Taxonomy Completion via Sector Hierarchy

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

M.Tech, CSAI

BY

Indraayudh Talukdar
Under the guidance of
Dr. Md Shad Akhtar

Computer Science and Engineering

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI
NEW DELHI– 110020

July,2024

Certificate

This is to certify that the thesis titled "Taxonomy Completion via Sector Hierarchy" being submitted by Indraayudh Talukdar to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

Dr. Md. Shad Akhtar
Department of Computer Science Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi 110 020
July, 2024

Acknowledgement

I extend my heartfelt gratitude towards my supervisor Dr. Md. Shad Akhtar for guiding me and providing me with an opportunity to work on this wonderful topic. I would also like to extend my gratitude towards Dr. Tanmoy Chakraborty for providing expert guidance to enrich my knowledge and enhance my research.

Abstract

Taxonomy is a hierarchical structure that deals with knowledge. It can be perceived as Knowledge graph with all the relations being only 'is-a'. Automatic creation of taxonomies can be achieved by creation scratch, by completion of existing taxonomy and by expansion of existing taxonomies. In taxonomy expansion, nodes can be inserted at the leaf only, while in taxonomy completion nodes can be inserted into any position of the existing taxonomy. In this thesis taxonomy completion has been explored with the introduction of polar embedding for a term. The taxonomy is perceived as a set of concentric circles and the inheritance of the terms is injected into the sectors formed on the circles. Each circle indicates a level of the taxonomy. As we go down the taxonomy, the radius of a circle or orbit increases exponentially. Semantically similar terms in the first orbit(or circle) are clustered to form a sector. The sectors in the following orbits(or circles), inherit the children of their corresponding sectors in the previous orbit. The number of sub-sectors of a particular sector indicates the number of sectors which will be children to it. Now, a sector can be represented using its starting and ending points which are angles. A term has a unique ID represented using r-hot vectors, which are one-hot vectors, but the non-zero values indicate the level of the term. In previous literature of taxonomy completion, most of the time the parent is found out using a scoring method. This thesis directly predicts the parent and the concept of unique ID described above is instrumental in doing so. In this thesis, two main things occur which do not follow the standard trend in the works of taxonomy completion or expansion. One of them is the use of polar coordinates, where radius indicates the level of the term and the angles indicate the place where the term will be present in a particular level. The other part is the use of unique IDs and direct prediction of the parent. Predicting the parent directly would save a lot of time during inference and also it would relieve us of the idea that we should be ranking parents. The reason being parents of a node cannot be ranked since it should be exact. Thus, this thesis tries to save time and tries to give exact information for a taxonomy.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Taxonomy is a hierarchical structure indicating world knowledge. It can be perceived as knowledge graphs with relation 'is-a'. All the relations are marked as 'is-a'. Taxonomies are of two types: Topic taxonomy and Concept taxonomy.Concept taxonomy is just a tree formed with entities and has no restriction regarding the children of a node. In topical or topic taxonomy, each term would be associated a cluster of terms and its children would belong to this cluster only. Automatic creation of a taxonomy is a difficult task and has mainly 3 parts, namely, Taxonomy Creation, Taxonomy Expansion, Taxonomy Completion. For creation initially there would not be manually annotated partial taxonomy called 'seed'.In taxonomy expansion , we need to add nodes as the leaf nodes in the seed taxonomy. In taxonomy completion, the nodes are added in any position of the taxonomy. This thesis mainly focuses on taxonomy completion. Taxonomies have to be highly accurate as they are required for representing knowledge or reasoning in large language models. Also, taxonomies can be used for recommendations , creating International Classification of Diseases(ICD) in medicine, prioritization of web pages by significance and user preferences and in many more fields like education, e-commerce and so on. We can see that taxonomies have huge applications and so, it is essential that taxonomies are highly accurate and precise. However, in practice, the situation is far from it. This leads us to the motivation for this project.

Chapter 2

Motivation

In the introduction, the application of taxonomies was discussed. This provided a compulsion for creating precise taxonomies, but in reality this standard is not met. Most of methods, be it taxonomy expansion or taxonomy completion, use a scoring based mechanism. The model is used to predict score when a pair of nodes is passed to it, or a path and a node is passed. The model would either provide a distance value or a score. If it is distance, the lower the better and vice-versa for scores. Other methods would include embeddings in hyperbolic space or other spaces which are non-euclidean. These embedding spaces guarantee vectors with hierarchical information. However, polar coordinates have not been used for this task initially. Also, rather than creating a vector(considering as the physical address) of the term to infuse the hierarchical quality, logical address can be used for that and we can map the logical address to the physical address of the term. The usage of a logical address and a physical address for a particular term and polar embedding is not seen prior to this thesis for taxonomy completion. Also, most approaches for taxonomy expansion and completion use a scoring based function for pair of nodes or a path and a node, this is quite time consuming when it comes to inference, as the query node has to be compared with all the other nodes present to find it's parent node. This thesis also tries to go past this method. Thus the main motivation of this project is that it might be difficult to use a single vector which indicates the physical address of a term in a taxonomy as well as provide the hierarchical information and second is to provide a faster inference procedure when it comes to taxonomies with huge amount of nodes and edges. The problem basically boils down to predicting the parent directly from the model using the definition of the query concept term or node.

Chapter 3

Related Works

Automatic creation of taxonomy mainly has three ways, taxonomy creation, taxonomy completion and taxonomy expansion. Taxonomy creation involves creating taxonomies from scratch, which means there will be no previous information about the given taxonomy. In taxonomy expansion, we get a seed taxonomy which provides us some knowledge about the taxonomy. We need to insert nodes into the leaf position only in this problem. In taxonomy completion, we need to insert nodes into any position of the given seed taxonomy.

3.1 Taxonomy Expansion

Taxonomy expansion includes works like TEMP, QEN, DNG, boxtaxo. TEMP checks if a given query node can be attached at the end of a selected path. It is a score based mechanism. Basically, they concatenate the definition of the query with a path and feed all of it to a transformer encoder model like Bert. From there, you will get a score. That score will indicate the strength of the query node getting attached to the given path. [1] basically tries to do parent detection and sibling detection. Along with that, they try to insert nodes such that it might be leaf node or might not. Boxtaxo [2] provides a fresh look into taxonomy expansion. This paper argues that it is not sufficient to have a single vector which can capture hierarchical relations. They call this as Box Embedding. A box embedding includes two vector and using this two we can represent a hyper-rectangle in n-dimensional space. The vectors include the center and the offset. Box embedding is useful when it comes to represent hierarchy, as we can have representations of nodes such that for two nodes p and c, c contained within p might indicate c is child of p. Likewise, the nodes can be disjoint or partially intersecting as well. HEF [3] is another paper which delves into the work of taxonomy expansion. Here they use ego tree of the anchor node. An ego tree basically consists of all the parents and children of the anchor node(node under which the new node called query node is supposed to get attached). They also use relative and absolute level embedding and the description of the query node. Their model tries to detect if the query node is placed on the right path and is on the right level or not.

3.2 Taxonomy Completion

Taxonomy completion includes works like TaxoEnrich, TaxoComplete, TMN. TaxoEnrich[4] is basically a taxonomy completion module. Taxoenrich uses psuedo-paths for ancestors and descendents of the query node. They use it as the vertical view information. For horizontal view information, they selectively choose siblings of the query node. Finally, they check if the embedding of the query node goes well with the aforementioned information. This is indicated by a scoring function. TaxoComplete[5] is another taxonomy completion module. Here we get the concept

of close neighbourhood and distant neighbourhood. Close neighbourhood includes all the ancestral nodes, the query node and its siblings and its children. Distant neighbourhood includes nodes who are not in close neighbourhood. A graph distance based score is used to determine if the anchor node is in close neighbourhood or not for a given query node. To determine the direction of the nodes, a variation of personalized PageRank algorithm is used called personalized propagation of neural predictions. TMN [], also known as Taxonomy Completion via Triplet Matching mechanism, also aims to complete taxonomies. TMN makes use of auxiliary scorers and a primary scorer. TMN specifically uses the triplet of the query node, candidate hypernym and candidate hyponym. The primary scorer gathers information from the auxiliary scorers and prepares the final decision to check if triplet mentioned above actually stands or not. They also designed a gating mechanism to prevent the presence of unnecessary information.

3.3 Taxonomy Creation

Taxonomy creation is the hardest problem among the three areas of taxonomy construction discussed above. It includes works like Graph2Taxo, Taxogen. Taxogen [6] is for generating topic taxonomies. They used a variation of spherical k-means clustering for adapting to the representativeness of the terms to the cluster of a node. This is done to ensure that, all the seemingly similar terms should not come into the cluster of node term. Representativeness is used to determine if a term should be considered in the cluster. Representativeness depends upon the factors of popularity and concentration. Popularity depends on the number of times a term appears in the document of its chosen head term. Concentration tells that a term should be more representative than the clusters of the sibling of its supposed parent. Graph2Taxo[7] is the lone example of domain adaptation in taxonomies. Here, a taxonomy is perceived as a directed acyclic graph(DAG). To adapt to different domains, they use the gold taxonomies and mixed it with noisy parent-child pairs obtained from a corpus to prepare a noisy graph. Then, they applied GCN with pooling on the graph, followed by clustering. Then again a round of GCN with pooling. This part constitutes the encoder. During decoding, they used cosine similarity with threshold.

Chapter 4

Dataset

Datasets used include MAG-CS and MAG-PSY. MAG or Microsoft Academic Graph consists of 660 thousand nodes and 700 thousand edges. Out of this, the MAG-CS and MAG-PSY is curated by the authors of TaxoComplete[5]. The curation procedure is the one developed by the works of TaxoExpan [8]. In TaxoExpan [8], they picked up a sub-graph with CS as root and used it to make the MAG-CS dataset. In similar manner, MAG-PSY was prepared by TaxoComplete[5]. The exact specifications are provided in the table below.

Dataset	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{D}\rvert$
MAG-CS	24,754	42,329	6
MAG-PSY	23,187	30,041	6

Table 4.1: Dataset specifications

Chapter 5

Methodology

5.1 Baselines

The baselines for this thesis include the works of TaxoComplete, TMN and TaxoEnrich. TaxoEnrich[4] utilizes the vertical and horizontal views of taxonomy. They create pseudopaths of the ancestors and the descendents of the query node. They also select siblings such that their information does not become noise. Their model selects siblings based on the relatedness to the query node. Finally, they check a fitness score how well embeddings of the pseudopaths and the siblings go with the embedding of the query node. TaxoComplete[5] uses close neighbourhood and distant neighbourhood concepts to determine the neighbourhood the query node likely to be in.

5.2 Sector Embeddings

This thesis considers a concept taxonomy as concentric circle. Each level of the taxonomy would be a circle from the set of concentric circles. Here, circles are called orbits and in rest of the work, the levelled circles will be called orbits. In each orbit we will have a number of sectors. All the terms are clustered such that semantically closer ones are present in the particular sector or in neighbouring sectors.The radius value of an orbit is dependent upon the level it is on. A sector can be represented using two angles, θ_{st}, θ_{end}. Here θ_{st} is the angle where the sector begins and θ_{end} is the angle where the sector ends. The representation of the sector is depicted using polar coordinates. So, we will require the radius of the sector, which would be the radius of the orbit the sector lies in. Mathematically, we can represent the sector embedding in the following manner,

$$\mathbf{e}_{\mathbf{s}c(o_i,j)} = r_{o_i} \cdot [sin(\theta^{st}_{s_{(o_i,j)}}), cos(\theta^{st}_{s_{(o_i,j)}}), sin(\theta^{end}_{s_{(o_i,j)}}), cos(\theta^{end}_{s_{(o_i,j)}})] \tag{5.1}$$

Here r_{o_i} represents the radius of the orbit o_i. $s_{(o_i,j)}$ represents j^{th} sector of orbit o_i. Now, the sectors in the first orbit are formed using K-Means clustering and ELBOW method. The number of clusters range from 2 to number of terms in the o_1. The elbow is obtained using KneeLocator algorithm. The KneeLocator algorithm constructs a straight line from the starting point to end point of the given curve. Then it measures the perpendicular distance of the points on the curve from the line. The point having maximum distance will be selected as the elbow. This elbow value is used for creating radius values of the orbits. Let us call it c. The radius value of an orbit o_k will be termed as c^k. After clustering is done in orbit o_1, one of the cluster is randomly selected and the rest of them are aligned such that they are K-nearest neighbours to the first sector or $s_{(o_1,1)}$. In the following orbits, we do not have clustering to determine the sectors. The number of sector at each orbit can be perceived as a function of its radius and number of sectors in its previous orbit. Mathematically,

we can define the number of sectors using the following recursion.

$$\begin{aligned} &|sec(o_i)| = f_{bf}(r_{o_i}, sec(pr_o(o_i))) = f_{bf}(r_{o_i}, sec(o_{i-1})) \\ &= r_{o_i} \times |sec(o_{i-1})| \end{aligned} \tag{5.2}$$

The base at the root level is shown below.

$$|sec(o_{root})| = 1 \tag{5.3}$$

The base at orbit o_1 is given in the following equation. As discussed earlier, in the first orbit or orbit o_1, number of sectors depend upon number of clusters formed by using elbow method on K-Means clustering. The following equation represents it mathematically.

$$\begin{aligned} &|sec(o_1)| = r_{o_1} \times |sec(o_{root})| = r_{o_1} = c^1; r_{o_i} = c^{o_i} \\ &c \text{ based on ELBOW method during } g_{\text{k-Means}}(.) \text{ with some margin} \end{aligned} \tag{5.4}$$

$$\begin{aligned} &|sec(o_i)| = f_{bf}(r_{o_i}, sec(pr_o(o_i))) = f_{bf}(r_{o_i}, sec(o_{i-1})) \\ &= r_{o_i} \times |sec(o_{i-1})| \end{aligned} \tag{5.5}$$

Rather, the concept of subsector is introduced for this. Before discussing the concept of subsectors, we need to understand the definition of generalized span of a sector and term angular density. Term angular density can be defined as the angle between any two consecutive terms. We represent it as $\Delta\theta_c$ for orbit o_1. Mathematically we can represent it using the following equation.

$$\begin{aligned} &\Delta\theta_c = \frac{2\pi}{|\{c(o_1, \bullet)\}|} \\ &\text{Here, } |\{c(o_1, \bullet)\}| \text{denotes the number of concepts in the orbit} o_1. \end{aligned} \tag{5.6}$$

Next up we discuss span of a sector. Simply put, span is the angle a sector subtends in its orbit. It can be represented as the $\Delta_s^{(o_i,j)}$. The below equation defines it in the terms of the angles representing a sector.

$$\Delta_s^{(o_i,j)} = \theta^{end}_{s_{(o_i,j)}} - \theta^{end}_{s_{(o_i,j)}} \tag{5.7}$$

Also, we can recursively determine the span of a sector from its parent sector. This recursive formulation of the span helps us provide a hierarchy to the sectors. The equation below determines the span of a sector recursively.

$$\begin{aligned} &\Delta_s^{(o_i,j)} = \frac{\Delta_s^{pr_s(o_i,j)}}{|\{s\text{-}sec(pr_s(o_i,j))\}|} \\ &\text{Base case at orbit } o_1 : \\ &\Delta_s^{(o_1,j)} = \left[|\{c(o1, \bullet) : c(o1, \bullet) \in Cl_{(o_1,j)}\}| \times \Delta\theta_c\right] \end{aligned} \tag{5.8}$$

Here $pr_s(o_i, j)$ is the parent sector for the sector $(o_i, j$, $|\{s\text{-}sec((o_i, j))\}|$ tells the number of subsectors present in the given sector (o_i, j). This span value is same for all the child sectors of a particular sector. Also this recursive approach prevents us from clustering at every stage in the sector creation. Clustering at every stage of making the sectors makes the situation overly dependent upon the data provided. The subsector concept is very significant when it comes to creating children sectors. The number of children sectors is equal to the number of subsectors. The number of subsectors of sector is again a function of the entity $\alpha_{ss_{(o_i.j))}}$ and number of subsectors of its parent sector. Mathematically, it can be represented in the following equation.

$$|s\text{-}sec(s_{c(o_i,j)})| = \alpha_{ss_{(o_i,l)}} \times |s\text{-}sec(s_{pr_c(c_{(o_i,j)})})| \tag{5.9}$$

Here, $c_{(o_i,j)}$ indicates j^th concept at orbit o_i, $pr_c(c_{(o_i,j)})$ indicates parent of the given concept or term. The base case at orbit o_1 is determined in the following manner.

$$|s\text{-}sec(s_{c(o_1,j)})| = |s\text{-}sec((o_1, l))| = \frac{\Delta_s^{(o_1,l)}}{\Delta\theta_c} = \alpha_{ss_{(o_1,l)}} \tag{5.10}$$

After we get the span of a child sector, we need to determine the beginning and end angles of the child sectors. Let a child sector be (o_i, j). Then, it's parent sector will be $pr_s(o_i, j)$.

$$\begin{aligned}
\theta^{st}_{s_{o_i,j}} &= \theta^{st}_{pr_s(s_{o_i,j})} + \delta_{s_{(o_i,j)}} \\
\delta_{s_{(o_i,j)}} &= \Delta_s^{(o_i,j)} \times |\{\mathbf{s}_{(o_i,k)} : k = [0, l)\}| \\
\theta^{end}_{s_{o_i,j}} &= \theta^{st}_{s_{o_i,j}} + \Delta_s^{(o_i,j)}
\end{aligned} \tag{5.11}$$

To assign sectors to a term, we need to know its parent's sector. After knowing the parent sector, we can add terms well within the vicinity of the angular span of the parent sector. However, there can be an issue of overflow of terms in the collection of child sectors. The reason being, if we keep the angle term density constant,total angular span might cross 2π which is not desirable for us. Hence, to avoid this, we need to make the angle term density a function of $\Delta\theta_c$ and the radius r_{o_i} of given orbit o_i. Hence, angle term density for each orbit can be represented as a ratio of angle term density at orbit o_1 and a function of orbit value i, and c being the number of clusters at orbit o_1. The following equation would provide the mathematical equation for it.

$$\Delta\theta_c^{(o_i,j)} = \frac{\Delta\theta_c}{c^{(i-1}} \tag{5.12}$$

This above orbit based definition of angle term density helps us accommodate exponential increase in the number of children in lower orbits down the levels of the taxonomy. All these mathematical parts ultimately lead us to the development of the sector embedding we discussed earlier in this section. We need the span of sector to determine the way to find the $\theta^{st}_{(o_i,j)}$ and $\theta^{end}_{(o_i,j)}$. Once we know this, we can determine the sector locations. The radius value is dependent on the initial number of clusters. It's equation is defined in (5.4). Thus, we see that the sectors created have a sense of hierarchy in them. This helps prepare unique ID of terms such that they do not need to have hierarchical information in them. The sector embedding can be thought of as a logical address which can be used to find the hierarchical information.

5.3 Term UID Representation

In the previous section, sector embeddings were discussed. These embeddings helped us find a way to determine hierarchy among the terms, though many terms can occur in a particular sector. However, the sector embedding do not actually provide us an unique representation of a term as many terms can be present in a particular sector. Hence, a UID representation is required for this. There many ways a term can be represented. We can represent a term in one-hot encoding, taking the whole vocabulary size into account. The problem is, this will face huge computational issue with exponential rise in size of the vocabulary. To mitigate this, we can introduce dense embedding. However, then we do not need the sector embedding and we will try to inject hierarchical information in the dense embedding itself. Doing so will result in a situation where the similar terms should be close enough. However, there might be situations where normalized values in the embeddings, bring not so similar concepts closer. To avoid these things, the physical and logical addresses are separated out. Since we cannot do one-hot vector and neither can we do a dense embedding, we choose r-hot vector to represent a unique ID vector. In r-hot vector, we denote terms in a particular orbit via one-hot vector. However, the non-zero dimension is not 1. Rather it is radius value of the orbit. So, the maximum dimension is basically the maximum value of number of terms per orbit. Mathematically we can view the embedding in the following equation.

$$\begin{aligned} &\mathbf{e_{uid}}_{c(o_i,j)} = r_{o_i} \times \vec{1}_{uid_{j=1:N_{uid}}}(c_{(o_i,j)}) \\ &N_{uid} = max(\{|\{c(o_i, \bullet\}|\forall i \in \mathcal{D}\}) \text{Here, } \mathcal{D}] \text{ is the set of all the orbits.} \end{aligned} \tag{5.13}$$

This method of r-hot vector helps to implement a robust approach for Unique ID for the terms. This approach is median approach to the two extremes of highly sparse one-hot vector and highly dense embedding.

5.4 Aggregation of Embeddings

We need to have an aggregated embedding or tensor for a concept. Primarily, a concept should have 2 embeddings to its unique tensor representation, which being sector embedding and UID. However, we will be keeping three vectors in the tensor. The reason being we are doing one-shot inference for the parent. The model should be predicting the parent information directly. Our model should learn to represent the UID of the child, UID of parent and the sector where the child should be present. Mathematically, we can determine the aggregated tensor of concept $c(o_i, j)$ as $T_{c(o_i,j)}$. The equation will show how to represent it.

$$T_{c(o_i,j)} = aggregate(e_{s_{c(o_i,j)}}, e_{uid_{c(o_i,j)}}, e_{uid_{pr_{c(o_i,j)}}}) \tag{5.14}$$

5.5 Model Architecture

The model architecture to incorporate the above concepts is fairly simple. Let us first discuss what we need to include in the architecture. The first step is to encode

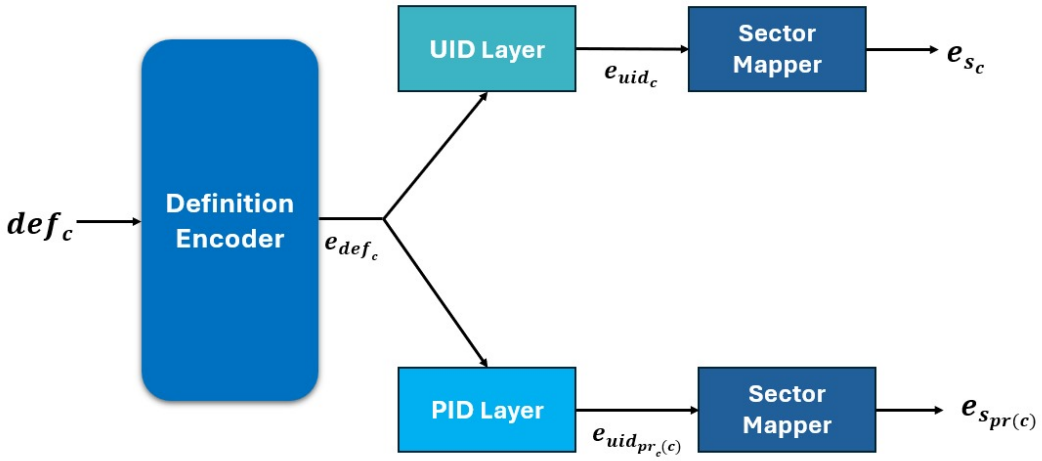

Figure 5.1: Model Architecture

the input definitions of the concepts. For this we require an encoder. It can be Bert, it can be SBert or any other LLM for this part. After we acquire the emebdding for the definition or description of the concept, by using CLS token if using Bert or by aggregating the embeddings when using SBert. Now, this embedding has to be used for UID finding and parent's UID finding. For this we require two linear layers. One of the linear layers will be for finding UID of the query concept and another would be for finding the UID of parent. The outputs of the both of these layers have to normalized using sigmoid function. Now, the UIDs have to mapped to their respective sectors. So, there should be a layer to map that as well. We can do that using a linear layer for that. This layer shows the mapping between physical address and logical address. The equations of model are listed below.

$$\begin{aligned} e_{def_c} &= Encoder(def_c) \\ \hat{e_{uid_c}} &= sigmoid(MLP_{uid}(e_{def_c})), \\ \hat{e_{uid_{pr_c(c)}}} &= sigmoid(MLP_{pid}(e_{def_c})) \\ \hat{e_{s_c}} &= MLP_{sector_{map}}(\hat{e_{uid_c}}) \\ \hat{e_{s_{pr_c(c)}}} &= MLP_{sector_{map}}(\hat{e_{uid_{pr_c(c)}}}) \end{aligned} \tag{5.15}$$

Encoder can be replaced with Bert or SBert. The diagram for the given work will be displayed in

5.6 Losses

The loss mechanism is quite interesting in this aspect. We need two losses who will not be combined together. We will have losses for parent finding, UID finding and also for the sectors. Now, the loss for sector mapping will be combined with loss of UID., for both parent as well as query concept. The loss to be used is MSE.

Mathematically, we can represent it using the following equations.

$$
\begin{aligned}
&\mathcal{L}_{s_c}(e_{s_c}, \hat{e_{s_c}}) = MSE(e_{s_c}, \hat{e_{s_c}}) \\
&\mathcal{L}_{uid_c}(e_{uid_c}, \hat{e_{uid_c}}) = MSE((e_{uid_c}), \hat{e_{uid_c}}) \\
&\mathcal{L}_c = \mathcal{L}_{s_c} + \mathcal{L}_{uid_c} \\
&\mathcal{L}_{s_{pr_c(c)}}(e_{s_{pr_c(c)}}, \hat{e_{s_{pr_c(c)}}}) = MSE(e_{s_{pr_c(c)}}, \hat{e_{s_{pr_c(c)}}}) \\
&\mathcal{L}_{uid_{pr_c(c)}}(e_{uid_{pr_c(c)}}, \hat{e_{uid_{pr_c(c)}}}) = MSE((e_{uid_{pr_c(c)}}), \hat{e_{uid_{pr_c(c)}}}) \\
&\mathcal{L}_{pr_c(c)} = \mathcal{L}_{s_{pr_c(c)}} + \mathcal{L}_{uid_{pr_c(c)}}
\end{aligned}
\tag{5.16}
$$

The final two losses defined $\mathcal{L}_c$ and $\mathcal{L}_{pr_c(c)}$ operate independent of each other. This means that, the loss pertaining to parent finding will not cause a trouble to finding the UID of the query concept. So, parameters will be updated separately based on these two losses. The reason being both are representations of separate entities. If we had interacted two of them, then representations of two different would impact each other's representation, which we do not want.

Chapter 6

Result

Taxonomy Completion via Sector Hierarchy(TCSH) is the model that has been discussed in this thesis. It beats the existing works of TMN[9], TaxoComplete [5] and TaxoEnrich [4]. It beats scoring methods and proves that separating the logical address from the pysical address would be helpful. The hierarchy should be inserted into the logical address. Also this separation allows for the direct determination of the parent, rather than interacting with all of the nodes to determine the parent.

Method	Dataset	HIT@1
TMN	MAG-PSY	0.097 ± 0.022
TMN	MAG-CS	0.04 ± 0.009
TaxoEnrich	MAG-PSY	0.094 ± 0.023
TaxoEnrich	MAG-CS	0.049 ± 0.013
TaxoComplete	MAG-PSY	0.170 ± 0.020
TaxoComplete	MAG-CS	0.166 ± 0.019
TCSH	MAG-PSY	0.24
TCSH	MAG-CS	0.21

Chapter 7

Future Scope

In the result section we witnessed that TCSH beats the existing models for taxonomy completion. However, the margin is not so much. In future works for this, the radius value can included in a positive manner. In other words, we can increase the span in subsequent orbits by including the radius value with angular span to determine the final span of the sectors. It will also eardicate the use of orbit dependent angle term density.

Bibliography

[1] Suyuchen Wang, Ruihui Zhao, Yefeng Zheng, and Bang Liu. QEN: Applicable Taxonomy Completion via Evaluating Full Taxonomic Relations. In *Proceedings of the ACM Web Conference 2022*, pages 1008–1017, Virtual Event, Lyon France, April 2022. ACM. ISBN 978-1-4503-9096-5. doi: 10.1145/3485447.3511943. URL https://dl.acm.org/doi/10.1145/3485447.3511943.

[2] Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. A Single Vector Is Not Enough: Taxonomy Expansion via Box Embeddings. In *Proceedings of the ACM Web Conference 2023*, pages 2467–2476, Austin TX USA, April 2023. ACM. ISBN 978-1-4503-9416-1. doi: 10.1145/3543507.3583310. URL https://dl.acm.org/doi/10.1145/3543507.3583310.

[3] Suyuchen Wang, Ruihui Zhao, Xi Chen, Yefeng Zheng, and Bang Liu. Enquire One's Parent and Child Before Decision: Fully Exploit Hierarchical Structure for Self-Supervised Taxonomy Expansion. 2021. doi: 10.48550/ARXIV.2101.11268. URL https://arxiv.org/abs/2101.11268. Publisher: arXiv Version Number: 1.

[4] Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. TaxoEnrich: Self-Supervised Taxonomy Completion via Structure-Semantic Representations. In *Proceedings of the ACM Web Conference 2022*, pages 925–934, Virtual Event, Lyon France, April 2022. ACM. ISBN 978-1-4503-9096-5. doi: 10.1145/3485447.3511935. URL https://dl.acm.org/doi/10.1145/3485447.3511935.

[5] Ines Arous, Ljiljana Dolamic, and Philippe Cudré-Mauroux. TaxoComplete: Self-Supervised Taxonomy Completion Leveraging Position-Enhanced Semantic Matching. In *Proceedings of the ACM Web Conference 2023*, pages 2509–2518, Austin TX USA, April 2023. ACM. ISBN 978-1-4503-9416-1. doi: 10.1145/3543507.3583342. URL https://dl.acm.org/doi/10.1145/3543507.3583342.

[6] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. TaxoGen: Unsupervised Topic Taxonomy Construction by Adaptive Term Embedding and Clustering. 2018. doi: 10.48550/ARXIV.1812.09551. URL https://arxiv.org/abs/1812.09551. Publisher: arXiv Version Number: 1.

[7] Chao Shang, Sarthak Dash, Md. Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, and Alfio Gliozzo. Taxonomy Construction of

Unseen Domains via Graph-based Cross-Domain Knowledge Transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2198–2208, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.199. URL https://www.aclweb.org/anthology/2020.acl-main.199.

[8] Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. TaxoExpan: Self-supervised Taxonomy Expansion with Position-Enhanced Graph Neural Network. In *Proceedings of The Web Conference 2020*, pages 486–497, Taipei Taiwan, April 2020. ACM. ISBN 978-1-4503-7023-3. doi: 10.1145/3366423.3380132. URL https://dl.acm.org/doi/10.1145/3366423.3380132.

[9] Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. Taxonomy Completion via Triplet Matching Network, 2021. URL https://arxiv.org/abs/2101.01896. Version Number: 3.