# Enriching Hindi WordNet Using Knowledge Graph Completion Approach

**Sushil Awale, Abhik Jana**
Universität Hamburg, Germany
sushil.awale@studium.uni-hamburg.de, abhik.jana@uni-hamburg.de

**Abstract**

Even though the use of WordNet in the Natural Language Processing domain is unquestionable, creating and maintaining WordNet is a cumbersome job and it is even difficult for low resource languages like Hindi. In this study, we aim to enrich the Hindi WordNet automatically by using state-of-the-art knowledge graph completion (KGC) approaches. We pose the automatic Hindi WordNet enrichment problem as a knowledge graph completion task and therefore we modify the Wordnet structure to make it appropriate for applying KGC approaches. Second, we attempt five KGC approaches of three different genres and compare the performances for the task. Our study shows that ConvE is the best KGC methodology for this specific task compared to other KGC approaches.

**Keywords:** Hindi WordNet, Knowledge Graph Completion

## 1. Introduction

The usability of WordNet for various Natural Language Processing tasks such as word-sense disambiguation, information retrieval, machine translation, etc. is well-known to the research community. The first-ever WordNet was created for the English language in 1985 at the Princeton University (Fellbaum and others, 1998), and over time researchers have invested efforts to develop WordNets for various languages. As WordNet is updated and maintained manually, it requires frequent revision which is an expensive and time-consuming task. Therefore there is a need for automatic enrichment of WordNet.

In this paper, we focus on the enrichment of Hindi (One of the low-resource languages from India) WordNet (Narayan et al., 2002). Formally, WordNet (Miller et al., 1990) is represented as a lexical graph database where the nodes represent synsets and the edges between them represent the type of relation. Synsets refer to a collection of synonymous words, and the relations that link the synsets include synonyms, hyponyms, meronyms, etc. On the other hand, knowledge graphs are graph structures that represent facts in the world. The facts are represented as triples $(h, r, t)$, where $h$ and $t$ represent the head and tail entity, and $r$ represents the relation between them. For example, *(Hamburg, cityIn, Germany)* is a fact in a knowledge graph. There is a genre of literature (Rossi et al., 2021), which deals with the completion of knowledge graphs where new edges or nodes are added to complete the knowledge graphs. In this work, we are considering WordNet as a type of knowledge graph with the synsets representing nodes and the lexical relations representing the edges. We explore the applicability of five knowledge graph completion techniques namely, TransE (Bordes et al., 2013), TransH (Wang et al., 2014), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), and ConvE (Dettmers et al., 2018). After experimenting with all these models, we see that ConvE produces the best performance for the Hindi WordNet completion task with an MRR value of 0.294 and Hit@10 of 0.385 which is promising to move forward in this direction. In a nutshell, the main contributions of this paper are twofold-

- We pose the enrichment of Hindi WordNet as a knowledge graph completion task which is the first-ever attempt for Hindi WordNet to the best of our knowledge.
- We attempt with five KGC approaches and compare their performances and show ConvE performs better than all the four other models by a significant margin. We make all the code and data publicly available [1].

## 2. Related Work

Knowledge Graph Completion (KGC) is a widely researched topic in the Natural Language Processing domain. Several KGC approaches have been proposed in the domain which includes rule-based reasoning, probabilistic graph, graph calculation, and representation learning approaches. (Chen et al., 2020) In one of the earliest rule-based reasoning methods Paulheim and Bizer (2014) deduces new relational instances from existing knowledge using a first-order relational learning algorithm. Rule-based reasoning approaches are difficult to generalize and scale. Similarly, probabilistic graph-based approaches that use joint probability distribution reasoning to predict new facts are also difficult to scale due to the high complexity of the models. These models mostly use Markov Logic Networks (Lao and Cohen, 2010) and Bayesian network (Han et al., 2017). In the graph calculation approaches, new relations are predicted based on incoming and outgoing node degrees and using an adjacent matrix. Path Ranking Algorithm (Nickel et al., 2011) is one of the

[1] `https://github.com/uhh-lt/hindi-wordnet-extension`

earlier graph completion methods followed by Coupled Path Ranking Algorithm (Hayashi and Shimbo, 2017). The traditional KGC methods suffered from the problem of computational efficiency, high algorithm complexity, and poor scalability. (Chen et al., 2020) As a result, researchers have shifted towards knowledge representation learning.

A few surveys focus on representation learning (Wang et al., 2021; Rossi et al., 2021). In one such study, Rossi et al. (2021) compare different knowledge graph (KG) embedding models based on effectiveness and efficiency. The KG models are grouped into three categories by their learning methods namely, tensor decomposition models, geometric models, and deep learning models. In another recent work, Wang et al. (2021) provides a theoretical analysis of different KG models and classifies the KG models into three-main categories based on the type of scoring function used, e.g. distance-based or semantic-matching-based. This work also compares the performance of the models on two popular English KG datasets, WN18RR and FB15K-237. In our study, we follow the classification presented in this paper to select at most two KG models from each class for our study.

In addition, other approaches to enrich a WordNet have been proposed. Montoyo et al. (2001) automatically adds new categories, drawn from classification systems, to an English WordNet using Word Sense Disambiguation. Giménez and Màrquez (2006) enriches Spanish WordNet by automatic addition of synset glosses obtained from English WordNet glosses using a phrase-based English-Spanish statistical machine translation system. Other works focus on using lexical patterns to extract new relations from the unlabelled corpus. Ruiz-Casado et al. (2007) enriches the English WordNet with the addition of new relations between synsets using the edit distance-based method to generate lexical patterns from raw text. Boudabous et al. (2013) adds new semantic relations in the Arabic WordNet using expert-defined morpho-lexical patterns.

# 3. Methodology

As a part of our exploration, we aim to enrich the Hindi WordNet by adding the missing edges in the WordNet graph. Following the well-known direction of knowledge graph completion approaches, firstly, we embed the WordNet graph into a continuous vector space where each entity ($h$ or $t$) is represented as a point in the vector space, and each relation $r$ represents an operation (translation, rotation, etc.) in the vector space. The entity and relation representations are learned by minimizing a global loss function which involves all the entities and relations in the graph. Next, these embedding representations are used for the task of link prediction.

## 3.1. Link Prediction

Link prediction is the task of predicting the missing entity in a triple $(h, r, t)$, i.e. predict $h$ given $(r, t)$ or predict $t$ given $(h, r)$. When predicting the missing entity, we replace it with all entities from the knowledge graph and rank them based on a scoring function. A higher score indicates that the triple is more likely to be true. We experiment with KGC models from different genres depending on the scoring function used.

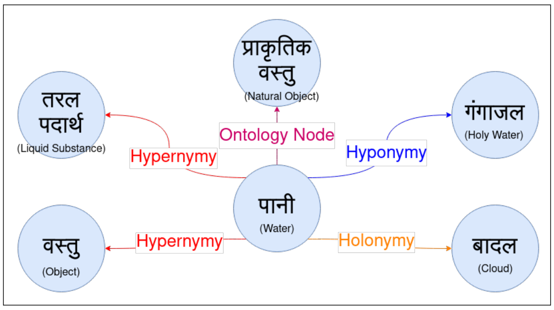


Figure 1: A snapshot of Hindi WordNet as a Knowledge Graph

### 3.1.1. Translation-distance-based models

The translation-distance-based models use some distance-based scoring functions for link prediction.

**TransE:** TransE (Bordes et al., 2013) is one of the first and simple translation-distance-based model. Given a triple $(h, r, t)$ where $h$, $t \in E$ (set of entities) and $r \in R$ (set of relationships), TransE learns vector embeddings $\boldsymbol{h}, \boldsymbol{t}$ and $\boldsymbol{r}$ such that distance between $\boldsymbol{h} + \boldsymbol{r}$ and $\boldsymbol{t}$ is minimum. In TransE, $L1$ or $L2$ norm is used to measure the distance, $d(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = ||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||$. To learn the embeddings, the following loss-function is minimized over the training set:

$$L = \sum_{(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})} \sum_{(\boldsymbol{h'},\boldsymbol{r},\boldsymbol{t'})} [\gamma + d(\boldsymbol{h}+\boldsymbol{r},\boldsymbol{t}) - d(\boldsymbol{h'}+\boldsymbol{r},\boldsymbol{t'})]_+ \tag{1}$$

where $[x]_+$ denotes the positive part of $x$, $\gamma > 0$ is a margin hyperparameter, and $d(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ is the distance of a positive sample, and $d(\boldsymbol{h'}, \boldsymbol{r}, \boldsymbol{t'})$ is the distance of a negative sample.

**TransH:** TransH introduces two vectors for a relation $\boldsymbol{r}$, a relation-specific-specific translation vector $\boldsymbol{d_r}$ and a relation-specific hyperplane $w_r$. Then the embedding vectors of head $\boldsymbol{h}$ and tail $\boldsymbol{t}$ are projected to the hyperplane which gives new vectors $\boldsymbol{h}\perp$ and $\boldsymbol{t}\perp$ respectively. Then the scoring function to measure the plausibility of a triple is defined as $f_r(\boldsymbol{h}, \boldsymbol{t}) = ||\boldsymbol{h}\perp + \boldsymbol{d_r} - \boldsymbol{t}\perp||$. When $||w_r||_2 = 1$ is restricted, we get,

$$\boldsymbol{h}\perp = \boldsymbol{h} - w_r^\top \boldsymbol{h} w_r, \qquad \boldsymbol{t}\perp = \boldsymbol{t} - w_r^\top \boldsymbol{t} w_r \tag{2}$$

Then, we have the scoring function as,

$$f_r(\boldsymbol{h}, \boldsymbol{t}) = ||(\boldsymbol{h} - w_r^\top \boldsymbol{h} w_r) + \boldsymbol{d_r} - (\boldsymbol{t} - w_r^\top \boldsymbol{t} w_r)|| \tag{3}$$

Now, the model is trained over the following the loss function,

$$L = \sum_{(\boldsymbol{h},\boldsymbol{r'},\boldsymbol{t})} \sum_{(\boldsymbol{h'},\boldsymbol{r},\boldsymbol{t'})} [\gamma + f_r(\boldsymbol{h}, \boldsymbol{t}) - f_r(\boldsymbol{h'}, \boldsymbol{t'})]_+ \tag{4}$$

where $[x]_+$ denotes the positive part of $x$, $\gamma$ is the margin separating positive and negative triples.

### 3.1.2. Semantic-matching-based models

Semantic-matching-based models use similarity-based scoring functions or add additional information to extract more knowledge.

**DistMult:** DistMult (Yang et al., 2014) is a semantic-matching-based multiplicative model in which the relationship vector is enforced to be a diagonal matrix. The head entity $\boldsymbol{h}$ and tail entity $\boldsymbol{t}$ are initialized as either a "one-hot" vector or an "n-hot" feature vector. Then the learned representations, $\boldsymbol{y_h} \in R$ and $\boldsymbol{y_t} \in R$ are given by, $\boldsymbol{y_h} = f(W\boldsymbol{h})$, and $\boldsymbol{y_t} = f(W\boldsymbol{t})$, where $f$ can be a linear or non-linear function, and $W$ is the parameter matrix which can be randomly initialized or initialized using pre-trained vectors.
The relation, similar to previously discussed models, is represented in the form of a scoring function. In DistMult, the function is formulated as bilinear,

$$S(\boldsymbol{y_h}, \boldsymbol{y_t}) = \boldsymbol{y_h}^T \boldsymbol{M_r} \boldsymbol{y_t} \tag{5}$$

where, $\boldsymbol{M_r} \in R^{nxn}$ is a matrix operator and is restricted to be a diagonal matrix.

$$L = \sum_{(h,r,t)} \sum_{(h',r,t')} max\{S_{(h',r,t')} - S_{(h,r,t)} + 1, 0\} \tag{6}$$

**ComplEx:** ComplEx (Trouillon et al., 2016) is another semantic-matching-based multiplicative model which follows the idea of forcing the relation embedding to be a diagonal matrix similar to DistMult. However, in ComplEx, the concept is extended in the complex space and as a result, the bi-linear product becomes a Hermitian product. In ComplEx, the set of entities is represented as $\epsilon$ with $|\epsilon| = n$ and the relation between two entities, head $h$ and tail $t$ is represented as a binary value $Y_{ht} \in \{-1, 1\}$. Its probability is given by the logistic inverse link function $P(Y_{ht} = 1) = \sigma(X_{ht})$, where $X \in R^{n \times n}$ is a latent matrix of scores, and $Y$ the partially observed sign matrix. The scoring function used in ComplEx is given by

$$\begin{aligned}\phi(r,h,t;\theta) =& < Re(w_r), Re(h), Re(t) > + \\ & < Re(w_r), Im(h), Im(t) > + \\ & < Im(w_r), Re(h), Im(t) > - \\ & < Im(w_r), Im(h), Re(t) >\end{aligned} \tag{7}$$

where $w_r$ in $C^k$ is a complex vector.
An advantage of projecting the embeddings in the complex space is it disables the commutative property of the scoring function that existed in DistMult.

### 3.2. Neural Network based models

**ConvE:** ConvE (Dettmers et al., 2018) is the first neural network-based model that applies a simple convolution over the entity embeddings. The entity embedding and the relation embedding are concatenated together before passing through the convolution layer with a set $\boldsymbol{W}$ of $m \times n$ filters. The output of the convolution layer is then fed into a dense layer with a single neuron and weights $\boldsymbol{W}$, giving out a fact score. In ConvE, the scoring function is defined by a convolution over the embeddings as follows:

$$(\boldsymbol{h}, \boldsymbol{t}) = f(vec(f(\bar{\boldsymbol{h}}; \bar{\boldsymbol{r}}) * w))W)t \tag{8}$$

where $w$ is a relation parameter, $\bar{h}$ and $\bar{r}$ denote 2D reshaping of $h$ and $r$ respectively.
The model is trained using logistic sigmoid function $p = \sigma(.)$ to the scores, and minimize the binary cross-entropy loss:

$$L(p,l) = -\frac{1}{N}\sum_i l_i log(p_i) + (1 - l_i) log(1 - p_i)) \tag{9}$$

where $l$ is the label vector.

| Relation | #(Synset) |
|---|---|
| ONTO_NODES | 44,857 |
| HYPERNYM | 33,972 |
| HYPONYM | 30,836 |
| MODIFIES_NOUN | 9,780 |
| ALSO_SEE | 1,814 |

Table 1: Statistics of top five relations from Hindi Wordnet with number of synsets.

| Model | MRR | H@1 | H@3 | H@10 |
|---|---|---|---|---|
| TransE | 0.156 | 0.055 | 0.221 | 0.334 |
| TransH | 0.133 | 0.032 | 0.199 | 0.308 |
| DistMult | 0.166 | 0.119 | 0.19 | 0.24 |
| ComplEx | 0.172 | 0.13 | 0.188 | 0.237 |
| ConvE | **0.294** | **0.24** | **0.316** | **0.385** |

Table 2: Performance of five KGC models

| Relation | MRR | Hit@10 |
|---|---|---|
| ONTO_NODES | 0.267 | 0.418 |
| VERB | 0.239 | 0.436 |
| ANTONYM | 0.177 | 0.2 |
| MODIFIES_NOUN | 0.16 | 0.405 |
| HYPERNYM | 0.142 | 0.241 |

Table 3: Top 5 relations based on MRR score of ConvE model

## 4. Experimental Results and Analysis

Our experimental framework with all the details are mentioned below.

**Hindi WordNet:** For our study, we take the Hindi WordNet developed as part of the Indo WordNet at the Center For Indian Languages Technology (Narayan et al., 2002). A snapshot of the Hindi WordNet is shown in Figure 1. The Hindi WordNet consists of 39,622 synsets with a total of 59 relations. The total amount of words in the WordNet amount to 148,865 with 103,365 unique words. The top five relations based on synset count are shown in Table 1.

| Model | 1-1 relations | | n-1 relations | | n-n relations | |
|---|---|---|---|---|---|---|
| | MRR | Hit@10 | MRR | Hit@10 | MRR | Hit@10 |
| TransE | 0.036 | 0.089 | 0.168 | 0.34 | 0.114 | 0.344 |
| TransH | 0.03 | 0.0874 | 0.137 | 0.31 | 0.116 | 0.35 |
| DistMult | 0.007 | 0.009 | 0.16 | 0.24 | 0.103 | 0.262 |
| ComplEx | 0.015 | 0.0336 | 0.18 | 0.25 | 0.154 | 0.212 |
| ConvE | **0.11** | **0.146** | **0.212** | **0.34** | **0.167** | **0.407** |

Table 4: MRR, and Hit@10 of the models for 1-1, n-1 and n-n relations

**Dataset:** We convert the WordNet into RDF-style [2] triples graph fitting for the Knowledge Graph Completion task. An RDF-style triples graph consists of a triple in the form ($head$, $relation$, $tail$), where $head$ and $tail$ are synset ids and $relation$ is the relationship that exists between the two synsets. Following Dettmers et al. (2018), we remove the inverse relations from our dataset to correctly evaluate the performance of models. Therefore, we simply remove the triples with obvious inverse relations like hyponym and holonym from the dataset. In addition, we also manually remove triples $(h, r, t)$ from the valid and test set, if $(h, r', t)$ exists in the train set. Moreover, we also narrow the 59 relations present in the Hindi WordNet to 16 relations by grouping relations. For example, we merge all the different types (ANTONYM_SIZE, ANTONYM_TIME, ANTONYM_ACTION) of antonym relations into the relation ANTONYM. The final dataset consists 39,609 entites with 16 relations and it comprises of 86,432 train, 4,712 valid, and 4,694 test triples.

Furthermore, we follow Bordes et al. (2013) and categorize the relationships in the test dataset into four categories based on cardinalities of their head and tail arguments. The four categories include $1-1$, $1-n$, $n-1$ and $n-n$ relations. In $1-1$ relationship, a head can appear with at most one tail. For example, relationships showing capital of countries like *(paris, capital_of, france)*, *(madrid, capital_of, spain)*, etc. In $1-n$ relationship, a head can appear with many tails. For example, *(germany, has_part, dusseldorf)*, *(germany, has_part, hanover)*, etc. In $n-1$ relationship, many heads can appear with the same tail. For example, *(wintertime, hypernym, time)*, *(years, hypernym, time)*, etc. and in $n-n$ relationship, multiple heads can appear with multiple tails. For example, *(run, derivationally_related_form, atrium)*, *(run, derivationally_related_form, runner)*, etc.

**Experimental Setup:** We run the TransE, TransH, DistMult and ComplEx models using the OpenKE toolkit (Han et al., 2018). We run all experiments using default settings. For ConvE, we run the model published in GitHub[3]. All these models were run on a Ubuntu 20.04.2 LTS server with NVIDIA GeForce RTX 2080 Ti GPU, and 256 GB RAM.

**Results and Analysis:** For our experiments, we report the performance using Mean Reciprocal Rank (MRR) and Hits@(1, 3, 10) on the *filtered* setting (Bordes et al., 2013). The performance of all five models is presented in Table 2.

We evaluate the models on the metric score $Hit@k$ with $k = 1, 3$ and 10. In general, the lower values of $k$ better indicate the performance of the models. At $k = 10$, we observe good performance from the TransE and ConvE models, whereas at $k = 3$ and $k = 1$, the ConvE models outperform all the other models significantly. In Table 3, we look at the performance of ConvE model which shows that the model does well with relations such as $onto\_nodes$, $verb$, $antonym$ which have a higher triple count in the training set. Clearly the relation class imbalance inherent in the WordNet is affecting the overall performance of the system. Some examples of induced triples from ConvE model with the rank 1 are *(दुहाना, onto_nodes, पियानो)*, *(अवसर, hypernym, समय)*, *(जन्मा, modifies_noun, जीव)* and some examples with the rank above 10 are *(रुलाई, antonym, हँसी)*, *(गुग्गुल, attributes, सुगंधित)*, *(लोकसेवा_आयोग, hypernym,समिति)*.

Further, we test the performance of the models on the different categories of the relationships discussed in Section 4. The results are shown in Table 4, with the best score marked in boldface. We do not evaluate the performance on $1-n$ relations as the test dataset contained only a few test instances.

In our results, we observe that the ConvE model outperforms the transition-distance-based models and the semantic-distance-based models. The ConvE model achieves a higher score across all relation-types signaling better generalization ability of the model.

## 5. Conclusion

In this study, we attempted to enrich Hindi WordNet by posing it as a knowledge graph completion task. We prepared a dataset from Hindi WordNet for evaluating such an attempt. We experimented with five Knowledge Graph Completion models of different genres and report the overall performances of all the models. We showed that the ConvE model outperforms all the other models across all relation types. However, to develop a fully automated Hindi WordNet enrichment process the evaluation results have to be improved. Hence, the next step would be to investigate approaches to mitigate relation-class imbalance in the WordNet dataset, and in future study extend WordNet enrichment for other low resource languages as well.

[2] `https://www.w3.org/RDF/`

[3] `https://github.com/TimDettmers/ConvE`

## 6. Bibliographical References


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

Boudabous, M. M., Kammoun, N. C., Khedher, N., Belguith, L. H., and Sadat, F. (2013). Arabic wordnet semantic relations enrichment through morpho-lexical patterns. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6. IEEE.

Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., and Duan, Z. (2020). Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Fellbaum, C. et al. (1998). Wordnet: An electronic lexical database mit press. *Cambridge, Massachusetts*.

Giménez, J. and Màrquez, L. (2006). Low-cost enrichment of spanish wordnet with automatically translated glosses: combining general and specialized models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 287–294.

Han, L., Yin, Z., Wang, Y., Hu, K., et al. (2017). Link prediction of knowledge graph based on bayesian network. *J. Frontiers Comput. Sci. Technol.*, 11(5):742–751.

Han, X., Cao, S., Xin, L., Lin, Y., Liu, Z., Sun, M., and Li, J. (2018). Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*.

Hayashi, K. and Shimbo, M. (2017). On the equivalence of holographic and complex embeddings for link prediction. *arXiv preprint arXiv:1702.05563*.

Lao, N. and Cohen, W. W. (2010). Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Montoyo, A., Palomar, M., Rigau, G., and Gargalló, P. (2001). Wordnet enrichment with classification systems. In *Proceedings of WordNet and Other Lexical Resources: Applications, Extensions and Customisations Workshop.(NAACL-01) The Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 101–106.

Narayan, D., Chakrabarti, D., Pande, P., and Bhattacharyya, P. (2002). An experience in building the indo wordnet-a wordnet for hindi. In *First International Conference on Global WordNet, Mysore, India*, volume 24.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816.

Paulheim, H. and Bizer, C. (2014). Improving the quality of linked data using statistical distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):63–86.

Rossi, A., Barbosa, D., Firmani, D., Matinata, A., and Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49.

Ruiz-Casado, M., Alfonseca, E., and Castells, P. (2007). Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia. *Data & Knowledge Engineering*, 61(3):484–499.

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, 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.

Wang, M., Qiu, L., and Wang, X. (2021). A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13(3):485.

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*.