. To illustrate more clearly, as showed in Fig.2, there is only one unique representation for relation $\mathrm{HasPart}$ in traditional models, thus the models made more errors when embedding the triples of the relation. Instead, in our proposed model, we leverage a Gaussian mixture embedding model to handle multiple relation semantics by mixing multiple translation components for a relation. Thus, different semantics are characterised by different components in our mixture model. For example, we can distinguish the two clusters $\mathrm{HasPart\_1}$ or $\mathrm{HasPart\_2}$, where the relation semantics are automatically clustered to represent the meaning of associated entity pairs.

To summarize, our contributions are as follows:

- We address a new issue in knowledge graph embedding, ***multiple relation semantics*** that a relation in knowledge base may have different meanings revealed by the associated entity pairs, which has never been studied previously.
- We propose a novel Gaussian mixture embedding model, TransG, to address this issue. The model can automatically discover semantic clusters of a relation, and leverage a mixture of multiple relation components for translating

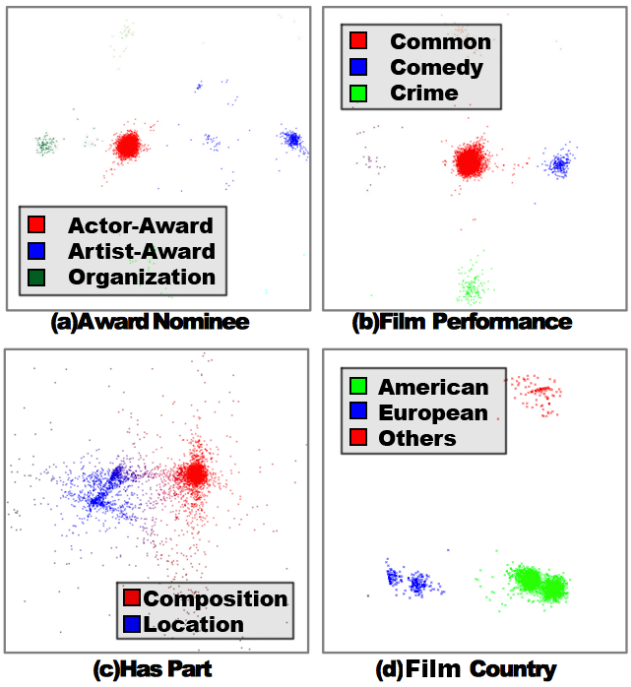


Figure 1: Visualization of TransE embedding vectors with PCA dimension reduction. Four relations (a $\sim$ d) are chosen from Freebase and Wordnet. A dot denotes a triple and its position is decided by the difference vector between tail and head entity ($\mathbf{t}-\mathbf{h}$). Since TransE adopts the principle of $\mathbf{t}-\mathbf{h}\approx\mathbf{r}$, there is supposed to be only one cluster whose centre is the relation vector $\mathbf{r}$. However, results show that there exist multiple clusters, which justifies our multiple relation semantics assumption.

an entity pair. Moreover, we present new insights from the generative perspective.

- Extensive experiments show that our proposed model obtains substantial improvements against the state-of-the-art baselines.

## Related Work

Prior studies are classified into two branches: translation-based embedding methods and the others.

### Translation-Based Embedding Methods

Existing translation-based embedding methods share the same translation principle $\mathbf{h}+\mathbf{r}\approx\mathbf{t}$ and the score function is designed as:

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

where $\mathbf{h_r},\mathbf{t_r}$ are entity embedding vectors projected in the relation-specific space. **TransE** (Bordes et al. 2013), lays the entities in the original entity space: $\mathbf{h_r}=\mathbf{h},\mathbf{t_r}=\mathbf{t}$. **TransH** (Wang et al. 2014), projects entities into a hyperplane for addressing the issue of complex relation embedding: $\mathbf{h_r}=\mathbf{h}-\mathbf{w_r}^\top\mathbf{h}\mathbf{w_r},\mathbf{t_r}=\mathbf{t}-\mathbf{w_r}^\top\mathbf{t}\mathbf{w_r}$. To address the same issue, **TransR** (Lin et al. 2015), transforms the entity embeddings by the same matrix: $\mathbf{h_r}=\mathbf{M_r}\mathbf{h},\mathbf{t_r}=\mathbf{M_r}\mathbf{t}$. TransR also proposes an ad-hoc clustering-based method, **CTransR**, where the entity pairs for a relation are clustered into different groups, and the pairs in the same group share the same relation vector. In comparison, our model is more elegant to address such an issue theoretically, and does not require a pre-process of clustering. Furthermore, our model has much better performance than CTransR, as we expect. **TransM** (Fan et al. 2014) leverages the structure of the knowledge graph via pre-calculating the distinct weight for each training triple to enhance embedding.

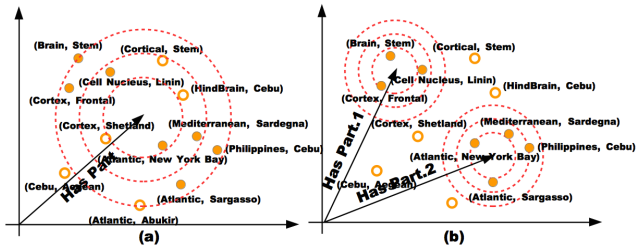


Figure 2: Visualization of multiple relation semantics. The data is selected from Wordnet. The dots are correct triples that belong to HasPart relation, while the circles are incorrect ones. The point coordinate is the difference vector between tail and head entity, which should be near to the centre. (a) The correct triples are hard to be distinguished from the incorrect ones. (b) By applying multiple semantic components, our proposed model could discriminate the correct triples from the wrong ones.

### Other Embedding Methods

There list other embedding approaches:

**Structured Embedding (SE).** The SE model (Bordes et al. 2011) transforms the entity space with the head-specific and tail-specific matrices. The score function is defined as $f_r(h,t)=||\mathbf{M_{h,r}}\mathbf{h}-\mathbf{M_{t,r}}\mathbf{t}||$. According to (Socher et al. 2013), this model cannot capture the relationship between entities.

**Semantic Matching Energy (SME).** The SME model (Bordes et al. 2012) (Bordes et al. 2014) can handle the correlations between entities and relations by matrix product and Hadamard product. In some recent work (Bordes et al. 2014), the score function is re-defined with 3-way tensors instead of matrices.

**Single Layer Model (SLM).** SLM applies neural network to knowledge graph embedding. The score function is defined as $f_r(h,t)=\mathbf{u_r}^\top g(\mathbf{M_{r,1}}\mathbf{h}+\mathbf{M_{r,2}}\mathbf{t})$ where $\mathbf{M_{r,1}},\mathbf{M_{r,2}}$ are relation-specific weight matrices. Collobert had applied a similar method into the language model, (Collobert and Weston 2008).

**Latent Factor Model (LFM).** The LFM (Jenatton et al. 2012), (Sutskever, Tenenbaum, and Salakhutdinov 2009) attempts to capture the second-order correlations between entities by a quadratic form. The score function is as $f_r(h,t)=\mathbf{h}^\top\mathbf{W_r}\mathbf{t}$.

**Neural Tensor Network (NTN).** The NTN model (Socher et al. 2013) defines a very expressive score function to combine the SLM and LFM: $f_r(h,t) = \mathbf{u_r}^\top g(\mathbf{h}^\top \mathbf{W}_{..\mathbf{r}}\mathbf{t} + \mathbf{M}_{\mathbf{r},\mathbf{1}}\mathbf{h} + \mathbf{M}_{\mathbf{r},\mathbf{2}}\mathbf{t} + \mathbf{b_r})$, where $\mathbf{u_r}$ is a relation-specific linear layer, $g(\cdot)$ is the $tanh$ function, $\mathbf{W} \in \mathbb{R}^{d\times d\times k}$ is a 3-way tensor.

**Unstructured Model (UM).** The UM (Bordes et al. 2012) may be a simplified version of TransE without considering any relation-related information. The score function is directly defined as $f_r(h,t) = ||\mathbf{h}-\mathbf{t}||_2^2$.

**RESCAL.** This is a collective matrix factorization model which is also a common method in knowledge base embedding (Nickel, Tresp, and Kriegel 2011), (Nickel, Tresp, and Kriegel 2012).

**Semantically Smooth Embedding (SSE).** (Guo et al. 2015) aims at further discovering the geometric structure of the embedding space to make it semantically smooth.

(Wang et al. 2014) jointly embeds knowledge and texts. (Wang, Wang, and Guo 2015) incorporates the rules. (Lin, Liu, and Sun 2015) is a path-based embedding model.

## Methods

In this section, we first review current translation-based methods from the generative perspective, and then introduce TransG, a generative mixture embedding model to capture multiple relation semantics.

### A Generative Perspective for Embedding

First of all, let's review previous translation-based embedding models. Existing methods obey the same rule $\mathbf{r} \approx \mathbf{t}-\mathbf{h}$ and apply Euclidean distance as loss metric. The score function is defined as follows:

$$f_r(h,t) = ||\mathbf{h}+\mathbf{r}-\mathbf{t}||_2^2 \tag{1}$$

Therefore, the generative process could be regarded that the difference vector between tail and head $(\mathbf{t}-\mathbf{h})$ is drawn from a Gaussian Distribution with $\mathbf{r}$ as mean and an identical matrix $\mathbf{E}$ as covariance.

$$\mathbf{t}-\mathbf{h} \mid \mathbf{r} \sim \mathcal{N}(\mathbf{r},\mathbf{E}) \tag{2}$$

$$\mathbb{P}\{(h,r,t)\} = \frac{1}{\mathcal{Z}} e^{-||\mathbf{h}+\mathbf{r}-\mathbf{t}||_2^2} \tag{3}$$

where $\mathcal{Z}$ is a normalization constant. If maximizing the data likelihood of the training set, we derive the TransE model. As discussed in "Related Work", different translation-based methods project entities into different vector spaces. Obviously, TransH and TransR have a similar generative process as below:

$$(\mathbf{t}-\mathbf{w_r}^\top\mathbf{t}\mathbf{w_r}) - (\mathbf{h}-\mathbf{w_r}^\top\mathbf{h}\mathbf{w_r}) \mid \mathbf{r} \sim \mathcal{N}(\mathbf{r},\mathbf{E}) \tag{4}$$

$$\mathbf{M_r}\mathbf{t}-\mathbf{M_r}\mathbf{h} \mid \mathbf{r} \sim \mathcal{N}(\mathbf{r},\mathbf{E}) \tag{5}$$

### TransG: A Generative Mixture Model for Embedding

As stated in "Introduction", only one single translation vector for a relation may be incompetent to model multiple relation semantics. This means, Eq.(2) should be generalised to a mixture model, such as Gaussian Mixture Model (Yu 2012). Thus, our TransG is proposed as follows:

$$\mathbf{t}-\mathbf{h} \mid \mathbf{r} \sim \sum_{m=1}^{M} \pi_{r,m}\mathcal{N}(\mathbf{u_{r,m}},\mathbf{E}) \tag{6}$$

$$\mathbb{P}\{(h,r,t)\} = \sum_{m=1}^{M} \pi_{r,m} e^{-||\mathbf{h}+\mathbf{u_{r,m}}-\mathbf{t}||_2^2} \tag{7}$$

where $M$ indicates how many components a relation should have. $\mathbf{u_{r,m}}$ is the $m$-th component translation vector of relation $r$. $\pi_{r,m}$ is the mixing factor, indicating the weight of $m$-th component. Note that, $\pi$ could be treated as the component prior in Bayesian Statistics, but TransG learns these coefficients from the data.

Inspired by Fig.1, TransG leverages a mixture of relation component vectors for a specific relation. Each component represents a specific latent semantics. By this way, TransG could distinguish multiple relation semantics. From the clustering perspective, TransG attempts to cluster the triples by their latent semantics automatically and then characterise each semantic component for a given entity pair $< h,t >$. Note that, our score function is the same as other translation-based methods, which is the probability of generating the triple $\mathbb{P}\{(h,r,t)\}$ and the optimization objective is the data likelihood.

### Perspective from Geometry

Previous studies always have geometric explanations, and so does TransG. In the previous methods, when the relation $r$ of triple $(h,r,t)$ is given, the geometric representations are fixed, as $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. However, TransG generalises this geometric principle to:

$$m^*_{(h,r,t)} = \arg\max_{m=1\ldots M} \left(\pi_{r,m} e^{-||\mathbf{h}+\mathbf{u_{r,m}}-\mathbf{t}||_2^2}\right) \tag{8}$$

$$\mathbf{h}+\mathbf{u}_{\mathbf{r},\mathbf{m}^*_{(\mathbf{h},\mathbf{r},\mathbf{t})}} \approx \mathbf{t} \tag{9}$$

where $m^*_{(h,r,t)}$ is the index of primary component. Though all the components contribute to the model, the primary one contributes the most due to the exponential effect $(exp(\cdot))$. When a triple $(h,r,t)$ is given, TransG works out the index of primary component then translates the head entity to the tail one with this worked-out primary translation vector.

For most triples, there should be only one component that have significant non-zero $\left(\pi_{r,m} e^{-||\mathbf{h}+\mathbf{u_{r,m}}-\mathbf{t}||_2^2}\right)$ value and the others would be small enough, due to the exponential decay. This property reduces the noise from the other semantic components to better characterise multiple relation semantics. In detail, $(\mathbf{t}-\mathbf{h})$ is almost around only one translation vector $\mathbf{u}_{\mathbf{r},\mathbf{m}^*_{(\mathbf{h},\mathbf{r},\mathbf{t})}}$ in TransG. Under the condition $m \neq m^*_{(h,r,t)}$, $||\mathbf{h}+\mathbf{u_{r,m}}-\mathbf{t}||_2^2$ is very large so that the exponential function value is very small. This is why the primary component could represent the primary semantics.

To summarize, previous studies make translation identically for all the triples of the same relation, but TransG automatically selects the best translation vector according to the specific semantics of a triple. Therefore, TransG could focus on the specific semantic embedding to avoid much noise

Table 1: Statistics of datasets

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

from the other unrelated semantic components and result in promising improvements than existing methods. Note that, all the components in TransG have their own contributions, but the primary one makes the most.

## Training Algorithm

The maximum data likelihood principle is applied for training. To better distinguish the true triples from the false ones, we maximize the ratio of likelihood of the true triples to that of the false ones. Notably, too many redundant components may do harm to embedding, and thus we impose a sparsity regularization term on $\pi_{r,m}$. Note that embedding vectors are initialized by (Glorot and Bengio 2010). Taking into account all the other constraints, the final objective function is obtained, as follows:

$$\begin{aligned}
\min \quad & -\sum_{(h,r,t)\in\Delta} ln\left(\sum_{m=1}^{M}\pi_{r,m}e^{-||\mathbf{h}+\mathbf{u_{r,m}}-\mathbf{t}||_2^2}\right)+ \\
& \sum_{(h',r',t')\in\Delta'} ln\left(\sum_{m=1}^{M}\pi_{r',m}e^{-||\mathbf{h'}+\mathbf{u_{r',m}}-\mathbf{t'}||_2^2}\right)+\Omega \\
s.t. \quad & \pi_{r,m}\geq 0,\ \ r\in R,\ \ m=1...M \\
& \Omega = C\left(\sum_{r\in R}\sum_{m=1}^{M}||\mathbf{u_{r,m}}||_2^2+\sum_{e\in E}||\mathbf{e}||_2^2\right) \\
& +\lambda\left(\sum_{r\in R}\sum_{m=1}^{M}||\pi_{\mathbf{r,m}}||_1\right) \qquad (10)
\end{aligned}$$

where $\Delta$ is the set of golden triples and $\Delta'$ is the set of false triples. $C$ controls the scaling degree and $\lambda$ controls the sparsity of components. $\Omega$ is the overall regularization term. $E$ is the set of entities and $R$ is the set of relations.

SGD is applied to solve this optimization problem. In addition, we apply a trick to control the parameter update process during training. For those very impossible triples, the update process should be skipped. Hence, we introduce a similar condition as TransE (Bordes et al. 2013) adopts: the training algorithm will update the embedding vectors only if the below condition is satisfied:

$$\frac{\mathbb{P}\{(h,r,t)\}}{\mathbb{P}\{(h',r',t')\}}=\frac{\sum_{m=1}^{M}\pi_{r,m}e^{-||\mathbf{h}+\mathbf{u_{r,m}}-\mathbf{t}||_2^2}}{\sum_{m=1}^{M}\pi_{r',m}e^{-||\mathbf{h'}+\mathbf{u_{r',m}}-\mathbf{t'}||_2^2}}\leq Me^{\gamma} \quad (11)$$

where $(h,r,t)\in\Delta$ and $(h',r',t')\in\Delta'$. $\gamma$ controls the updating condition.

# Experiments

Our experiments are conducted on four public benchmark datasets that are the subsets of Wordnet and Freebase, respectively. The statistics of these datasets are listed in Tab.1. Experiments are conducted on two tasks : Link Prediction and Triples Classification. To further demonstrate how the proposed model approaches multiple relation semantics, we present semantic component analysis in the last of this section.

## Link Prediction

In order to testify the performance of knowledge graph completion, link prediction task is conducted. When given an entity and a relation, the embedding models predict the other missing entity. More specifically, in this task, we predict $t$ given $(h,r,*)$, or predict $h$ given $(*,r,t)$. The WN18 and FB15K are two benchmark datasets for this task. Note that many AI tasks could be enhanced by Link Prediction such as relation extraction (Hoffmann et al. 2011).

**Evaluation Protocol.** We adopt the same protocol used in previous studies. For each testing triple $(h,r,t)$, we corrupt it by replacing the tail $t$ (or the head $h$) with every entity $e$ in the knowledge graph and calculate a probabilistic score of this corrupted triple $(h,r,e)$ (or $(e,r,t)$) with the score function $f_r(h,e)$. By ranking these scores in descending order, we then obtain the rank of the original triple. There are two metrics for evaluation: the averaged rank (Mean Rank) and the proportion of testing triple whose rank is not larger than 10 (HITS@10). This is called "Raw" setting. When we filter out the corrupted triples that exist in the training, validation, or test datasets, this is the"Filter" setting. If a corrupted triple exists in the knowledge graph, ranking it ahead the original triple is also acceptable. To eliminate this case, the "Filter" setting is more preferred. In both settings, a lower Mean Rank and a higher HITS@10 mean better performance.

**Implementation.** As the datasets are the same, we directly reproduce the experimental results of several baselines from the literature, as in (Bordes et al. 2013), (Wang et al. 2014) and (Lin et al. 2015). We have attempted several settings on the validation dataset to get the best configuration. Under the "bern." sampling strategy, the optimal configurations are: learning rate $\alpha=0.001$, embedding dimension $k=50$, $\gamma=0.2$, $M=10$, $\lambda=0.01$ on WN18; $\alpha=0.0015$, $k=400$, $\gamma=0.25$, $M=10$, $\lambda=0.01$ on FB15K. Note that all the symbols are introduced in "Methods". We train the model until it converges, but at most 10,000 rounds (usually converges at about 1,000 rounds).

**Results.** Evaluation results on WN18 and FB15K are reported in Tab.2 and Tab.4. We observe that:

1. TransG outperforms all the baselines obviously. Compared to TransR, TransG makes improvements by 2.9% on WN18 and 23.0% on FB15K, and the averaged semantic component number on WN18 is 2.61 and that on FB15K is 5.39. This result demonstrates capturing multiple relation semantics would benefit embedding.
2. The model has a bad Mean Rank score on the WN18 dataset. Further analysis shows that there are 24 testing triples (0.5% of the testing set) whose ranks are more than

Table 2: Evaluation results on link prediction

| Datasets | WN18 | | | | FB15K | | | |
|---|---|---|---|---|---|---|---|---|
| Metric | Mean Rank | | HITS@10(%) | | Mean Rank | | HITS@10(%) | |
| | Raw | Filter | Raw | Filter | Raw | Filter | Raw | Filter |
| SE(Bordes et al. 2011) | 1,011 | 985 | 68.5 | 80.5 | 273 | 162 | 28.8 | 39.8 |
| SME(linear) (Bordes et al. 2012) | 545 | 533 | 65.1 | 74.1 | 274 | 154 | 30.7 | 40.8 |
| SME(bilinear) (Bordes et al. 2012) | 526 | 509 | 54.7 | 61.3 | 284 | 158 | 31.3 | 41.3 |
| LFM (Jenatton et al. 2012) | 469 | 456 | 71.4 | 81.6 | 283 | 164 | 26.0 | 33.1 |
| TransE (Bordes et al. 2013) | 263 | 251 | 75.4 | 89.2 | 243 | 125 | 34.9 | 47.1 |
| TransH (Wang et al. 2014) | 401 | 388 | 73.0 | 82.3 | 212 | 87 | 45.7 | 64.4 |
| TransR (Lin et al. 2015) | 238 | 225 | 79.8 | 92.0 | 198 | 77 | 48.2 | 68.7 |
| CTransR (Lin et al. 2015) | **231** | **218** | 79.4 | 92.3 | 199 | 75 | 48.4 | 70.2 |
| TransG (this paper) | 377 | 365 | **82.5** | **94.7** | **152** | **56** | **55.9** | **84.7** |

Table 3: triples classification: accuracy(%) for different embedding methods.

| Methods | WN11 | FB13 | AVG. |
|---|---|---|---|
| SE | 53.0 | 75.2 | 64.1 |
| SME(bilinear) | 70.0 | 63.7 | 66.9 |
| LFM | 73.8 | 84.3 | 79.0 |
| NTN | 70.4 | 87.1 | 78.8 |
| TransE | 75.9 | 81.5 | 78.7 |
| TransH | 78.8 | 83.3 | 81.1 |
| TransR | 85.9 | 82.5 | 84.2 |
| CTransR | 85.7 | N/A | N/A |
| TransG | **87.4** | **87.3** | **87.4** |

30,000, and these few cases would lead to about 150 mean rank loss. Among these triples, there are 23 triples whose tail or head entities have never been co-occurring with the corresponding relations in the training set. In one word, there is no sufficient training data for those relations and entities. Since TransG applies multiple relation components and the clustering centers are more scattered in the vector space while other models just apply single relation vector, TransG is less competitive to handle these extreme cases.

3. Compared to CTransR, TransG solves the multiple relation semantics problem much better for two reasons. Firstly, CTransR clusters the entity pairs for a specific relation then performs embedding for each cluster, but TransG deals with embedding and multiple relation semantics at the same time, where the two processes can be enhanced by each other. Secondly, CTransR models a triple by only one cluster, but TransG applies a mixture to refine the embedding.

## Triples Classification

In order to testify the discriminative capability between true and false facts, triples classification task is conducted. This is a classical task in knowledge base embedding, which aims at predicting whether a given triple $(h, r, t)$ is correct or not. WN11 and FB13 are the benchmark datasets for this task.

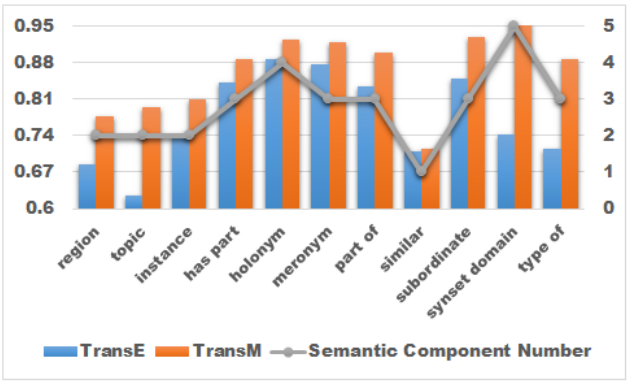


Figure 3: Accuracies of each relations in WN11 for triples classification. We count the components with significant non-zero $\pi_{r,m}$ as "Semantic Component Number" for each relation.

Note that evaluation of classification needs negative samples, and the datasets have already been built with negative triples.

**Evaluation Protocol.** The decision process is very simple as follows: for a triple $(h, r, t)$, if $f_r(h, t)$ is below a threshold $\sigma_r$, then positive; otherwise negative. The thresholds $\{\sigma_r\}$ are determined on the validation dataset.

**Implementation.** As all methods use the same datasets, we directly re-use the results of different methods from the literature. We have attempted several settings on the validation dataset to find the best configuration. The optimal configurations of TransG are as follows: "bern" sampling, learning rate $\alpha = 0.001$, embedding dimension $k = 50$, $\gamma = 6.0$, $M = 10$, $\lambda = 0.01$ on WN11, and "bern" sampling, $\alpha = 0.002$, $k = 400$, $\gamma = 0.2$, $M = 10$, $\lambda = 0.01$ on FB13. We limit the maximum number of epochs to 500 but the algorithm usually converges at around 100 epochs.

**Results.** Accuracies are reported in Tab.3 and Fig.3. We observe that:

1. TransG outperforms all the baselines remarkably. Compared to TransR, TransG improves by 1.7% on WN11 and 5.8% on FB13, and the averaged semantic compo-

Table 4: Evaluation results on FB15K by mapping properties of relations(%)

| Tasks | Predicting Head(HITS@10) | | | | Predicting Tail(HITS@10) | | | |
|---|---|---|---|---|---|---|---|---|
| Relation Category | 1-1 | 1-N | N-1 | N-N | 1-1 | 1-N | N-1 | N-N |
| SE(Bordes et al. 2011) | 35.6 | 62.6 | 17.2 | 37.5 | 34.9 | 14.6 | 68.3 | 41.3 |
| SME(linear) (Bordes et al. 2012) | 35.1 | 53.7 | 19.0 | 40.3 | 32.7 | 14.9 | 61.6 | 43.3 |
| SME(bilinear) (Bordes et al. 2012) | 30.9 | 69.6 | 19.9 | 38.6 | 28.2 | 13.1 | 76.0 | 41.8 |
| TransE (Bordes et al. 2013) | 43.7 | 65.7 | 18.2 | 47.2 | 43.7 | 19.7 | 66.7 | 50.0 |
| TransH (Wang et al. 2014) | 66.8 | 87.6 | 28.7 | 64.5 | 65.5 | 39.8 | 83.3 | 67.2 |
| TransR (Lin et al. 2015) | 78.8 | 89.2 | 34.1 | 69.2 | 79.2 | 37.4 | 90.4 | 72.1 |
| CTransR (Lin et al. 2015) | 81.5 | 89.0 | 34.7 | 71.2 | 80.8 | 38.6 | 90.1 | 73.8 |
| TransG (this paper) | **85.9** | **96.5** | **46.5** | **82.0** | **85.4** | **60.1** | **95.5** | **85.4** |

Table 5: Different clusters in WN11 and FB13 relations.

| Relation | Cluster | triples (Head, Tail) |
|---|---|---|
| PartOf | Location | (Capital of Utah, Beehive State), (Hindustan, Bharat), (Hoover Dam, Battle Born State) ... |
| | Composition | (Monitor, Television), (Bush, Adult Body), (Cell Organ, Cell), (Indian Rice, Wild Rice)... |
| Religion | Catholicism | (Cimabue, Catholicism), (Bruno Heim, Catholicism), (St.Catald, Catholicism) ... |
| | Others | (Michal Czajkowsk, Islam), (Honinbo Sansa, Buddhism), (Asmahan, Druze) ... |
| DomainRegion | Abstract | (Computer Science, Security System), (Computer Science, Programming Language).. |
| | Specific | (Computer Science, Router), (Computer Science, Disk File), (Psychiatry, Isolation) ... |
| Profession | Scientist | (Michael Woodruf, Surgeon), (El Lissitzky, Architect), (Charles Wilson, Physicist)... |
| | Businessman | (Enoch Pratt, Entrepreneur), (Charles Tennant, Magnate), (Joshua Fisher, Businessman)... |
| | Writer | (Vlad. Gardin, Screen Writer), (John Huston, Screen Writer), (Martin Fri, Screen Writer) ... |

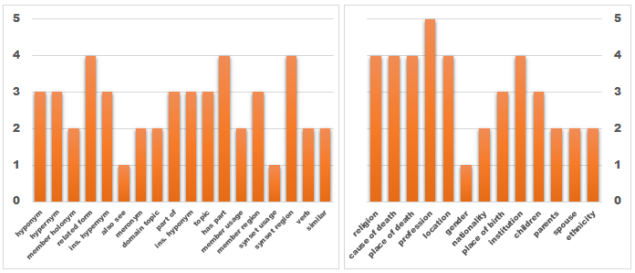


Figure 4: Semantic component number on WN18 (left) and FB13 (right). For each relation, we only count the components with non-zero $\pi_{r,m}$.

nent number on WN11 is 2.72 and that on FB13 is 3.08. This result shows the benefit of capturing multiple relation semantics for a relation.

2. The relations, such as "Synset Domain" and "Type Of", which hold more semantic components, are improved much more. Conversely, the relation "Similar" holds only one semantic component and is almost not promoted. This further demonstrates that capturing multiple relation semantics can benefit embedding.

## Semantic Component Analysis

In this subsection, we analyse the number of semantic components for different relations. Since sparsity is considered in our approach, we list the component number on the dataset WN18 and FB13 in Fig.4 where only the component with non-zero $\pi_m$ is counted.

**Results.** As Fig. 4 and Tab. 5 illustrates, we observe that:

1. Multiple semantic components are indeed necessary for most relations. Except for relations such as "Also See", "Synset Usage" and "Gender", all other relations have more than one semantic component.
2. Different components indeed correspond to different semantics, justifying the theoretical analysis and effectiveness of TransG. For example, "Profession" has at least three significant semantics: science-related as $(\text{ElLissitzky}, \text{Architect})$, business-related as $(\text{EnochPratt}, \text{Entrepreneur})$ and writer-related as $(\text{Vlad.Gardin}, \text{ScreenWriter})$.
3. WN11 and WN18 are the different subsets of Wordnet. As we know, the semantic component number is decided on the triples in the dataset. Therefore, It's reasonable that similar relations, such as "Synset Region" and "Synset Usage" may hold different semantic numbers for WN11 and WN18.

## Conclusion

In this paper, we address a new issue, multiple relation semantics, and propose TransG, a generative mixture embedding model for knowledge graph embedding. TransG can discover the latent semantics of a relation automatically and leverage a mixture of relation components for embedding. Extensive experiments show our method achieves substantial improvements against the state-of-the-art baselines. To reproduce our results, experimental codes and data will be published in github[1].

[1]http://www.github.com

## References


[Bollacker et al. 2008] Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. ACM.

[Bordes et al. 2011] Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y.; et al. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence*.

[Bordes et al. 2012] Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*, 127–135.

[Bordes et al. 2013] Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 2787–2795.

[Bordes et al. 2014] Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 94(2):233–259.

[Collobert and Weston 2008] Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167. ACM.

[Fan et al. 2014] Fan, M.; Zhou, Q.; Chang, E.; and Zheng, T. F. 2014. Transition-based knowledge graph embedding with relational mapping properties. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, 328–337.

[Glorot and Bengio 2010] Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, 249–256.

[Guo et al. 2015] Guo, S.; Wang, Q.; Wang, B.; Wang, L.; and Guo, L. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of ACL*.

[Hoffmann et al. 2011] Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 541–550. Association for Computational Linguistics.

[Jenatton et al. 2012] Jenatton, R.; Roux, N. L.; Bordes, A.; and Obozinski, G. R. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, 3167–3175.

[Lin et al. 2015] 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 Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[Lin, Liu, and Sun 2015] Lin, Y.; Liu, Z.; and Sun, M. 2015. Modeling relation paths for representation learning of knowledge bases. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics*.

[Miller 1995] Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

[Nickel, Tresp, and Kriegel 2011] Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 809–816.

[Nickel, Tresp, and Kriegel 2012] Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, 271–280. ACM.

[Socher et al. 2013] Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 926–934.

[Sutskever, Tenenbaum, and Salakhutdinov 2009] Sutskever, I.; Tenenbaum, J. B.; and Salakhutdinov, R. 2009. Modelling relational data using bayesian clustered tensor factorization. In *Advances in neural information processing systems*, 1821–1828.

[Wang et al. 2014] Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.

[Wang, Wang, and Guo 2015] Wang, Q.; Wang, B.; and Guo, L. 2015. Knowledge base completion using embeddings and rules. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.

[Yu 2012] Yu, J. 2012. A nonlinear kernel gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes. *Chemical Engineering Science* 68(1):506–519.