TaxoComplete: Self-Supervised Taxonomy Completion Leveraging Position-Enhanced Semantic Matching

Ines Arous
University of Fribourg
Fribourg, Switzerland
ines@exascale.info

Ljiljana Dolamic
armasuisse S+T
Thun, Switzerland
ljiljana.dolamic@armasuisse.ch

Philippe Cudré-Mauroux
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

ABSTRACT

Taxonomies are used to organize knowledge in many applications, including recommender systems, content browsing, or web search. With the emergence of new concepts, static taxonomies become obsolete as they fail to capture up-to-date knowledge. Several approaches have been proposed to address the problem of maintaining taxonomies automatically. These approaches typically rely on a limited set of neighbors to represent a given node in the taxonomy. However, considering distant nodes could improve the representation of some portions of the taxonomy, especially for those nodes situated in the periphery or in sparse regions of the taxonomy.

In this work, we propose TaxoComplete, a self-supervised taxonomy completion framework that learns the representation of nodes leveraging their position in the taxonomy. TaxoComplete uses a self-supervision generation process that selects some nodes and associates each of them with an anchor set, which is a set composed of nodes in the close and distant neighborhood of the selected node. Using self-supervision data, TaxoComplete learns a position-enhanced node representation using two components: (1) a query-anchor semantic matching mechanism, which encodes pairs of nodes and matches their semantic distance to their graph distance, such that nodes that are close in the taxonomy are placed closely in the shared embedding space while distant nodes are placed further apart; (2) a direction-aware propagation module, which embeds the direction of edges in node representation, such that we discriminate <node, parent> relation from other taxonomic relations. Our approach allows the representation of nodes to encapsulate information from a large neighborhood while being aware of the distance separating pairs of nodes in the taxonomy. Extensive experiments on four real-world and large-scale datasets show that TaxoComplete is substantially more effective than state-of-the-art methods (2x more effective in terms of HIT@k).

CCS CONCEPTS

• **Information systems** → **Information extraction**.

KEYWORDS

Taxonomy Completion, Self-Supervision, Semantic Matching

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583342

ACM Reference Format:
Ines Arous, Ljiljana Dolamic, and Philippe Cudré-Mauroux. 2023. TaxoComplete: Self-Supervised Taxonomy Completion Leveraging Position-Enhanced Semantic Matching. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3543507.3583342

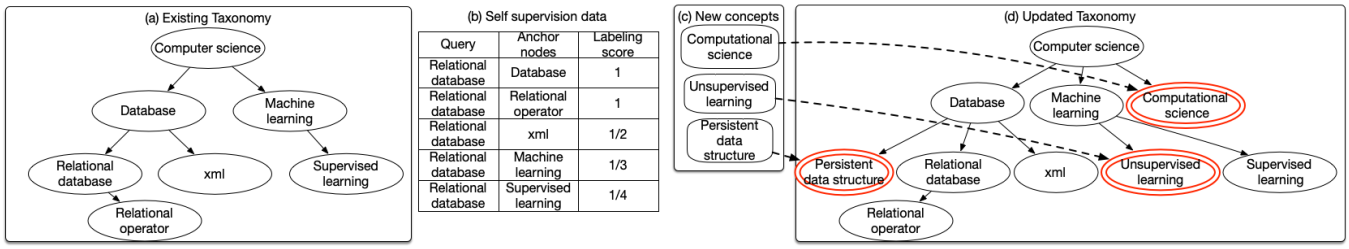
1 INTRODUCTION

Taxonomies are widely used to organize concepts in a hierarchical structure through "is-a" relations between concepts. They have a wide range of applications, including product search [20] and recommendation [23] in the e-commerce domain, articles classification and clustering [24] in the scientific domain, and content browsing [19] in the web search domain. Many platforms such as Amazon [1], Google [2] and Bing [3] rely on curated taxonomies to enhance search quality, user profiling, ads targeting, and content categorization. Most of these taxonomies are manually curated by domain experts, making them highly valuable resources.

With the emergence of new concepts, existing taxonomies slowly become obsolete. Therefore, it becomes crucial to maintain them such that they remain relevant. A straightforward approach towards that end consists in relying on human experts to update existing taxonomies. Pinterest, for example, reported on their use of eight curators to append 5,000 new nodes to their taxonomy over a period of two months [3]. This updated taxonomy measurably increased revenue gains, thanks to the improvement it led in targeted ads. Nonetheless, this manual curation process was proven to be prohibitively expensive and slow, as it cannot keep up with the millions of new content items created by Pinterest users daily. Due to the rapid growth in content creation on many platforms, updating existing taxonomies dynamically becomes an ever more pressing challenge [10].

Several strategies have been developed to update taxonomies dynamically. Overall they are based on two key modules: a propagation module and a matching module. The propagation module represents nodes in the existing taxonomy by aggregating information surrounding a given node, for instance from its local neighborhood [15], local mini-paths [21], its ancestral and descendant paths [5], or paths from the root to leaf nodes [8]. The matching module learns how to identify the best parent for the given node using the node representation obtained from the propagation module. Existing methods typically map this problem to a binary classification task and predict whether a new query node is the direct child of a parent node. To that end, existing methods either rely on a classifier [8, 15] or combine multiple scoring functions [5, 22].

[1] www.amazonlistingservice.com/blog/amazon-store-taxonomy-organization
[2] www.google.com/basepages/producttype/taxonomy.en-US.txt
[3] help.ads.microsoft.com/#apex/ads/en/51112/1



Query	Anchor nodes	Labeling score
Relational database	Database	1
Relational database	Relational operator	1
Relational database	xml	1/2
Relational database	Machine learning	1/3
Relational database	Supervised learning	1/4

Figure 1: An example of a completion task for a computer science taxonomy. The left figures illustrate: (a) the existing taxonomy of computer science; (b) the self-supervision data generated from the existing taxonomy by extracting a query node "relational database" and its anchor set composed of nodes from its close neighborhood such as "database" and from its distant neighborhood such as "supervised learning" and labeling each pair with the inverse of the graph distance separating them; the right figure illustrates: (c) new concepts that are added to (d) the updated taxonomy.

These methods suffer from several limitations. First, the propagation module often consists of a variant of graph convolutional networks, which only use the information of a limited neighborhood for each node [1]. Second, they aggregate the structural information collected from the chosen neighborhood, which leads to losing fine-grained information that can be useful for learning the semantic distance between node pairs and their connection with the rest of the taxonomy. Third, using a binary classifier as the matching module to predict if a pair of nodes have a child-parent relation or not raises the issue of lexical memorization [7, 11]: instead of learning the child-parent relation, the matching module might mistakenly learn the existence of a different taxonomic relation such as <node, sibling> relation.

In this paper, we introduce TaxoComplete, a novel self-supervised framework for the taxonomy completion task, where given a new node and an existing taxonomy, we aim to identify the position of the new node in the existing taxonomy. Our framework automatically constructs candidate node pairs from an existing taxonomy and uses them to learn a position-enhanced semantic matching model for completing the taxonomy. Specifically, we sample some concepts in the existing taxonomy and view them as query nodes. Then, we associate each of them with an *anchor set*, i.e., a set of nodes in the close and distant neighborhood of a query node. The close neighborhood of a query node includes nodes representing its parent, siblings, and children, which helps define the query node's exact position. The distant neighborhood includes nodes randomly sampled from the existing taxonomy, which helps locate nodes with a small close neighborhood such as nodes situated in the periphery or sparse regions of the taxonomy. Each pair of <query node, anchor node> is labeled with a score proportional to the graph distance, i.e., the length of the shortest path separating them in the taxonomy. For instance, in Figure 1, the graph distance separating the <Relational database, Machine learning> pair is 3; therefore, we label this pair with a score of 1/3. Unlike existing methods which consider only positive and negative pairs, our labeling method allows us to capture fine-grained information.

To make the best use of the above self-supervision data, our framework semantically matches nodes in the taxonomy by utilizing two modules: a query-anchor semantic matching mechanism and a direction-aware propagation module. First, the query-anchor semantic matching mechanism matches the semantic distance between two nodes to the graph distance separating them in the taxonomy. To that end, TaxoComplete inherits from recent advances in the semantic search domain and adopts bi-encoders [13, 17] to efficiently compare node pairs. With bi-encoders, we encode pairs of nodes using their definition from a supporting corpus such that they can be compared with cosine-similarity. Then, we optimize model learning such that the semantic distance separating the definitions of a pair of nodes matches the distance separating them in the taxonomy. Our query-anchor semantic matching mechanism learns the taxonomic position of nodes but not edges direction, which limits its ability to discriminate <node, parent> relation from other types of taxonomic relations. To enhance the node representations obtained from the bi-encoders with the edges' direction, we design a direction-aware propagation module. This module weights the node representations with the significance of other nodes directed to it. Through our query-anchor semantic matching mechanism and direction-aware propagation module, we preserve fine-grained information and obtain node representations that faithfully captures the taxonomic structure.

We conduct extensive experiments on four real-world datasets to compare TaxoComplete performance to state-of-the-art methods. In addition, we design several variants of our framework to conduct an ablation study and shed additional light on the performance of our approach and its various sub-modules. Our results show that TaxoComplete can accurately predict the correct position of a query node. To the best of our knowledge, we are the first to enhance semantic matching with the node position for the taxonomy completion problem.

In summary, we make the following key contributions:

- We develop TaxoComplete – a self-supervised taxonomy completion framework leveraging position-enhanced semantic matching;
- We propose a query-anchor semantic matching mechanism that encodes pairs of nodes and matches their semantic distance to their graph distance, which is subsequently enhanced with a direction-aware propagation module;
- We conduct an extensive empirical evaluation of our method on four real-world datasets and show that TaxoComplete improves the state of the art by a considerable margin (2x performance improvement in terms of HIT@k).

2 RELATED WORK

In this section, we discuss the state of the art in dynamic maintenance of taxonomies, before methodologically reviewing related work in semantic similarity search.

2.1 Taxonomy Expansion and Completion

Updating taxonomies when new concepts emerge is a laborious and expensive process. Several methods have been developed to tackle the taxonomy maintenance problem automatically. We distinguish between two tasks for maintaining taxonomies: expansion and completion. In the taxonomy expansion task, new concepts are added as leaf nodes. To this end, several methods have been developed [11, 12, 15]. TaxoExpan [15], for example, learns a position-enhanced graph neural network to predict the relative position of nodes in the taxonomy and uses the InfoNCE [12] loss for training their model. Manzoor et al. [11] tackle the expansion problem for taxonomies with heterogeneous edge semantics and learn latent representations of both edges and nodes to measure the relatedness between nodes. Yu et al. [21] propose to map the problem to a mini-path prediction task where they use a multi-view co-training objective to match a query with a mini-path sampled from the existing taxonomy. A similar approach is adopted by TEMP [8], which predicts the position of new concepts by ranking the sub-paths from the taxonomy. In the taxonomy completion task, new concepts are added at any level of the taxonomy. Among the first methods developed for the completion task is TMN [22], which reformulates the problem to a one-to-pair matching problem and uses different scoring functions to estimate a matching score between a query node and a pair of hypernym and hyponym concepts. An extension of the TMN approach was developed by Jiang et al. [5], where the authors consider the siblings of a query node by incorporating horizontal structural information in the taxonomy in addition to encoding candidate pairs.

Most of the existing methods train their models with positive and negative candidates, disregarding fine-grained information such as the graph distance between pairs of nodes. As a result, they often mistakenly predict the existence of a taxonomic relationship instead of the targeted parent-child relation. Compared to these methods, our approach adopts a position-enhanced semantic matching framework where we incorporate the graph distance separating pairs of nodes into the learning of the node representation. Our method effectively captures the position of the nodes in the taxonomy as well as the overall structure of the taxonomy.

2.2 Semantic Similarity Search

In taxonomy completion, nodes in the taxonomy are often associated with a definition from a supporting corpus. To find the adequate parent of a new node in the existing taxonomy, one typically searches through the definitions of all nodes in the existing taxonomy. This problem is akin to a semantic similarity search problem, which aims to identify the most similar items in a large-scale corpus. Recently, much work has been dedicated to addressing the semantic similarity search problem in document collections, especially for the task of open-domain question answering. Many state-of-the-art methods in semantic similarity search retrieve relevant documents using a bi-encoder to encode the query document and a document from the corpus, respectively, and estimates their relevance by computing a single similarity score between two representations [25]. For instance, Karpukhin et al. develop a dense passage retriever [6] which uses a dense encoder to map a textual document to a d-dimensional vector and then measures the similarity between two documents with the dot product. TwinBERT [9] uses bi-encoders to represent a query and a document separately, then combines the vector outputs of encoders with a crossing layer to measure the relevance between the two. Similarly, bi-encoders were used to derive semantically meaningful sentence embeddings that can be compared with cosine-similarity [2, 13].

In our work, we draw inspiration from these developments and leverage a bi-encoder to measure the similarity between node pairs. Furthermore, we enhance their representation to reflect their graph distance in the taxonomy. The idea of using semantic similarity measures in taxonomies has been studied for a wide range of applications [18, 26]. To the best of our knowledge, however, we are the first to propose a new semantic similarity method leveraging bi-encoders to tackle the taxonomy completion problem effectively.

3 TAXOCOMPLETE FRAMEWORK

In this section, we introduce our position-enhanced semantic matching framework that learns to predict the best position of a query node in a given taxonomy. We first formally define our problem and then discuss how we generate self-supervision data from the existing taxonomy. We then describe the modules of our framework followed by our algorithm for taxonomy completion.

3.1 Problem Formulation

In this section, we formally define the concept of taxonomy and then formulate our problem.

Definition 3.1 (Taxonomy). A taxonomy $\mathcal{T}$ is a directed acyclic graph with nodes $\mathcal{N}$ connected with edges $\mathcal{E}$. Each node $n \in \mathcal{N}$ represents a concept associated with its definition d_n from a supporting corpus $\mathcal{D}$. Each directed edge $< n_p, n_c >$ denotes a relation where n_p is a more general concept than n_c. We refer to n_p as a parent node and n_c as a child node.

Definition 3.2 (Problem Definition). Let $\mathcal{T}_0 = (\mathcal{N}_0, \mathcal{E}_0)$ be a seed taxonomy where $\mathcal{N}_0$ and $\mathcal{E}_0$ are the nodes and the edges in $\mathcal{T}_0$, respectively. Let C be a set of new terms and $\mathcal{D}$ a corpus defining all nodes in the existing taxonomy $\mathcal{T}_0$ and the new terms C. Our goal is to complete the taxonomy $\mathcal{T}_0$ and construct its updated version $\mathcal{T}$ such that $\mathcal{T} = \mathcal{T}_0 \cup C$. Specifically, we aim to represent nodes in the embedding space such that their semantic distance matches the distance separating them in the taxonomy. Formally, given a pair of nodes q and a represented with vectors w_q and w_a respectively, our goal is to approximate their graph distance with their cosine similarity, i.e., we aim to minimize the following function:

$$cos(w_q, w_a) - distance(q, a) \tag{1}$$

3.2 Self-Supervised Learning with Position-Enhanced Semantic Matching

3.2.1 Self-supervision Generation. We create query nodes and their respective anchor sets from the seed taxonomy to train our taxonomy completion model. We proceed by randomly sampling query

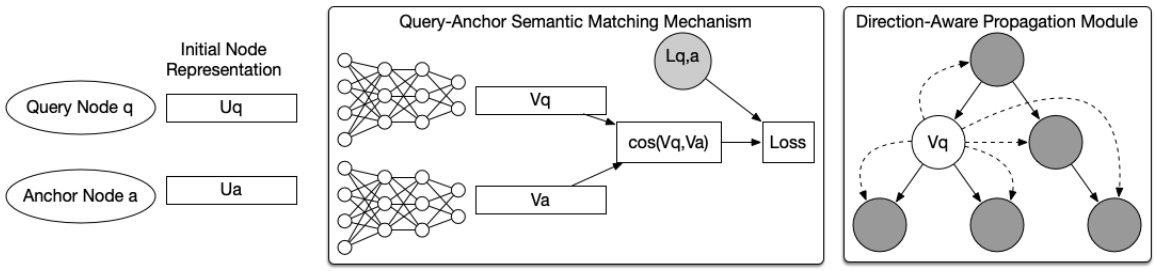


Figure 2: Overview of the TaxoComplete framework. From left to right: we initialize the node representation of a query node and a node from its anchor set. Then, we learn the semantic relatedness of nodes with our query-anchor semantic matching mechanism. Finally we inject edges' direction using a direction-aware propagation module.

nodes from the seed taxonomy. Then, for each query node q, we construct its anchor set composed of a close and a distant neighborhood:

- **close neighborhood**: For each query node, we select its siblings, its ancestral nodes, and its children nodes denoted respectively $\mathcal{S}_q$, $\mathcal{A}_q$ and $\mathcal{C}_q$. Sibling nodes $\mathcal{S}_q$ are nodes that share the same parent with a query node q. Ancestral nodes $\mathcal{A}_q$ identify all nodes in the path connecting the root node to the query node q, while children nodes $\mathcal{C}_q$ are the children of q. Overall, the close neighborhood $\mathcal{B}_{c,q}$ of a query node q is composed of all siblings $\mathcal{S}_q$, ancestral nodes $\mathcal{A}_q$, and children nodes $\mathcal{C}_q$, i.e., $\mathcal{B}_{c,q} = \mathcal{S}_q \cup \mathcal{A}_q \cup \mathcal{C}_q$.
- **distant neighborhood**: We randomly select a set of nodes from the seed taxonomy that are not in the close neighborhood of the query node q. We choose the size of the distant neighborhood $\mathcal{B}_{d,q}$ to be larger than the size of the close neighborhood by a sampling rate s_r, i.e., $|\mathcal{B}_{d,q}| = s_r \times \mathcal{B}_{c,q}$. We empirically study the impact of the distant neighborhood size on the discriminative capabilities of TaxoComplete in Section 4.4.2.

We associate a pair of <query node, anchor node> with the following labeling function:

$$l_{q,a} = \frac{1}{f(d_{q,a})}, \tag{2}$$

where $d_{q,a}$ is the graph distance separating a query node q from an anchor node a in the close or the distant neighborhood and $f(.)$ is a linear function. To measure the distance between a query node and an anchor node, we omit the direction in the taxonomy and consider it as an undirected graph. We empirically compare the impact of including and omitting the direction in the labeling function in Section 4.4.1 and found that omitting the direction led to better results.

3.2.2 Query-Anchor Semantic Matching Mechanism. Given a query node q with its definition d_q and a node from its anchor set a with its definition d_a, we use a bi-encoder to generate their respective representations u_q and u_a. This bi-encoder is a twin network with shared Transformer weights [4, 13]. Each network in the bi-encoder is composed of a pre-trained Transformer model E (e.g., distllBERT [14]) that maps the nodes' definitions to an initial node representation u_q and u_a, then a pooling layer is applied on each vector to represent the query node v_q and the anchor node v_a. The encodings v_q and v_a are formulated as follows:

$$v_q = Pooling(E(d_q, \theta)), v_a = Pooling(E(d_a, \theta)) \tag{3}$$

We fine-tune the bi-encoder and update the model parameters θ such that nodes that are close in the taxonomy are placed closely in the shared embedding space while unrelated nodes are placed further apart. To that end, we use a regression loss function, specifically the mean-squared error loss function, to approximate the cosine similarity between the two node encodings v_q and v_a to our labeling function.

$$\mathbf{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(cos(v_q, v_a) - l_{q,a})^2 \tag{4}$$

At inference time, nodes from the seed taxonomy with the highest similarity score with a new query node can be efficiently retrieved using our semantic matching mechanism. However, it does not consider edges' direction in node representations, which limits its ability to identify <node, parent> relation. To mitigate this problem, we design a direction-aware propagation module.

3.2.3 Direction-Aware Propagation Module. We inject the direction of the edges using a direction-aware propagation module in order to enhance the node representation. To do so, we propagate the node features with a variation of personalized PageRank, namely the personalized propagation of neural predictions [1]. Formally, we associate the taxonomy $\mathcal{T}_0$ with its adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$, where $|\mathcal{N}|$ is the number of nodes in $\mathcal{T}_0$. We define the symmetrically normalized adjacency matrix as follows:

$$\hat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}} \tag{5}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_{n,n}$ is the adjacency matrix with added self-loops, $\mathbf{I}_{|\mathcal{N}|}$ is the identity matrix and $\widetilde{\mathbf{D}} = \sum_{j\in\mathcal{N}}\widetilde{\mathbf{A}}_{i,j}$ is the graph degree matrix with the addition of self-loops. The position-enhanced node representation is given by:

$$w_q = \alpha(\mathbf{I}_{|\mathcal{N}|} - (1-\alpha)\hat{\mathbf{A}})^{-1}v_q \tag{6}$$

where α is a propagation factor, empirically defined between 0 and 1. The resulting vector w_q can be viewed as the node representation vector v_q weighted by the significance of other nodes directed to the query node q.

Calculating the term $(\mathbf{I}_{|\mathcal{N}|} - (1-\alpha)\hat{\mathbf{A}})^{-1}$ requires to construct an $|\mathcal{N}| \times |\mathcal{N}|$ matrix, which is computationally intensive in large

Algorithm 1: Learning TaxoComplete Parameters

Input : Seed taxonomy $\mathcal{T}_0$, adjacency matrix $\mathbf{A}$
Output : Nodes representation w_n, updated parameters θ

1 Q = sample($\mathcal{T}_0$)
2 # self-supervision generation
3 **for** q *in* Q **do**
4 $\mathcal{B}_{c,q}$ = close_neighborhood(q);
5 $\mathcal{B}_{d,q}$ = distant_neighborhood(q);
6 # Query-Anchor Semantic Matching Mechanism
7 **while** $\mathbf{L}(\theta)$ *has not converged* **do**
8 **for** q *in* Q **do**
9 $v_q = Pooling(E(d_q, \theta))$;
10 $v_a = Pooling(E(d_a, \theta))$, with $a \in \mathcal{B}_{c,q} \cup \mathcal{B}_{d,q}$;
11 Update θ based on Q
12 # Direction Aware Propagation module
13 w_n = propagate(v_n, $\mathbf{A}$), with n in $\mathcal{T}_0$;

taxonomies. An approximation of Eq.(6) is calculated as follows [1]:

$$w_q^0 = v_q$$
$$w_q^k = (1-\alpha)\hat{\mathbf{A}} w_q^{(k-1)} + \alpha v_q \quad (7)$$

The computation of v_q^k is repeated K times until convergence where k denotes the iteration number. Using this approximation, we do not construct a full $\mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$ matrix, which allows for efficient inference.

3.3 Algorithm

Our overall training algorithm is given in Algorithm 1. Given a seed taxonomy $\mathcal{T}_0$ and the adjacency matrix $\mathbf{A}$, TaxoComplete first generates a set of self-supervision data (rows 1-5), then learns node representations with a query-anchor semantic matching mechanism (rows 7-11) and finally injects the direction of the edges with a direction-aware propagation module (row 13). In rows 1-5, query nodes from the seed taxonomy are sampled (row 1), and close and distant neighborhoods are built for each query node (rows 3-5). In rows 7-11, the parameters θ are learned by minimizing the loss function from Eq. (4). Finally, we inject the direction information into the node representation in row 13 using Eq. (7).

4 EXPERIMENTS

In this section, we present the results of our empirical evaluation[4]. We first discuss our experimental setup, then evaluate the performance of TaxoComplete by comparing it with baseline methods. Finally, we perform an in-depth analysis of TaxoComplete 's main properties. We aim at answering the following questions:

- Q1: How effective is our framework in identifying the correct position of a query node? (Section 4.2).
- Q2: What is the impact of injecting the direction of the edges into our node representation? (Section 4.3).
- Q3: What is the impact of the labeling function on the performance of our method and how effective is our sampling strategy in capturing the position of the nodes in the taxonomy? (Section 4.4).

[4]Source code and data are available at https://github.com/eXascaleInfolab/TaxoComplete

Table 1: Description of the taxonomy Datasets.

Dataset	#nodes	#edges	Depth
SemEval-Noun	75,359	76,810	20
SemEval-Verb	13,715	13,407	13
MAG-WIKI-CS	25,170	40,314	6
MAG-WIKI-PSY	10,671	14,080	6

4.1 Experimental Setup

4.1.1 Datasets. We evaluate the performance of TaxoComplete using the following datasets:

- SemEval: This dataset is based on WordNet 3.0[5], which contains verbs, nouns and the relations among them. We derive two datasets from it, which we refer to as **SemEval-Noun** and **SemEval-Verb**, respectively.
- MAG-WIKI: This dataset is derived from the original Microsoft Academic Graph [16] dataset. MAG is a heterogeneous graph containing over 660k fields of study and 700k taxonomic relations. For each concept in the MAG, we extract its definition from Wikipedia. Following previous studies [15, 22], we construct two datasets **MAG-CS-WIKI** and **MAG-PSY-WIKI**, which are respectively based on the subgraphs for computer science and psychology.

These taxonomies interlink concepts, not just words. As a result, words that have similar meanings are semantically disambiguated. Each node in the taxonomy is associated with a definition from a supporting corpus. We use the SemEval corpus and Wikipedia to define nodes respectively in the SemEval and MAG taxonomies. Both taxonomies are manually-verified and are often used for taxonomy expansion and completion tasks. Key statistics on the collected dataset are reported in Table 1.

4.1.2 Evaluation Metrics. Our method generates a list of candidate positions for each query node in the test set. To evaluate its performance, we use the following ranking metrics:

- Mean Rank (MR) calculates the mean rank position of a query concept's true parent among all candidates. Smaller MR value indicates better model performance.
- HIT@k is the number of query concepts' true positions ranked in the top-k, divided by k.

4.1.3 Baseline Methods. We compare our approach with the following state-of-the-art techniques:

- Arborist [11]: is used for taxonomy expansion. It takes into account heterogeneous edge semantics and optimizes a large-margin ranking loss with a dynamic margin function.
- TaxoExpan [15]: uses position-enhanced graph neural networks to capture local information. This methods was developed for the taxonomy expansion task.

[5]https://wordnet.princeton.edu/

Table 2: Performance (MR and Hit@k) comparison of taxonomy completion techniques on four datasets. The best performance is highlighted in bold; the second best performance is marked by '*'. We run all methods five times with different seeds and report the average result with standard deviation.

Method	SemEval-Noun				SemEval-Verb			
	MR	HIT@1	HIT@5	HIT@10	MR	HIT@1	HIT@5	HIT@10
TaxoExpan	1236.4 ± 465*	0.069 ± 0.005	0.172 ± 0.023	0.248 ± 0.035	876.1 ± 123*	0.072 ± 0.010	0.186 ± 0.021	0.251 ± 0.021*
TMN	2237.4 ± 1087	0.036 ± 0.006	0.112 ± 0.009	0.174 ± 0.016	1931.9 ± 525	0.063 ± 0.007	0.160 ± 0.020	0.224 ± 0.026
Arborist	3993.1 ± 1295	0.020 ± 0.003	0.076 ± 0.009	0.122 ± 0.015	1878.8 ± 329	0.032 ± 0.005	0.100 ± 0.013	0.159 ± 0.018
TaxoEnrich	1703.5 ± 319	0.094 ± 0.015*	0.229 ± 0.033*	0.312 ± 0.038*	2762.0 ± 679	0.087 ± 0.027*	0.188 ± 0.046*	0.240 ± 0.063
TaxoComplete	**474.4 ± 57**	**0.176 ± 0.008**	**0.427 ± 0.009**	**0.541 ± 0.008**	**589.3 ± 132**	**0.123 ± 0.010**	**0.316 ± 0.016**	**0.421 ± 0.028**
Method	MAG-PSY-WIKI				MAG-CS-WIKI			
	MR	HIT@1	HIT@5	HIT@10	MR	HIT@1	HIT@5	HIT@10
TaxoExpan	2688.0 ± 1434	0.070 ± 0.021	0.187 ± 0.045	0.252 ± 0.062	7320.1 ± 3065	0.007 ± 0.003	0.026 ± 0.006	0.047 ± 0.012
TMN	3225.7 ± 1918	0.097 ± 0.022*	0.189 ± 0.043	0.226 ± 0.05	5271.9 ± 4154	0.040 ± 0.009	0.110 ± 0.022	0.150 ± 0.032
Arborist	3698.0 ± 2083	0.046 ± 0.023	0.134 ± 0.032	0.176 ± 0.04	5925.7 ± 4843	0.020 ± 0.007	0.062 ± 0.019	0.095 ± 0.029
TaxoEnrich	2664.9 ± 1473*	0.094 ± 0.023	0.215 ± 0.054*	0.272 ± 0.069*	4954.9 ± 3117*	0.049 ± 0.013*	0.131 ± 0.037*	0.183 ± 0.052*
TaxoComplete	**560.6 ± 23**	**0.170 ± 0.020**	**0.392 ± 0.025**	**0.488 ± 0.019**	**1085.9 ± 115**	**0.166 ± 0.019**	**0.346 ± 0.016**	**0.440 ± 0.018**

- TMN [22]: is used for taxonomy completion where it maps the problem to a one-to-pair matching problem.
- TaxoEnrich [5]: encodes for each query node its ancestral and descendant paths in addition to its siblings. This method was designed for the taxonomy completion task.

We adapt methods designed for taxonomy expansion, namely Arborist and TaxoExpan, to the taxonomy completion problem by using the same comparison strategy used in [5, 22] where we concatenate the representation of the parent node and the child node as the candidate node representation of a given node.

4.1.4 Data Split. For each dataset, we randomly sample 1000 nodes to construct each of our validation and test sets. The remaining nodes represent our seed taxonomy $\mathcal{T}_0$, from which we construct our training set. Each node is associated with a definition from a supporting corpus, from which we learn an initial embedding vector of a fixed-size u_q by leveraging pre-trained language models. We use DistllBert fine-tuned on various question answering datasets [6] as pre-trained word embeddings for all datasets.

4.2 Comparison with the State of the Art

Table 2 summarizes the performance of TaxoComplete against all comparison methods on both the SemEval and MAG datasets. We make several observations.

First, among the baslelines, we observe that TaxoEnrich performs better than other methods in terms of HIT@k on both datasets. Recall that TaxoEnrich encodes the position of a node with its close neighborhood, including its ancestral and descendant paths in addition to its siblings, which better captures structural information than methods encoding only parents and children, such as TaxoExpan and TMN. Second, we observe that methods that encode the local neighborhood of a candidate node with Neural Tensor Networks, such as TaxoEnrich and TMN, consistently outperform other methods on the MAG datasets. This is probably due to the way they model different types of relations in the close neighborhood of a candidate node, which is useful in dense taxonomies such as the MAG datasets, in particular in the MAG-CS-WIKI (see Figure 5 (c,d) in the appendix). Third, we observe that TaxoExpan performs relatively well on the SemEval dataset, while TMN performs better than TaxoExpan on the MAG datasets. Note that the main difference between these two methods resides in their query-anchor matching mechanism, where TaxoExpan uses a log-bilinear model, while TMN uses a channel-wise gating mechanism. This result indicates that for sparse taxonomies such as SemEval (see Figure 5 (a,b) in the appendix), it is hard to learn the relatedness between query nodes and candidate positions using local neighborhoods only.

Most importantly, TaxoComplete achieves the best performance both in MR and HIT@k. Overall, our method is overall 2x better than the second-best method in terms of HIT@k on all datasets. For instance, it achieves 0.170 HIT@1 compared with the second-best method, TMN, which achieves 0.097 HIT@1 in the MAG-PSY-WIKI. Our method also reduces the MR by a large margin, achieving an MR of 560.6, while the second-best method TaxoEnrich achieves 2664.9 MR in the MAG-PSY-WIKI. This substantial difference is mainly due to how we represent the semantic distance between node pairs as dependent on their graph distance in the taxonomy, which encapsulates fine-grained information about a node's position, unlike existing methods that consider pairs of nodes as either related or not. These significant gains demonstrate the effectiveness of our framework in completing taxonomies.

4.3 Ablation Study

To further confirm the impact of our direction-aware propagation module, we conduct an ablation study comparing TaxoComplete to a simplified version with only the query-anchor semantic matching mechanism (TaxoComplete-Matching). We also compare the exact version of the direction-aware propagation module (TaxoComplete-Exact) against its approximation in TaxoComplete. The results are shown in Figure 3.

We observe that TaxoComplete substantially improves the performance of TaxoComplete-Matching by over 59.12% HIT@1 on the SemEval datasets and by 10.54% HIT@1 on the MAG datasets. TaxoComplete also decreases the MR by 15.14% on the SemEval datasets and by 3.84% on the MAG datasets. This result indicates that our direction-aware propagation module substantially improves the

[6] https://huggingface.co/sentence-transformers/multi-qa-distilbert-cos-v1

Table 3: Top-4 predicted parents by TaxoComplete and TaxoComplete-Matching on the MAG-CS-WIKI. The parent of the query node is highlighted in bold.

Query	Predicted Candidates with TaxoComplete
depth first search	beam search, **search algorithm**, graph traversal, incremental heuristic search
interaction overview diagram	**class diagram**, object diagram, communication diagram, sequence diagram
schema migration	**database schema**, database model, database design, information schema
coupling loss	**optical fiber**, aperture to medium coupling loss, photonic chip, reflection loss
Query	Predicted Candidates TaxoComplete-Matching
depth first search	graph traversal, beam search, best first search, brute force search
interaction overview diagram	object diagram, communication diagram, uml tool, **class diagram**
schema migration	**database schema**, information schema, database refactoring, database theory
coupling loss	aperture to medium coupling loss, photonic chip, insertion loss, reflection loss

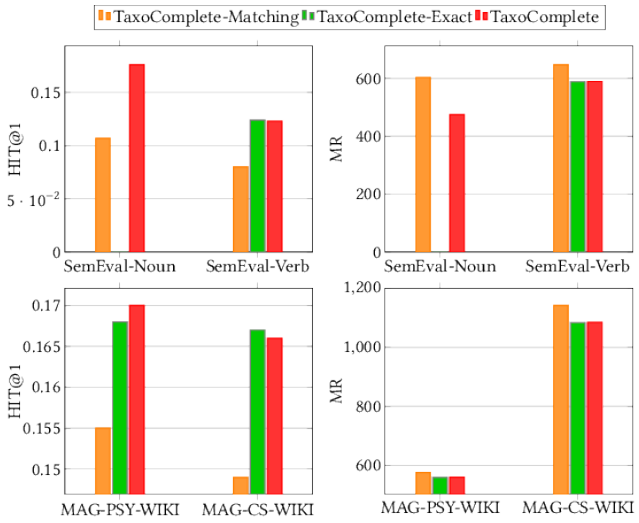


Figure 3: Comparison of TaxoComplete with its variants on SemEval (top figures) and MAG (bottom figures) datasets. Note that TaxoComplete-Exact runs out of memory in the SemEval-Noun dataset.

node representation obtained through the semantic matching mechanism. An in-depth analysis of TaxoComplete-Matching results reveals two main properties: 1) Nodes that are close in the taxonomy are closely placed in the embedding space. We find that the cosine similarity between the node representation vectors obtained using TaxoComplete-Matching of a random sample of pairs of nodes is similar to the inverse of their graph distance, where their Pearson correlation is 0.594 (see Figure 7 in appendix); and 2) TaxoComplete-Matching is unable to distinguish <query, sibling> from <query, parent> relation as illustrated through the example 4.1. We also observe that TaxoComplete-Exact can only be applied on moderately-sized taxonomies, while it runs out of memory on the SemEval-Noun, the largest taxonomy in our dataset. This is probably due to the procedure of TaxoComplete-Exact that builds a dense matrix $\mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$, which is computationally intensive for inference. The performance of TaxoComplete is similar to TaxoComplete-Exact in terms of MR and HIT@1 on the MAG and SemEval-Verb, which confirms the effectiveness of our approximation of direction-aware propagation, as it achieves similar performance to TaxoComplete-Exact while being able to scale to much larger taxonomies.

Example 4.1. Table 3 reports the top-ranked predicted parents for some query nodes by both TaxoComplete and by TaxoComplete-Matching, where we highlight in bold the true parents. We observe that all incorrectly predicted candidates by both TaxoComplete and TaxoComplete-Matching are siblings with the query node in the MAG-CS-WIKI taxonomy. We also observe that TaxoComplete is more likely to identify the true parent in the top-ranked candidates than TaxoComplete-Matching. These results confirm that the semantic matching mechanism alone can identify some relatedness between pairs of nodes but mistakenly identifies the <query, sibling> relation as a <query, parent> relation due to the similarity in the descriptions of query nodes and their siblings. For instance, "depth-first search" is defined as "an algorithm for *traversing* or searching tree or *graph* data structures, etc.", which contains both keywords "traverse" and "graph". These keywords are also found in the definition of "graph traversal", which is the top candidate predicted by TaxoComplete-Matching but is in our taxonomy a sibling to "depth-first search" and not his parent.

4.4 Method Properties

4.4.1 Impact of the Labeling Function. Our loss function strongly depends on the choice of our labeling function (see Eq. 4). The labeling function has to reflect the following intuition: nodes close in the taxonomy should have a higher score than those placed further apart. We compare different labeling functions ($1/d$, $\pm 1/d$, $1/d^2$, $\pm 1/d^2$) that reflect our intuition. The sign in the labeling functions $\pm 1/d$, $\pm 1/d^2$ indicates the direction of the edge between pairs of nodes, where a positive sign indicates a <query, ancestral node> relation while a negative sign indicates a <query, descendant node> relation. Unsigned labeling functions omit the direction within the taxonomy. We also compare with a binary labeling function where a pair of nodes is assigned a label "one" if they have a <query, parent> relation and a label "zero" for all other candidate pairs. Note that existing methods commonly use this binary labeling strategy. We measure the performance of TaxoComplete-Matching on the SemEval-Verb (see Figure 4 (a)) and the MAG-PSY-WIKI (see Figure 4 (b)) datasets in terms of HIT@1, HIT@10, and MR.

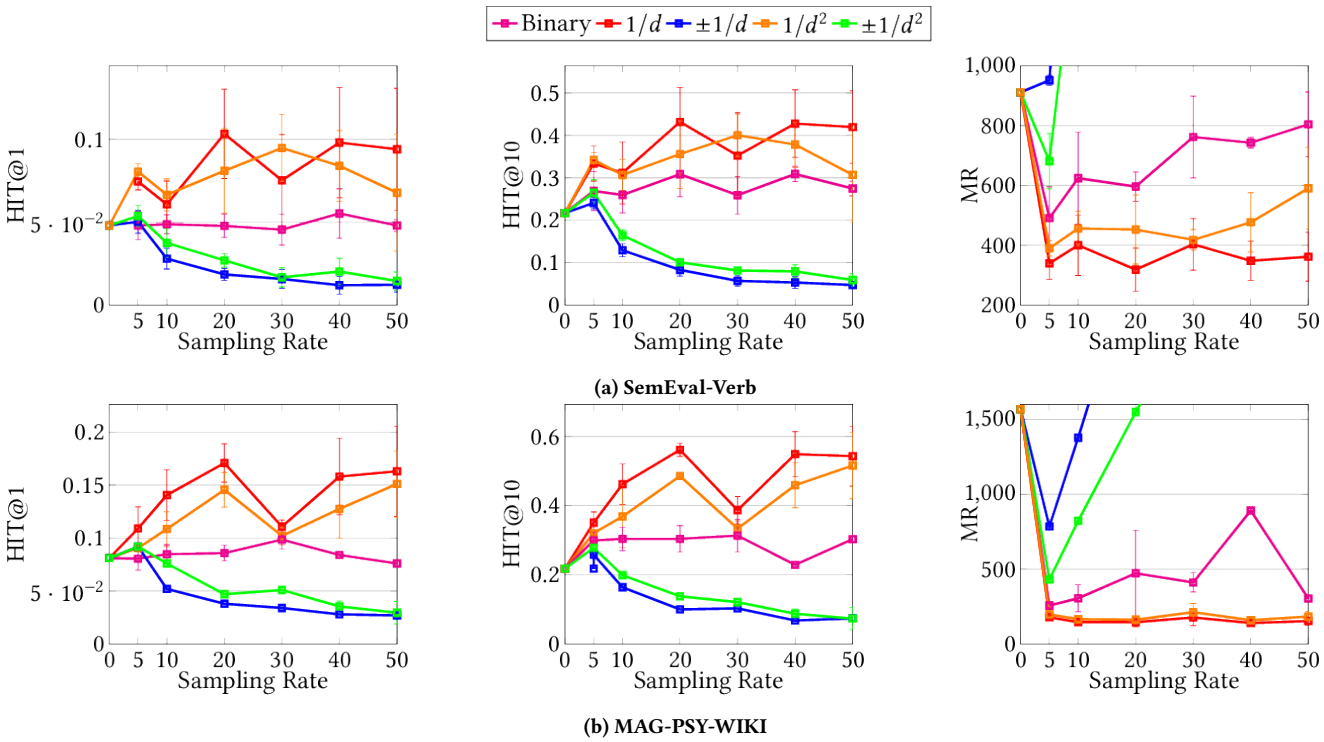


Figure 4: Performance of TaxoComplete-Matching with different labeling functions measured by HIT@1, HIT@10 and MR, while varying the sampling rate on the (a) SemEval-Verb (top figures) and (b) MAG-PSY-WIKI (bottom figures).

We observe that TaxoComplete-Matching with signed labeling function (i.e., $\pm 1/d$, $\pm 1/d^2$) has lower performance than other labeling functions. When using a negative sign to indicate the direction of an edge, TaxoComplete-Matching learns that concepts related as <query, descendant node> represent opposite semantics, which deteriorates its performance. We also observe that TaxoComplete-Matching with unsigned labeling functions (i.e., $1/d$, $1/d^2$) achieves better performance than it does with binary and signed labeling functions. These results show the effectiveness of using the graph distance between nodes in learning their semantic relatedness.

4.4.2 Impact of the Sampling Rate. The sampling rate s_r controls the size of the randomly sampled candidate nodes in the distant neighborhood of a query node. We study the impact of this rate in Figure 4, where we vary the sampling rate between 0 and 50 on the SemEval-Verb (see Figure 4 (a)) and the MAG-CS-WIKI (see Figure 4 (b)). We measure the performance of TaxoComplete-Matching with all discussed labeling functions in Section 4.4.1 in terms of HIT@1, HIT@10, and MR. First, we observe that, as the sampling rate increases, the performance of TaxoComplete-Matching with signed labeling functions decreases. Second, we observe that the sampling rate does not impact the performance of TaxoComplete-Matching with the binary labeling function, particularly in HIT@1, where it remains almost constant when varying the sampling rate. Finally, we observe that the performance of TaxoComplete-Matching with unsigned labeling increases when increasing the sampling rate until reaching a sampling rate of 20, then it fluctuates with higher sampling rates. Such a result is consistent on both datasets, measured by HIT@1, HIT@10, and MR. The optimal performance is reached by TaxoComplete-Matching with the labeling function $\frac{1}{d}$ for $s_r = 20$ on both datasets. The observed performance fluctuation of TaxoComplete-Matching could be due to the large size of the distant neighborhood compared with the local neighborhood (over 30x), which decreases TaxoComplete-Matching ability to discriminate between the two. Overall, the variation of the performance of TaxoComplete-Matching with different s_r indicates the importance of selecting an adequate sampling rate.

5 CONCLUSION

In this paper, we presented TaxoComplete, a self-supervised taxonomy completion framework that learns the representation of nodes leveraging their position in the taxonomy. Our framework draws inspiration from recent advances in semantic matching to learn the semantic distance separating pairs of nodes. In addition, it injects the direction of the edges into the node representations using a direction-aware propagation module. Extensive validation on real-world datasets demonstrates the effectiveness of TaxoComplete, which substantially outperforms state-of-the-art methods on taxonomy completion tasks.

ACKNOWLEDGMENTS

This work was supported by the armasuisse Science and Technology, R&D agency of the Swiss Armed Forces.

REFERENCES

[1] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*. OpenReview.net.

[2] Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulić, and Iryna Gurevych. 2021. Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval. *arXiv preprint arXiv:2103.11920* (2021).

[3] Rafael S Gonçalves, Matthew Horridge, Rui Li, Yu Liu, Mark A Musen, Csongor I Nyulas, Evelyn Obamos, Dhananjay Shrouty, and David Temple. 2019. Use of OWL and semantic web technologies at pinterest. In *International Semantic Web Conference*. Springer, Springer, Switzerland, 418–435.

[4] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*. OpenReview.net.

[5] Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. 2022. TaxoEnrich: Self-Supervised Taxonomy Completion via Structure-Semantic Representations. In *Proceedings of the ACM Web Conference 2022*. ACM, USA, 925–934.

[6] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, Online, 6769–6781.

[7] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations?. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. The Association for Computational Linguistics, Denver, Colorado, 970–976.

[8] Zichen Liu, Hongyuan Xu, Yanlong Wen, Ning Jiang, Haiying Wu, and Xiaojie Yuan. 2021. TEMP: Taxonomy Expansion with Dynamic Margin Loss through Taxonomy-Paths. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3854–3863.

[9] Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. ACM, USA, 2645–2652.

[10] Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*. ACM / IW3C2, USA, 2044–2054.

[11] Emaad A. Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding Taxonomies with Implicit Edge Semantics. In *WWW*. ACM / IW3C2, USA, 2044–2054.

[12] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992.

[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[15] Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. TaxoExpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*. ACM / IW3C2, USA, 486–497.

[16] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*. ACM, USA, 243–246.

[17] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 296–310.

[18] Chaoqun Yang, Yuanyuan Zhu, Ming Zhong, and Rongrong Li. 2019. Semantic similarity computation in knowledge graphs: Comparisons and improvements. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, IEEE, USA, 249–252.

[19] Hui Yang. 2012. Constructing task-specific taxonomies for document collection browsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. ACL, Jeju Island, Korea, 1278–1289.

[20] Xiaoxin Yin and Sarthak Shah. 2010. Building taxonomy of web search intents for name entity queries. In *Proceedings of the 19th international conference on World wide web*. ACM, USA, 1001–1010.

[21] Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, USA, 1026–1035.

[22] Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. 2021. Taxonomy completion via triplet matching network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. AAAI Press, USA, 4662–4670.

[23] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. 2014. Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, USA, 243–252.

[24] Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. 2021. MATCH: Metadata-aware text classification in a large hierarchy. In *Proceedings of the Web Conference 2021*. ACM / IW3C2, USA, 3246–3257.

[25] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774* (2021).

[26] Ganggao Zhu and Carlos A Iglesias. 2016. Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2016), 72–85.

A APPENDIX

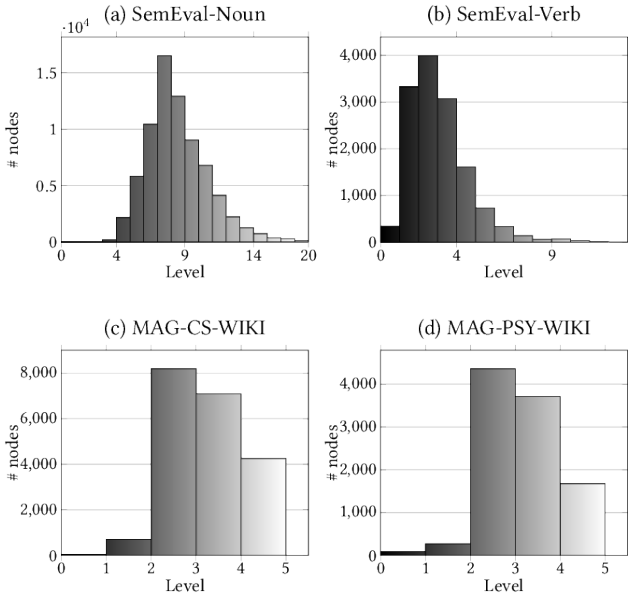


Figure 5: Distribution of nodes over the levels of the taxonomy in (a,b) SemEval and (c,d) MAG.

A.1 Dataset Properties

Figure 5 and 6 illustrate the properties of our datasets. We represent the number of nodes per level for the SemEval datatset in Figure 5 (a,b) and the MAG datatset in Figure 5 (c,d). We observe that the SemEval dataset have a skewed distribution of nodes over different levels of the taxonomy compared with the MAG dataset. We also represent the number of nodes per degree for the SemEval datatset in Figure 6 (a,b) and the MAG datatset in Figure 6 (c,d). For clarity sake, we do not represent the number of nodes with a degree one because they are very high compared with the number of nodes with a degree two, where there are 43622 nodes with degree one on average in SemEval datasets and 17444.5 on average in MAG datasets. We observe that the decrease of the number of nodes

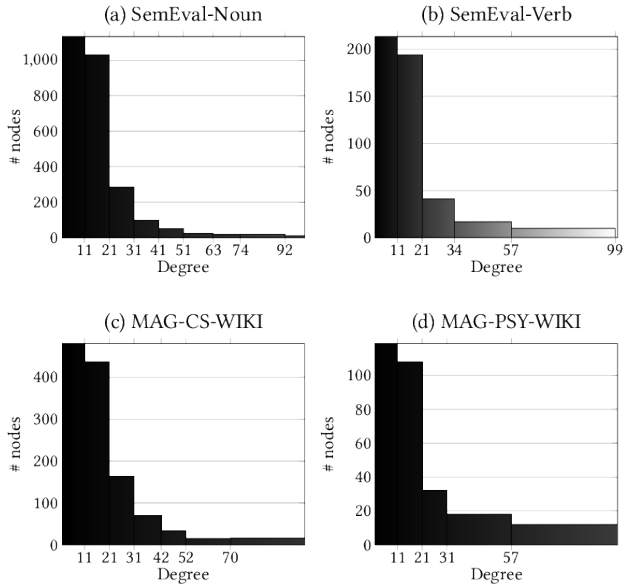


Figure 6: Number of nodes per degree in (a,b) SemEval taxonomy and (c,d) MAG taxonomy.

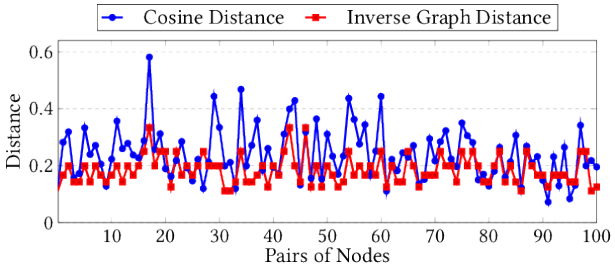


Figure 7: Comparison of the cosine distance with the inverse of the graph distance in MAG-CS-WIKI.

per degree is slower in the MAG dataset than the SemEval dataset. Hence, the MAG dataset is denser than the SemEval dataset.

A.2 Semantic Similarity with TaxoComplete-Matching

Figure 7 illustrates the comparison between the semantic similarity computed using TaxoComplete-Matching and the inverse of the graph distance separating a random sample of pairs of nodes in the MAG-CS-WIKI. To measure the semantic similarity, we use the cosine distance between pairs of node representation. We observe that the cosine distance correlates with the inverse of the graph distance which shows TaxoComplete-Matching ability to learn the taxonomic position of nodes.