Expansion of Hindi WordNet using Knowledge Graph Completion Methods: A Survey

Independent Study

Sushil Awale

01.02.2022

Abstract

WordNet is a valuable resource with wide use-cases in both industry and academic settings. With the introduction of the English WordNet in 1995, WordNets for various languages have been developed over the years. However, these WordNets require frequent update as new words and facts are introduced and suffer from the problem of incompleteness. Link prediction (LP) is a widely known task in the field to counter the incompleteness problem of Knowledge Graphs (KG). In this work, we look at the expansion of the Hindi WordNet as a link prediction task. In this survey paper, we introduce the Hindi WordNet dataset as a KG accounting for test leakage in the dataset. In addition, we also look at various embedding-based Knowledge Graph Completion (KGC) models and evaluate the performance of these models on the dataset. For the evaluation, we look at both overall and relation-category specific performance.

1 Introduction

Arrangement of words on the basis of semantic concepts rather than the traditional alphabetical way enables numerous applications. Such large collection of words is called a WordNet which has use cases in various Natural Language Processing tasks such as word-sense disambiguation, information retrieval, machine translation and more. With the first WordNet created for the English language in 1985 at the Princeton University [4], WordNet for various languages have been developed over the years. As languages grow and evolve, WordNets also require revision which is an expensive and time-consuming task. WordNet for high-resource languages are updated from time to time, while low-resource languages are being ignored.

In this survey paper, we focus on the expansion of WordNet, especially for a low-resource language such as Hindi. WordNet [5] is 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. In order to expand the WordNet, we can add new relations between the entities, or add new entities to an existing relation-entity pair. The latter approach is known as link prediction and is widely researched as a knowledge graph completion task [7].

The research on knowledge graph completion task has largely focused on English WordNet but to our knowledge no research work has been focused on the Hindi WordNet. This survey paper is the first to study the KGC task on the Hindi WordNet.

The main contributions of this paper are as follows:

- Introduction of an RDF-style triple Hindi WordNet graph database suitable for KGC task
- Survey of different KGC models and its performance on the introduced Hindi WordNet graph database

2 Related Work

Two of the popular knowledge graph datasets derived from WordNet are WN18 and WN18RR [2]. WN18 is the KG dataset extracted from the English WordNet which was introduced in the TransE paper [1] . The dataset was built by iterative filtering out of entities and relationships with too few entries. WN18RR is a subset of the WN18 dataset introduced in ConvE paper. In this dataset, in order to counter test leakage in WN18, the inverse relations are removed. Furthermore, the identical relations in the test and valid set that are linked with the train set are also removed. We follow this approach in the creation of the Hindi WordNet dataset.

Knowledge Graph Completion is widely researched topic in the natural language processing domain, and in along with research works in the development of various KGC models, different survey works are also published. Rossi et. al. [7] compare different KG embeddings model based on effectiveness and efficiency. In this survey paper, the KG models are grouped into three categories by their learning methods; tensor decomposition models, geometric models, and deep learning models. In this study, the efficiency of the KG models on different popular English KG datasets are studied in terms of training time and prediction time, and the effectiveness of the models in terms of structure of the training graph. Wang et. al. [10] provides a theoretical analysis of the different KG models, and classify the KG models into three-main categories based on the type of scoring function used, e.g. distance-based or semantic-matching-based. The paper also compares the performance of the models on two popular English KG datasets, WN18RR and FB15K-237. In our paper, we follow the classification presented in this paper to select at most two KG models from each class for our study.

Our work is different from these surveys in that we study the performance of the KG models for a new dataset, Hindi WordNet.

3 Background

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.

3.2 Evaluation

Evaluation for the link prediction task is carried out based on the rank position of the correct entity in the list of ranked entities. For each test triple from the test set, the model replaces the missing entity with all the entities from the knowledge graph set and ranks them based on a score using the scoring function. The position of the correct entity r is taken into account and used to calculate the following metrics.

Mean Rank (MR) is the average of the obtained ranks.

$$MR = \frac{1}{|R|} \sum_{r \in R} r \tag{1}$$

Mean Reciprocal Rank (MRR) is the average of the inverse of the obtained ranks.

$$MRR = \frac{1}{|R|} \sum_{r \in R} \frac{1}{r} \tag{2}$$

Hits@K (H@K) is the ratio of predictions for which the rank is equal or lesser than a threshold K

$$H@K = \frac{|r \in R : r \leq K|}{|R|} \tag{3}$$

The common values used for K are 1, 3, 5 and 10.

The evaluation can be carried out in two different settings, *raw* and *filtered*. When predicting the missing entity, the predicted entity may not be targeted entity, but still be valid. A triple is considered valid, if it exists in the dataset. If the valid entity ranks higher than the target entity, but it is considered a mistake and does not account when computing the rank, then this scenario is considered *raw scenario*. On the other hand, if the valid entity ranks higher than the correct entity, and it is considered no considered a mistake and considered when computing the rank, then the scenario is called *filtered scenario*. [7]

4 Translation-distance-based models

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

4.1 TransE

TransE [1] is one of the first and simple translation-distance-based model. Given a triple (h, r, t) where h, $e \in E$ (set of entities) and $r \in R$ (set of relationships), TransE learns vector embeddings for h, t and r such that distance between $h + r$ and t is minimum. In TransE, $L1$ or $L2$ norm is used to measure the distance, $d(h, r, t) = ||h + r - t||$. To learn the embeddings, the following loss-function is minimized over the training set:

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

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

TransE model is known to struggle with one-to-many/many-to-one/many-to-many relations.

4.2 TransH

TransH [11] overcomes the problems of TransE in modeling of one-to-many/many-to-one/many-to-many relations by interpreting a relation as a translation operation on a hyperplane. It introduces two vectors for a relation r, a relation-specific-specific translation vector d_r and a relation-specific hyperplane w_r. Then the embedding vectors of head h and tail t are projected to the hyperplane which gives new vectors $h\perp$ and $t\perp$ respectively. Then the scoring function to measure the plausibility of a triple is defined as $f_r(h, t) = ||h\perp + d_r - t\perp||$. When $||w_r||_2 = 1$ is restricted, we get,

$$h\perp = h - w_r^\top h w_r, \qquad t\perp = t - w_r^\top t w_r$$

Then, we have the scoring function as,

$$f_r(h, t) = ||(h - w_r^\top h w_r) + d_r - (t - w_r^\top t w_r)|| \tag{5}$$

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

$$L = \sum_{(h,r',t)} \sum_{(h',r,t')} [\gamma + f_r(h, t) - f_r(h', t')]_+ \tag{6}$$

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

5 Semantic-matching-based models

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

5.1 DistMult

DistMult [12] is a semantic-matching-based multiplicative model in which the relationship vector is enforced to be a diagonal matrix. DistMult uses neural networks with energy-based objectives to learned the representations.

The head entity h and tail entity t are initialized as either a "one-hot" vector or an "n-hot" feature vector. Then the learned representations, $y_h \in R$ and $y_t \in R$ are given by,

$$y_h = f(Wh), \qquad y_t = f(Wt)$$

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 previous discussed models, is represented in the form of scoring function. In DistMult, the function is formulated as bilinear,

$$S(y_h, y_t) = y_h^T M_r y_t$$

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

$$L = \sum_{(h,r,t)} \sum_{(h',r,t')} max\{S_{(h',r,t')} - S_{(h,r,t)} + 1, 0\} \tag{7}$$

5.2 ComplEx

ComplEx [9] 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 bilinear product becomes a Hermitian product.

In ComplEx, the set of entities is represented as ϵ 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

$$\begin{aligned}\phi(r,h,t;\theta) =& < Re(w_r), Re(h), Re(t) > + < Re(w_r), Im(h), Im(t) > + \\ & < Im(w_r), Re(h), Im(t) > - < Im(w_r), Im(h), Re(t) >\end{aligned} \tag{8}$$

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.

6 Neural Network based models

6.1 ConvE

ConvE [2] 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 w of $m \times n$ filters. The ouput of the convolution layer is then fed into a dense layer with a single neuron and weights W, giving out a fact score. In ConvE, the scoring function is defined by a convolution over the embeddings as follows:

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

where r is a relation parameter, $\bar{h}$ and $\bar{t}$ dentoe 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.

7 Experiment

7.1 Hindi WordNet

For our study, we take the Hindi WordNet developed at Center For Indian Languages Technology [6]. 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

Table 1: Count of Synsets by Relation Type

Relation	Synset count
ONTO_NODES	44,857
HYPERNYM	33,972
HYPONYM	30,836
MODIFIES_NOUN	9,780
HYPONYM	1,814

7.2 Dataset

For the dataset, we convert the WordNet into RDF-style triples graph fitting for the task of Knowledge Graph Completion. 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.

As noted by [8], the training dataset of WN18 has 94% test leakage i.e. triples in the test set have inverse relations that are linked to the train set. Dettmers et. al. [2] show the severity of the problem by building a rule-based inverse model that easily learn these inverse mappings. Hence, following Dettmers et. al. [2], we remove these inverse relations from our dataset to correctly evaluate the performance of models. For this, 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 of antonym relations under the relation 'antonym'. The statistics of the dataset is shown in Table 2.

Table 2: Statistics of Dataset

Entity	Relation	Triple	Train	Valid	Test
39,609	16	95,838	86,432	4,712	4,694

7.3 Setup

We run the TransE, TransH, DistMult and ComplEx models using OpenKE toolkit [3]. We run all experiments in the their default setting. For ConvE, we run the model published in GitHub from the authors. All these models were run on a Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64) server with NVIDIA GeForce RTX 2080 Ti GPU, and 256 GB RAM.

8 Results and Analysis

For the model performance, we report the results obtained in filtered evaluation scenario for the metrics MR, MRR, $Hit@1$, $Hit@3$ and $Hit@10$. All the models were evaluated on the Hindi WordNet dataset.

Table 3: MR, and MRR of the models

Model	MR	MRR
TransE	**3125**	0.156
TransH	4123	0.133
DistMult	3510	0.166
ComplEx	4119	0.172
ConvE	3405	**0.294**

For Mean Rank, the lower the score the better it is. In Table 3, we observe that the TransE model performs the best. However, MR is sensitive to outliers and MRR is used to counter this. The ConvE model perform the best in the MRR score.

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, where as at $K = 3$ and $K = 1$, the ConvE models outperforms all the other models significantly.

Table 4: Hit@1, Hit@3, Hit@10 of the models

Model	Hit@1	Hit@3	Hit@10
TransE	0.055	0.221	0.334
TransH	0.032	0.199	0.308
DistMult	0.119	0.19	0.24
ComplEx	0.13	0.188	0.237
ConvE	**0.24**	**0.316**	**0.385**

We further test the model on the subsets of the test data. These subsets correspond to the different categories of the triples which include $1-1$, $n-1$ and $n-n$ relations. The model performances for each category are shown in tables 5, 6, 7, with the best score marked in bold face.

In our results, we observe that the ConvE model outperforms the transition-distance-based models and the semantic-distance-based models in MRR and $Hit@10$ metrics. The ConvE model achieves a stable score across all relation-types signaling better generalization ability of the model.

Table 5: MR, MRR, and Hit@10 of the models for 1-1 relations

Model	MR	MRR	Hit@10
TransE	5509	0.036	0.089
TransH	6659	0.03	0.0874
DistMult	8582	0.007	0.009
ComplEx	8389	0.015	0.0336
ConvE	**3462**	**0.29**	**0.384**

Table 6: MR, MRR, and Hit@10 of the models for n-1 relations

Model	MR	MRR	Hit@10
TransE	**3249**	0.168	0.34
TransH	4219	0.137	0.31
DistMult	3974	0.16	0.24
ComplEx	4046	0.18	0.25
ConvE	3505	**0.294**	**0.389**

Table 7: MR, MRR, and Hit@10 of the models for n-n relations

Model	MR	MRR	Hit@10
TransE	**1334**	0.114	0.344
TransH	1490	0.116	0.353
DistMult	2459	0.103	0.262
ComplEx	4963	0.154	0.212
ConvE	3477	**0.293**	**0.388**

9 Conclusion

In this survey paper, we look at the expansion of Hindi WordNet as a link prediction task under the problem of Knowledge Graph Completion. Most of the work carried out under this research problem focus on English Knowledge Graph datasets, such as WN18 and WN18RR. In our work, we introduce a new Hindi WordNet as a Knowledge Graph dataset. The dataset design is inspired by the WN18RR dataset which accounts for test leakage in the dataset. In addition, we evaluate the different Knowledge Graph Completion models developed in the literature and apply it to the newly created dataset. We report the performance of these models on the dataset with respect to both overall and relation-category specific performance.

References

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[2] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[3] Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*, 2018.

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

[5] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[6] Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande, and Pushpak Bhattacharyya. An experience in building the indo wordnet-a wordnet for hindi. In *First International Conference on Global WordNet, Mysore, India*, volume 24, 2002.

[7] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49, 2021.

[8] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509, 2015.

[9] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[10] Meihong Wang, Linling Qiu, and Xiaoli Wang. A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13(3):485, 2021.

[11] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[12] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.