2024

Hierarchical multi-label text classification in Earth science datasets

Rajashree Dahal

HIERARCHICAL MULTI-LABEL TEXT CLASSIFICATION IN EARTH SCIENCE DATASETS

Rajashree Dahal

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in
The Department of Computer Science
to
The Graduate School
of
The University of Alabama in Huntsville

May 2024

Approved by:

Dr. Tathagata Mukherjee, Research Advisor/Committee Chair
Dr. Chaity Banerjee Mukherjee, Committee Member
Dr. Vineetha Menon, Committee Member
Dr. Letha Etzkorn, Department Chair
Dr. Rainer Steinwandt, College Dean
Dr. Jon Hakkila, Graduate Dean

Abstract

HIERARCHICAL MULTI-LABEL TEXT CLASSIFICATION IN EARTH SCIENCE DATASETS

Rajashree Dahal

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

Computer Science

The University of Alabama in Huntsville

May 2024

This thesis addresses the challenging problem of hierarchical multi-label text classification and introduces a novel zero-shot approach that recommends the label up to the depth of hierarchy in which it is confident. In order to validate the efficacy of the proposed method, we experimented using various potential embedding models such as text-embedding-ada-002, mpnet-all, instructor embeddings, and nasa-smd-ibm-st on Earth science datasets. The experimental results reveal that all considered embedding models surpass the baseline model supervised learning classifier, demonstrating the superiority of the proposed zero-shot approach. This proposed solution can minimize the label imbalance problem typically observed in the supervised learning approach. The findings from this research can help scholars, researchers, policymakers and environmental scientists better understand and tackle urgent global issues. Experimenting with the proposed framework on datasets belonging to other domains such as biology, physics, medicine, etc. can be a next step to better understand the rigidity of the model.

Acknowledgements

I would like to thank my advisor, Dr. Tathagata Mukherjee, for his continuous guidance and dedication throughout this thesis journey. My master's journey would not have been this easier and smoother without his expertise, insightful feedback, and encouragement at different times in the overall journey.

I am indebted to the Interagency Implementation and Advanced Concepts Team (IMPACT), especially Mr. Muthukumaran Ramasubramanian and Mr. Iksha Gurung, for recognizing this project and providing me with a solid platform to pursue it. I would also like to express additional gratitude to Mr. Muthukumaran Ramasubramanian for his invaluable time and unwavering dedication. His commitment to actively listening to my concepts and constant motivation to refine and enhance them has been instrumental in shaping the development of my ideas. I would also like to thank IMPACT employees for providing me with a valuable platform to share updates. Their insightful suggestions for improvements have added value to the research.

I am grateful to and fortunate enough to get constant encouragement and support from all teaching and non-teaching staff of the Department of Computer science, The University of Alabama in Huntsville, which helped me successfully achieve the objectives of my project. I am confident that this experience will be beneficial in my professional career.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
CLIP	Contrastive Language-Image Pretraining
COVID-19	Corona Virus Disease 2019
DAG	Directed Acyclic Graph
DHC	Deep Hierarchical Classification
GCMD	Global Change Master Directory
GD	Ground Distribution
IMPACT	Interagency Implementation and Advanced Concepts Team
LWAN	Label Wise Attention Network
MLM	Masked Language Modeling
NASA	National Aeronautics and Space Administration

Abbreviation	Description
NLI	Natural Language Inference
NLP	Natural Language Processing
PLM	Permuted Language Modeling
PRF	Precision, Recall, and F-score
SBERT	Sentence-Bidirectional Encoder Representations from Transformers
SMD	Science Mission Directorate
STE	Semantic Text Embedding
SVM	Support Vector Machine
USP	Upward Score Propagation
ZS-MTC	Zero-Shot Multi-Label Text Classification

Chapter 1. Introduction

1.1 Background

There has been a radical shift in different classification approaches over the past few centuries. In this age of information overload, it becomes crucial to categorize documents for better information retrieval. For instance, its observed that every 12 years, the number of publication doubles. By February 2021, there were already over 213,326 papers on COVID-19 [5]. Because of this extensive range of publications, it becomes crucial to accurately categorize them into various levels of themes in order to track the most relevant material. Hierarchical grouping of text is considered a natural and efficient method of organizing texts, especially when multiple labels associated with a specific text-based data set. Extensive research has been conducted on this problem over the last two decades [15].

Gao [6] in their research introduced hierarchical representation in the classification model and hierarchical inconsistency in the training process as two key complexities in hierarchical text classification. If we add the case of multi-label in this classification approach, this will add further complexity to the approach. Items or categories within Earth science, physics, biology, or recommendation systems in retail indeed demonstrate characteristic behaviors or patterns. These domains encompass a wide range of data that contribute to understanding and

solving various issues. A dataset can talk about many topics. For example, a document discusses how snakes evolved in the Savanna region. It includes information about the animal kingdom, and, as we go further, it talks about reptiles and, finally, becomes detailed on snakes. Similarly, it discusses the Savanna region and the land area.

An effective categorization of these datasets is necessary for academics, policymakers, and profitable businesses. Given the complexity and extensive domain that can be formed in these fields, these datasets can be organized more effectively through computer-aided methods such as hierarchical multi-label text classification. Supervised learning classifiers, which are trained on certain labels and find it difficult to forecast on datasets with labels they haven't seen before, are challenged by this disparity. Therefore, a unique method that takes this constraint into account is required to ensure robust classification even for datasets with unobserved labels. The success of the research depends on selecting the correct model and algorithms for hierarchical multi-label text classification. We will explore cutting edge approaches in machine learning and natural language processing to create models that can efficiently classify documents within earth science datasets.

This research has significant potential as the proper categorization makes required information more easily accessible, encouraging multidisciplinary research partnerships, which will help scholars, decision-makers, researchers and scientists better understand and tackle urgent issues in diverse fields.

1.2 Problem Statement

Most text classification approaches focus on correct labels and treat classes other than the ground truth as equally wrong. Our focus lies not only on correctly classifying labels within datasets but also ensuring confidence in their broader categorization. This approach allows us to capture the specialized aspects of the domain, thereby improving information organization and retrieval through more accurate mistakes. The complexities involved in hierarchical classification are as follows:

- A document can belong to multiple labels.
- A single label node can have multiple parents in the hierarchy.
- Keywords can appear at different levels within the hierarchy rather than being restricted to specific level.

These complexities are very common in different domains such as physics, Earth science, biology, and many more. Our goal is to test our proposed method to solve this problem by considering Earth science data.

1.3 Research Objectives

1.3.1 Main Objective

To classify documents into labels representing hierarchical categories to capture the specialized aspects of the domain and improve the information organization and retrieval by working on Earth science datasets.

1.3.2 Specific Objectives

- To classify the document to the depth of hierarchy it is confident in.
- Explore embedding based approach for solving hierarchical multi-label text classification problem.
- Examine methods for managing imbalanced labels, as document exhibit variation in the organization of categories. It will ensure that the model performs equally well, even when they are unevenly distributed.

Chapter 2. Related Works

2.1 Multi-label Text Classification

There have been revolutionary changes in Natural Language Processing (NLP) in the recent years due to advances in massive language modeling, deep learning, and instruction learning. Text classification remains essential component in various downstream applications, including document filtering and search engines, even though NLP tasks vary widely [14]. Likewise, muti-label text classification extends text classification task by assigning multiple potential labels to a give text document. Multi-label text classificaiton problems are common in real-world applications, such as classifying scientific literature, online shopping sites, and more. It demonstrates the significance of text categorization across different domains. The problem definition for multi label text classification is as follows:

Let f be a function that maps each document d_i from the universal set of documents D to a set of labels l_i from the universal set of labels L:

$$f : D \rightarrow l,$$

where $l \subseteq L = \{l_1, l_2, l_3, \ldots, l_k\}$ and L contains k distinct labels.

2.2 Hierarchical Text Classification Complexities

Gao [6] in their research introduced two main challenges in hierarchical classification, which are as follows:

2.2.1 Hierarchical Representation in Classification Model

This complexity discusses how to incorporate hierarchical information into selected text classificaiton models such as SVM and Neural Networks. It involves understanding how to incorporate hierarchical information in selective models such as SVM and Neural Networks. One such research is the DHC model introduced by Gao [6], which directly incorporates class hierarchy information into neural networks. The author has introduced a hierarchical representation sharing strategy, indicating that the representation of one lower layer should include the representation information about its upper layer.

2.2.2 Hierarchical Inconsistency in Training Process

If a text is predicted as "bottle" in the first layer and "snake" in the second layer, then none of the approaches can deal with inconsistency as far as known. To solve this issue, Gao [6] defined a hierarchical loss function composed of the layer loss and the dependence loss. Layer loss is the same loss as in flat classification. However, dependence loss introduces the concept of loss between the layers. When the two predicted classes in different layers do not belong to the parent-child relationship, additional dependence loss will be added. This loss

is hierarchically related and is regarded as punishment when the predictions are not in parent-child structure.

2.3 Approaches Used in Hierarchical Multi-label Text Classification

There have been significant improvement on image classification over the past few decades. However, these have been made by considering performance metrics that treat all classes other than ground class as equally incorrect. Due to this, mistakes are less likely to happen than they formerly were, but when they do, they can be disastrous. It is necessary to evaluate the extremes of severity. Including taxonomic hierarchy tree can be a measure to make better mistakes. In an effort to reduce errors in the situation of hierarchical categorization, Bertinetto *et al.* [1] offered the following three strategies:

- Creating hierarchical loss function by changing the arguments in the loss function.
- Modifying the network's architecture in a hierarchically informed manner.
- Using an alternative embedding to express the class representation.

Likewise, a research by Taoufiq *et al.* [20] on image datasets for hierarchical building classification have introduced a concept of a new multiplicative layer, which is able to improve the accuracy of the finer prediction by considering the feedback signal of the coarse layers. The multiplicative layer, in reality, is an implementation of conditional probability. Likewise, Liu *et al.* [13] in their paper mentioned tree-based approach, embedding-based approach, graph-based approach,

and ensemble-based approach as the main approaches that have been used for hierarchical multi-label text classification problems.

2.4 Challenges in Hierarchical Multi-label Text Classification

2.4.1 Label Sparsity and Imbalance

Label sparsity and imbalance refers to the condition where a few labels have large number of training instances, but many labels are rare. The model may underfit low-frequency labels and overfit high-frequency labels as a result of this distribution. In order to overcome label imbalance, some models use global information or incorporate anticipated labels from earlier levels [22], [16]. However, these models may require large number of parameters and can pose bias difficulties as they lack general knowledge.

2.4.2 Low Resource Labeled Data

One major problem in machine learning is the lack of labeled data. Zero Shot learning can be a solution in cases where some labels are specified and do not have matching training data. Changing this into a closest neighbor search issue is a popular method. Currently, label hierarchies are used in a model based on LWAN to improve zero-shot learning Chalkidis *et al.* [4].Nevertheless, large computations and reduced accuracy are caused by the extensive label space and complex interactions between labels and text.

2.5 Zero-Shot Approaches to Hierarchical Multi-label Text Classification

Not all labels are adequately represented in the training set. On the top of it, label hierarchies are changed on a regular basis, which necessitates the use of models that can generalize zero-shot data. The use of natural langauge names to generate embeddings for each class, model such as CLIP can do exceptionally well in zero-shot classification.

A research by Haj-Yahia *et al.* [8] presents an unsupervised text categorization technique that uses word embeddings to broaden the document's similarity to category labels, which are enhanced with keywords supplied by humans. The words in the document and labels were replaced by their corresponding embeddings, and cosine similarity was calculated between the document and labels. Since the experiment was carried out with pre-trained Glove, and Word2Vec, there is room for experimentation for pretrained transformer models, and other openAI based embedding models. Similarly, Stammbach and Ash [18] in their research used SBERT to encode documents and extract top five nearest neighbors for every datapoint with the intuition that a data point and its nearest neighbor in vector representation point to the same label. The algorithm starts with learning representation via self-learning, extracting the nearest neighbor and fine tuning the network in the weak signal that the two neighbors share the same signal. The most likely clusters are labeled by the weakly supervised model, which records embeddings into category-based clusters during testing. However, the semantics

of embeddings is not leveraged in the process of clustering which can be a room for improvement for further experiments. Also, a document can map to multiple labels, which can complicate the process of obtaining solutions. A potential approach to overcome these two limitations could be using embedding models to extract top k similar labels based on the embedding of the document and embeddings of the whole label space. It will be conducted as one of the baselines in our research. In benchmark tests, LLMs have performed quite well, especially in zero or few-shot conditions. But when it comes to solving real-world problems like hierarchical categorization, these standards frequently fall short. In order to address this, research on restructuring standard tasks on hierarchical datasets into a long-tail prediction job that is more representative have been carried out [2]. The use of entailment-contradiction prediction in conjunction with LLMs is suggested as a solution to overcome the constraints of LLMs in these contexts. This method shows strong performance in stringent zero-shot conditions without necessitating resource-intensive parameter changes across several datasets. The research notes that LLMs do not perform well as a stand-alone model for long-tail classification because of their constraints. By priming the model with an entailment prediction via a prompt, these outcomes can be enhanced. Reinforced Label Hierarchical Reasoning is a revolutionary technique that was developed in a paper by H. Liu *et al.* [11]. The goal of this method was to train for the Zero Shot Multi-Label Text Classification challenge by encouraging dependency between labels inside hierarchies. On the ZS-MTC task across three real-world datasets, the addition of a rollback algorithm—which may correct logical flaws in predictions

during inference—showed improved performance above previous non-pretrained approaches.

Yin *et al.* [23] proposed a method that converts each label into a hypothesis (*e.g.,* "This document is about label") and refines BERT on three Natural Language Inference (NLI) tasks in order to tackle zero-shot text categorization. After that, the model decides if the document fits the hypothesis and applies the appropriate label. But the process adds arbitrary judgment in the formulation of hypothesis I. However, setting the hypothesis for a document for all the labels (suppose N) will result in N inputs and this N can be in thousands. Halder *et al.* [9] reframed text categorization as a general 0/1 issue and processed the text and label as inputs using BERT. To improve transferability, the model predicts 1 if the label adequately represents the text and 0 otherwise. But managing a large number of labels can be difficult because of the exponential increase in complexity that occurs when processing both text and labels at the same time, especially in taxonomies that have hundreds or even thousands of labels.

A study by Bongiovanni *et al.* [3] suggested a way to categorize text completely without the need of labeled data, based on a fixed taxonomy structure. They used zero-shot to assign a prior similarity score for each taxonomy label based on the semantic information stored in the pre-trained Deep Learning Models, and then they used the hierarchical structure (also called Upwards Score Propagation concept) to support this prior belief. However, the experiment was performed considering only one node in each level leading to a path formed by 'N' nodes if it is a 'N' level taxonomy. Since each node is extracted based

on its highest prior score in each level, the obtained resulting path might not belong to the taxonomic. Talking about similarity complexity in this paper, it first computes the cosine similarity between labels and document embeddings by independently encoding N taxonomy labels and M documents. The complexity of the technique is O(N + M) since just the document text has to be encoded for every new document. Although there are cosine similarity calculations (O(N $\times$ M)), they take a very little amount of computing time when compared to deep model forward passes. On the other hand, O(N $\times$ M) complexity results from other state-of-the-art algorithms for zero-shot text categorization that demand to deliver each label with every document.

A new concept has been introduced by Sappaadla *et al.* [21] for multi-label zero-short text classification and has been carried out considering three approaches which are label presence, label word similarity, and semantic similarity. If the actual label name is present in the document, then the corresponding label is predicted to be true. Likewise, in case of extremely long label names, the label is predicted to be true if for a user-defined threshold t and a maximum window of size 'C', it is textually similar to the document. Here, the window is placed in both the document as well as label name which makes it a computationally expensive task. Also, the performance of label presence also depends on the usage of similarity function. Our proposed method tried to infuse several approaches that we have collected in literature reviews for zero shot multi label text classification and aims to do the following:

- We leverage the architecture for hierarchical multi-label text classification in place of hierarchical single-label text classification which was introduced by Bongiovanni *et al.* [3].

- We introduce top k nearest labels where each label is tagged up with its parent form for a corresponding document.

- Hierarchical information injection is not only propagated through Upward Score Propagation (USP), but also by the nature of path of each label, selecting top k full path labels for a given document.

- We leverage the concept of label presence and modified version of label word similarity in the model architecture before USP calculation.

- It predicts at different levels of hierarchy upto which it is confident, rather than selecting each node all k levels.

2.6 Embedding Models

It is not a good idea to choose the best Semantic Text Embedding (STE) model to utilize for our zero shot model based on the models' stated performance since all of the STE models that are currently available are primarily trained to capture the semantics of context-rich text. In addition, we need to compare their performance to context-rich texts in terms of encoding the meanings of short keywords and chained forms of short keywords. Therefore, the following embedding models will be used for this project which will be discussed below:

2.6.1 Instructor Embeddings

Su *et al.* [19] in their paper introduced INSTRUCTOR, an innovative method that creates text embeddings by appending task instructions to each text paragrapah. This encoder can generate domain-specific text embedding based on the instruction given rather than requiring further training. Due to this nature of instructor embeddings, it will be one of the models that will be used for embedding the documents, and labels in our zero shot settings. The model name is "hkunlp/instructor-large" and it has embedding dimension of 768.

2.6.2 mpnet-all

BERT uses Masked Language Modeling (MLM) for pretraining but it disregards the dependencies between projected tokens. However, XLNet model solves this issue using Permuted Language Modeling (PLM). Despite that, XLNet model has a position mismatch between pre-training and fine-tuning. The main reason for chosing MPNet is that it has a unique pre-training techniques that hides the drawbacks of XLNets and BERT while retaining their benefits [17]. The model name is "all-mpnet-base-v2" and it has embedding dimension of 768.

2.6.3 text-embedding-ada-002

OpenAI's text-ada-002, a well-known embedding model, can fit about 6,000 words into a 1,536-dimensional vector. This model is available only via an API call and each call incur API charges. Greene *et al.* [7] mentioned that

text-embedding-ada-002, which is priced 99.8 % less than Davinci, is a replacement for five different models for text search, text similarity, and code search which has even surpassesed Davinci, their previous most competent model, on most workloads.

2.6.4 nasa-smd-ibm-st

A Bi-encoder sentence transformer model called nasa-smd-ibm-st was developed by fine-tuning the nasa-smd-ibm-v0.1 encoder model. In addition to a domain-specific dataset of 2.6 million instances from documents selected by NASA Science Mission Directorate (SMD), it is trained on 271 million examples.The goal for this sentence transformer model is to improve natural language technology for NASA SMD NLP applications, like intelligent search and information retrieval. Since, our dataset is a part of NASA based earth science dataset, this model can also be considered for embedding purposes in our research.

2.7 Evaluation Metrics

The PRF (precision, recall, and F-score) metrics are commonly used for evaluating classification performance. These metrics will not be leveraged as they are not suitable for hierarchical text classification tasks, where wrong classification predictions could not be clearly discriminated with. There is also a widely-used hierarchical measure based on the notion of distance that overcomes this problem. However, it has some limitations. First, it is not easily extendable to DAG hierarchies (where multiple paths between two categories can exist) and multi-label

tasks. Second, it doesn't change with depth. For example: Misclassification into a sibling category of a top level node and misclassification into a sibling of the node 10-level deep are considered the same type of error (distance of 2). However, an error at the 10 th level seems a lot less harmful than an error at the top level.

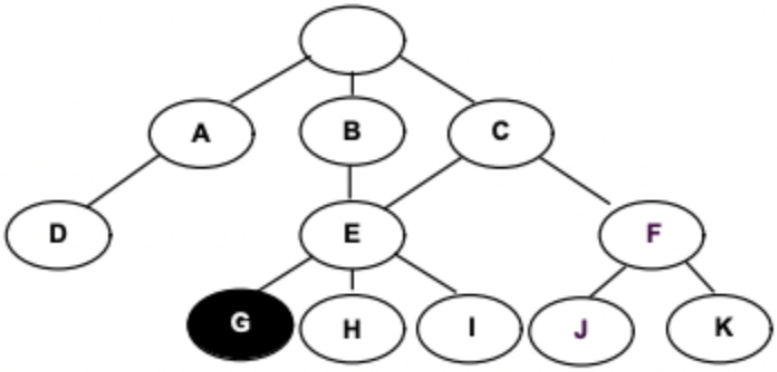


Figure 2.1: A sample tree hierarchy [10].

Svetlana *et al.* [10] in his paper formulated the following requirements to express the desired properties of a hierarchical evaluation measure (HM) which are as follows:

- The measure gives credit to partially correct classification, *e.g.* misclassification into node I with more common ancestral nodes should be considered less severe than misclassification into node D with less common ancestral node. This means, distant errors should be heavily reflected in performance metrics.

- The measure punishes errors at higher levels of a hierarchy more heavily, *e.g.* misclassification into node I when the category is its sibling G is less

severe than misclassification into node C when the correct category is its sibling A.

We will use the approach as suggested by Svetlana. The new measure is recall with the following additional: each example belongs not only to its class, but also to all ancestors of the class in a hierarchical graph. These new measure is known as hR (hierarchical recall).Also, in multi-label settings, for any instance (d_i,C_i) classified into subset C_i' we extend sets C_i and C_i' with the corresponding ancestor labels: $hR = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i}{\sum_i |\hat{C}_i|}$.

For example, suppose a document is classified into class F while it really belongs to class $Root--B--E--G$. To calculate our hierarchical measure, we extend the set of real classes $C_i = \{Root--B--E--GG\}$ with all ancestors of class in its true path G: $C'_i = \{B, E, G, Root\}$. We also extend the set of predicted classes $C'_i = \{Root--C--F\}$ with all ancestors of class F: $C''_i = \{C, F, Root\}$. Since we will be focusing on the top k predictions for our evaluation metrics, we consider recall@k as our evaluation metric. In this case, as our prediction will be carried out in hierarchical path, we only consider those ancestral nodes that falls under the path predicted.

Chapter 3. Datasets

3.1 Dataset Description

This dataset is scraped from earth science cmr query with sortlist hierarchical path with selected path from category to term and is scraped till variableLevel1. The category which is the root node is set to "Earth Science" for convenience. The total number of dataset is around 23988. The hierarchical path is in the following format: Category– Topic – Term – VariableLevel1- VariableLevel2 - VariableLevel3 which will be discussed as Level1 – Level2 – Level3 – Level4 – Level5 – Level6 in the following sections.

3.2 Label Proportion Analysis

Distinct labels in the given dataset: 1080 labels

GCMD labels formed by distinct nodes from the above labels: 1149

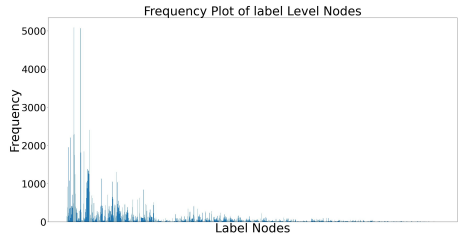
From Table 3.1, it is worth noting that the path to Level4 is representative of around 87% of the labels. This analysis will be useful in extracting top k similar labels based on similarity search approach.

Table 3.1: True Label Proporition Analysis of the data.

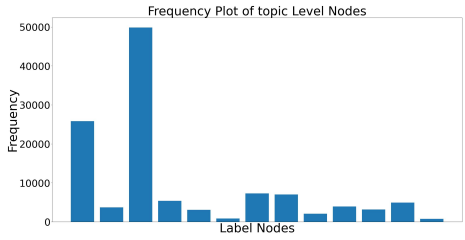
Level	**Percentage Representation**
$Level1$	0 %
$Level1 - -Level2$	16.53 %
$Level1 - -Level2 - -Level3$	12.18 %
$Level1 - -Level2 - -Level3 - -Level4$	87.05 %
$Level1 - -Level2 - -Level3 - -Level4 - -Level5$	0 %
$Level1 - -Level2 - -Level3 - -Level4 - -Level5 - -Level6$	0.599 %

3.3 Dataset Visualization

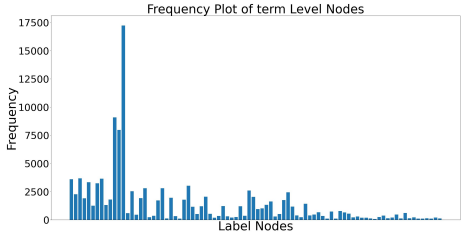
From Figure 3.1 we can see that the labels are distributed in a highly imbalanced way at different levels. This nature of data is called an extreme multi label text classification problem.



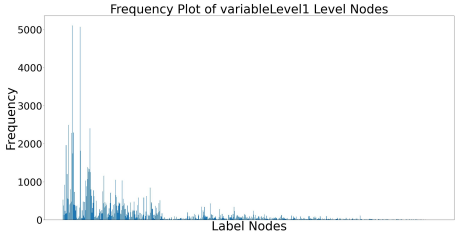
(a) Label Frequency Analysis.



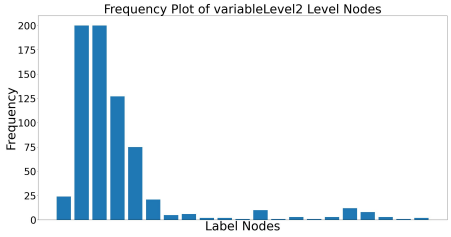
(b) Level2 Frequency Analysis.



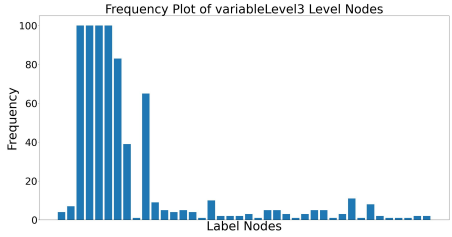
(c) Level3 Frequency Analysis.



(d) Level4 Frequency Analysis.



(e) Level5 Frequency Analysis.



(f) Level6 Frequency Analysis.

Figure 3.1: Frequency Analysis of the dataset across different levels of hierarchy.

Chapter 4. Method and Implementation

4.1 Terminologies

Before delving deeper into the methodology, lets introduce the following terminologies and concepts.

4.1.1 Top k Similar Label Paths

Our dataset contains 87% labels that belong to L1234, *i.e,* the depth of 4. A leaf node in a L1234 path contains its ancestral information as a part of L1234. This is why it is necessary to integrate the nearest neighbor search approach, the results of which will aid in hierarchical text classification before implementing a zero-shot approach. Below are the details of this approach: **Inputs:**

- Embedding of all the gcmd labels in L1234 form
- Embedding of document

  Process: cosine similarity between embedding of labels and documents

  Outputs: top k similar L1234 labels

4.1.2 Zero-shot Semantic Text Classification (Z-STC)

In this approach, we calculate cosine similarity of distinct nodes of all possible GCMD paths in each node with the document embedding rather than considering all L1234 paths. So, it is a method that computes the initial node scores in the taxonomy. This approach does not consider the hierarchical structure of the taxonomy of the labels.

It involves using a text encoder Ψ that is based on Semantic Text Embedding (STE) to map the text of a document d and a taxonomy label l separately into the same semantic vector space. From there, a initial relevance score $P(l)$ can be assigned by comparing their cosine similarity:

$$p_d(l) = \mathrm{Sc}(\Psi_D(d), \Psi_L(l)), \tag{4.1}$$

$$\mathrm{Sc}(A, B) = \frac{A \cdot B}{|A| \cdot |B|}, \tag{4.2}$$

where the closer $p(l)$ is to 1, the more confidently the given document D can be assigned to the label l.

4.1.3 Relevance Threshold (α)

We have function called S_{USP} which will be discussed in the next section. Relevance threshold can be defined as the minimum relevance score of a distinct node that indicates a high likelihood that the given node might actually represents the given document. In the taxonomy tree, every node has a relevance threshold

(α). It is calculated by taking into account the statistical distribution of prior Z-STC scores of each node for all nodes over a set of fixed Earth science unrelated Wikipedia articles.

4.1.3.1 Statistical Relevance for α

When a number significantly deviates from a particular Ground Distribution, it is sometimes referred to as highly significant in statistics. In our scenario, the distribution of a label l's scores $p_{GD}(l)$ over a set of irrelevant documents is referred to as its ground distribution. Based on this, we deduce that a node l is associated with a new document d if its similarity score with the document $S_{USP}(l)$ is statistically higher than its Ground Distribution.

By calculating the similarity scores $p_{GD}(l)$ of unique nodes l with more than 1000 randomly chosen Wikipedia articles, the ground distribution of irrelevant documents is produced. The Ground Distribution is meant to be calculated across a collection of documents that have nothing to do with the labels in the taxonomy we are utilizing. As Figure 4.1 shows, we set α_l to surpass 95% (2σ for Gaussian distribution) of the Ground Distribution, as is customary for statistical significance. Any value $p_d(l) > \alpha_l$ for a particular document indicates that the label l is very significant.

The ground distribution of label relevance scores over a thousand randomly crawled Wikipedia articles is represented by the blue histogram in Figure 4.1, and its fit with a log-normal distribution is indicated by the yellow line. The Relevance Threshold, or α for a given label, is the value that is larger than 95% of the Ground

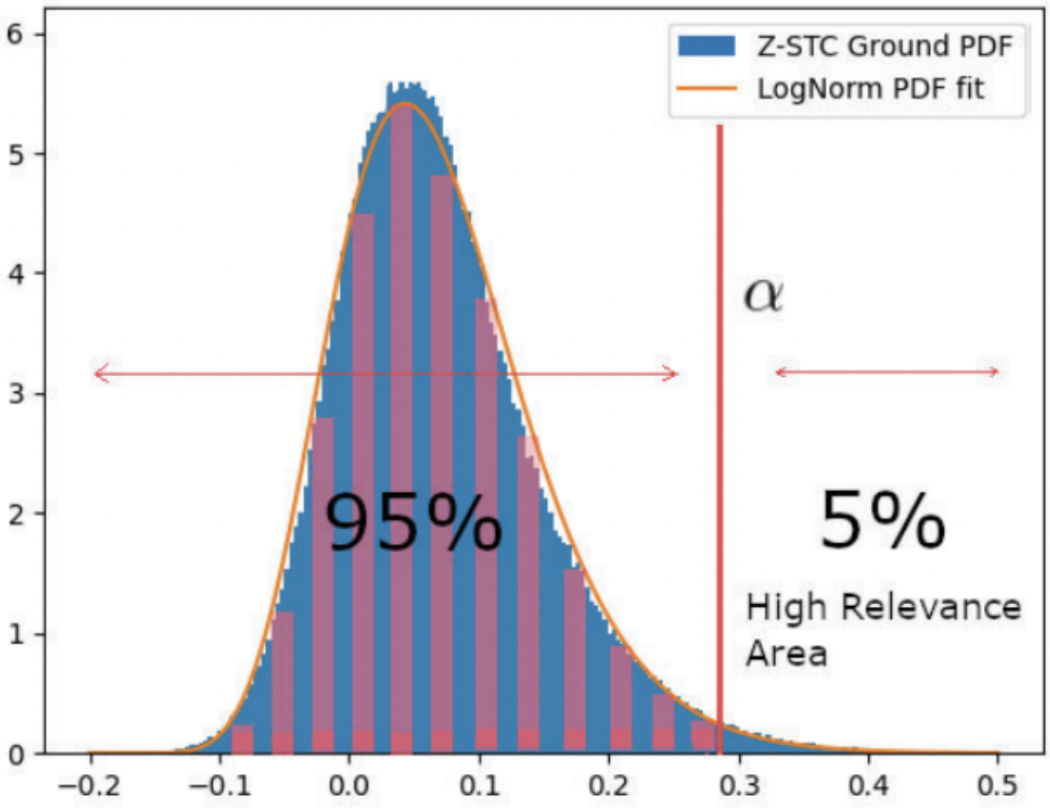


Figure 4.1: Ground Distribution of label relevance scores of 1000 Wikipedia articles [3].

Distribution. The ground distribution of unrelated Wikipedia pages is modeled using the Log-Normal Distribution as the probability distribution function.

4.1.4 Gold and Silver Nodes

Checking if the label name exists in the document is the easiest way to forecast a label 'l' as relevant given a document 'd'. As our label path contains ancestral information, we implement this concept in a distinct node basis. For

a set of nodes, if the any node is present in the document, it is categorized as gold node. While not every node is precisely present in the document, words that are related to one another often have the same meaning. For instance, the document's term "radiation" and the node "radioactive" have the same meaning. These words fall under the category of silver nodes, and they are selected only when the document contains 85% of the node substring. Let us talk about the advantage of the this concept with an example: Suppose we have two labels:

"EARTH SCIENCE>>ATMOSPHERE >>ATMOSPHERIC PRESSURE"

"EARTH SCIENCE>>OCEANS>>OCEAN TEMPERATURE>>RADIOACTIVE"

And the document is: *"Ocean temperature, intimately linked with atmospheric conditions, shapes climate patterns and influences global circulation. Furthermore, the presence of radioactive compounds in the ocean environment has impacted the heat distribution."*

Based on the definition of gold and silver nodes, we have the following sets of gold and silver nodes:

Gold nodes: [RADIOACTIVE, OCEAN TEMPERATURE]

Silver nodes: [ATMOSPHERE, OCEANS]

Since some gold and silver nodes are reflective of particular levels in a potential hierarchical route, this method of extracting gold and silver nodes by splitting the possible hierarchical path into separate nodes is crucial to the hierarchical text categorization process. When propagating scores from the leaf node to the ancestral node—which will be covered in the following section—it was able to extract nodes in Levels 2 and 4, which is a good representation for our scenario.

4.1.5 Upwards Score Propagation (USP)

This approach was first presented in the baseline paper that we are following. Here, the taxonomy's hierarchical structure, relevance threshold, and similarity scores between each node and the provided documents are utilized to propagate the confidence scores from the lowest level, or leaf nodes, up the hierarchy. Each node in the tree currently has two sets of scores: a relevance threshold score and a similarity score.

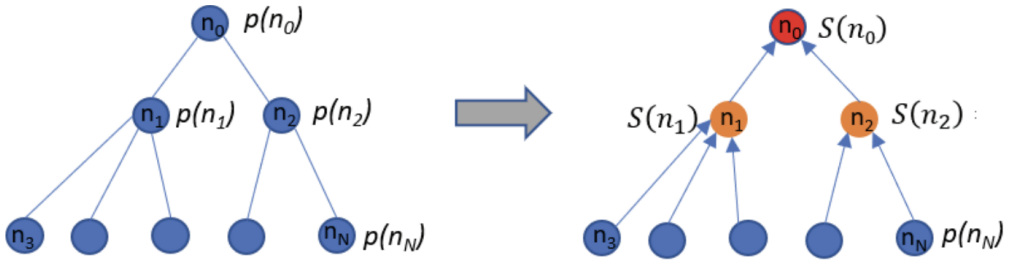


Figure 4.2: Upwards Score Propagation Concept [3].

The left side of the above figure shows the initial condition of the tree where each node has its similarity score and relevance threshold score. The right side of the figure shows the score after USP is implemented where score for each node is propagated upwards to its ancestor based on the equation (4.3) that is presented below:

$$S_{USP}^{(i)}(l) = \begin{cases} S_l^{(i-1)} & \text{if } S_{c_i} \leq S_l^{(i-1)} \\ S_l^{(i-1)} \cdot e^{(S_{c_i} - S_l^{(i-1)})} & \text{if } S_l^{(i-1)} l \leq S_{c_i} \leq \alpha_c \\ S_{c_i} & \text{if } S_{c_i} \geq S_l^{(i-1)}, \end{cases} \tag{4.3}$$

$$S_l^{(0)} = \max(0, p(l)), \qquad S_{USP}(l) = S_l^{(N)}. \tag{4.4}$$

Here, $S_{USP}(l)$, the final posterior score for the label l, is obtained after accounting for all N children of c. In order to guarantee the convergence of S_{USP}, $p(l)$ negative values are shifted into 0. The initial score for each label is $S_l^{(0)} = \max(0, p(l))$. This is supported by the observation that semantic similarity is expressed by values of $p(l)$ close to 1, whereas dissimilarity is transmitted by oscillations around the value $p(l) = 0$, as can be seen from the shape of the distribution of label similarity over unrelated texts. This makes it possible to repeatedly apply the Upwards Score Propagation process to any taxonomic level. The equation's full explanation is provided below:

- If a child's similarity score is greater than both its relevance threshold and parent's similarity score, its score gets propagated to the parent node.
- If a child's similarity score is greater than its parent's similarity score but its score cannot beat its relevance threshold, then the parent node gets boosted by e^{Δ} where Δ represents the difference between the similarity score of child node and its parent node.

- If the child's similarity score is lower than its parent's similarity score, then the parent's similarity score remains as it is. The whole concept that USP works on is that if a child node is relevant for a document, then its parent node must be as well.

4.2 Proposed Methodology

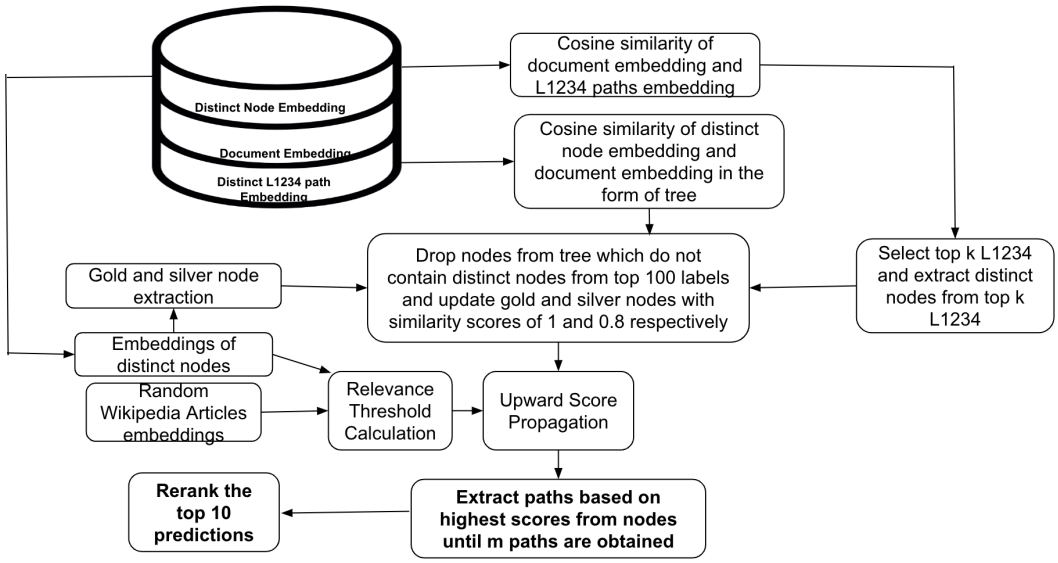


Figure 4.3: Flow chart of proposed multi-label zero shot text classification problem.

Figure 4.3 shows the flowchart of methodology implemented for the proposed zero shot text classification. However, a slight tweak in the architecture of the methodology will be carried out to experiment on different settings. Based on this, different experiments will be discussed below. However, before this, let us discuss few procedures which are the same in different experiments.

4.2.1 Relevance Threshold Calculation

Relevance threshold calculation is carried out considering random 1000 Wikipedia articles and all the possible nodes using same embedding model in all setting. This score will be constant in all experiments.

4.2.2 Cosine Similarity of Document and Node Embeddings

Experiments will be carried out considering different embedding models that was introduced in the literature review. First of all, the we extract embeddings of distinct nodes and embedding of document, and cosine similarity score between document and nodes is calculated based on those embeddings.

4.2.3 Gold and Silver Nodes Extraction

After this gold and siver nodes is found for a document and its similarity is updated to 1 in case of gold node and 0.8 if its a silver node.

4.2.4 Variations in the Experiments

4.2.4.1 Top k Similar Paths

This is the most baseline version of the experiment after tencent's NeuralNLP-NeuralClassifier. In this experiment, cosine similarity of distinct L1234 labels its full hierarchical form is carried out with document using different embedding models.

4.2.4.2 Baseline Zero Shot Model

The baseline model is the model introduced by Bongiovanni *et al.,* (2023). Since the paper only extracts one path considering all possible levels, the difference lies in the way how the multi labels are extracted.

4.2.4.3 Baseline Zero Shot Model with Gold Silver Nodes

In this case, the only addition in the baseline zero shot model is introduction of gold and silver nodes. Here, the prior scores calculation are updated with values 1 and 0.8 for gold and silver nodes respectively. Only then USP is carried out.

4.2.4.4 Proposed Zero Shot Model

The proposed zero shot model is some modification on the baseline zero shot model. Here, since our label full path also contains ancestral information in it. The goal is to leverage this fact and use L1234 which is representative of more than 87% of true labels in the overall dataset. Based on top k similar L1234 labels which gives satisfiable recall @ k, the tree formed by experiment introduced in 4.2.2, all the nodes that fall under L1234 and its potential L5,L6 are extracted and those nodes are only considered as representative nodes of the tree. This model introduces the concept of bulb of tree, where all those nodes that fall under L1234, and its deeper nodes are only considered and the rest of the nodes are removed from the tree. This introduces the concept of dynamic

tree. After, this gold and silver nodes in the filtered tree are updated with the values 1 and 0.8 respectively and then USP is carried out.

4.2.4.5 Proposed Zero Shot Model without Gold Silver Nodes

This model architecture is the same as that of proposed zero shot model. However, the gold and silver nodes' score are not updated.

4.2.4.6 Baseline Supervised Model

This model was proposed by Liu *et al.* [12] and is an open source toolkit for neural hierarchical multi-label text classification where the taxonomy is organized in the form of a tree or DAG. The instances are multi-labelled during training and testing. The architecture of the NeuralClassifier is given below:

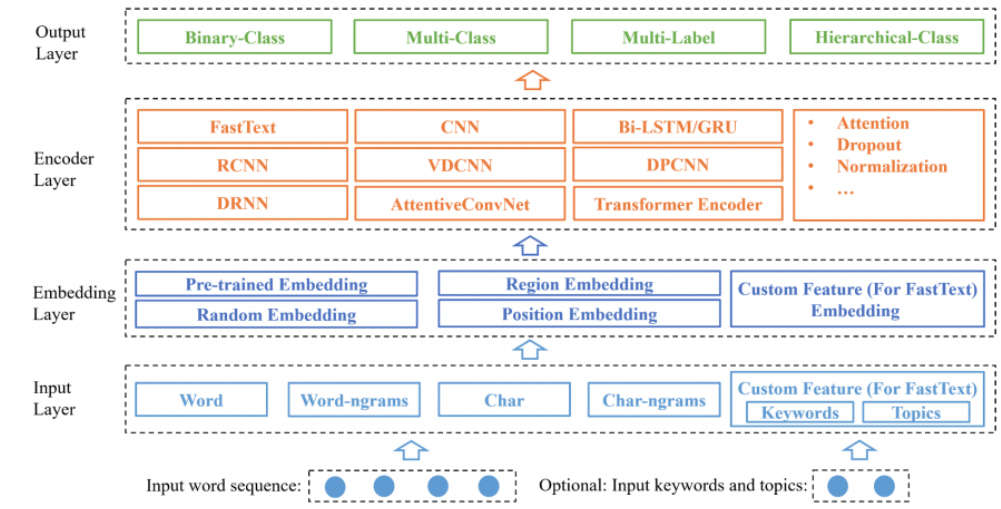


Figure 4.4: Architecture of Neural Classifier [12].

In case of input layer, the given input text sequence will be processed. A sample of input data is given in the figure below: In our case, our document

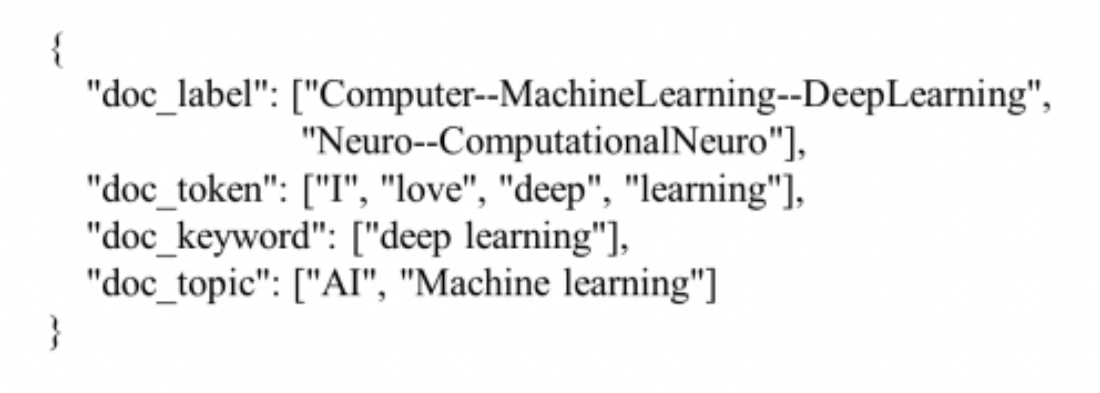


Figure 4.5: Input Sample for Neural Classifier [12].

content will be passed as doc_token, and our potential labels will be passed on doc_label. We will not consider doc_keyword, and doc_topic in our case. Likewise region based embedding has been chosen in the embedding layer star-transformer encoder is used as encoder layer. Similarly, BCELoss has been used for multi-label classification and a recursive regularization has been added for hierarchical classification. The objective of this regularization framework is to make sure that the parent and child share the similar model parameters.

Configurations:

- train dataset: 14392
- test dataset: 4798
- validation dataset: 4798

The analysis with the proposed zero shot model will be carried out considering test dataset only.

4.2.5 Algorithm for Extracting Top 10 Label Paths

Since we will be focusing on recall@k and our maximum k value is 10, we will be extracting top 10 label paths. We will start with the highest score and its corresponding nodes from the list of posterior scores and try to formulate the path until we get the top 10 paths. While extracting a list of paths, if one path is substring of another path, the first path will be removed, and this step will be carried out until we extract the top 10 labels. This is done to obtain the deepest label of the path rather than its sub forms. Example: extracting a >> b >> c >> d >> e, is enough than extracting a, a >> b, a >> b >> c and a >> b >> c >> d as one of the labels in the top 10 labels. The detailed steps for extracting top k label paths are discussed below:

- Create two lists, one for putting extracted paths called **"paths"**, and other for keeping track of distinct nodes called **"dist_nodes"**. Initially, both are empty.
- Find the highest score from the tree, extract all the labels in those nodes with highest score and put it in dist_nodes.
- Try to form a path of formed dist_nodes from the given taxonomy.
- Until the length of formed paths is not 10, repeat the above steps by considering next highest scores.

Note: In the process of tracking paths, if one path is substring of another path, remove the previous one.

For example, let us consider a tree with posterior scores which is obtained after USP as shown in the Figure 4.3.

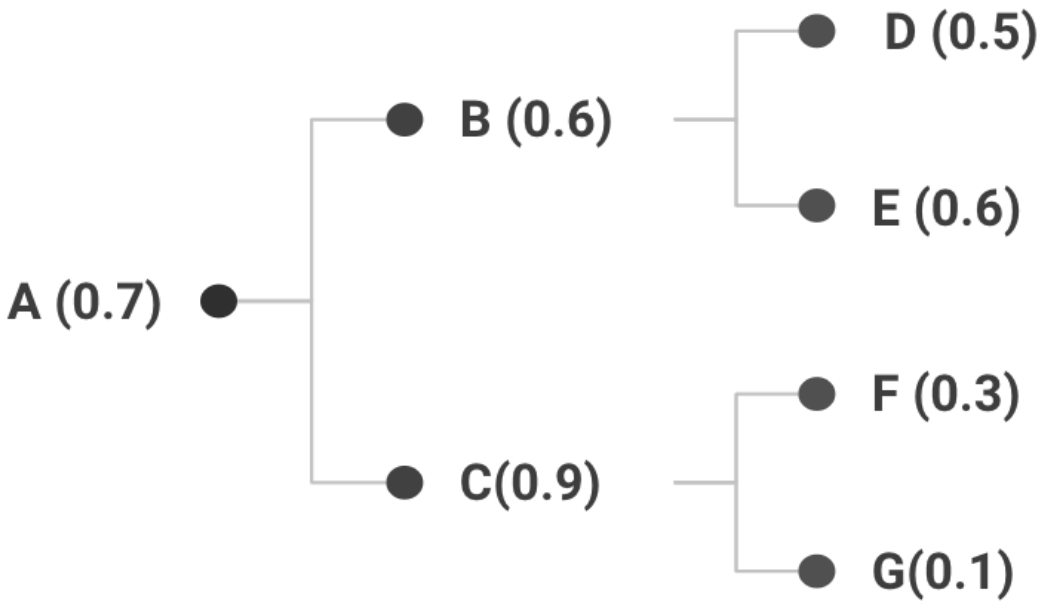


Figure 4.6: Tree Structure containing posterior scores in each node.

Algorithm

- `score_list` $= [0.5, 0.6, 0.9, 0.5, 0.6, 0.7, 0.1]$
- `Paths, dist_nodes` $= []$ *(Initial condition)*

Find the highest score in `score_list`:

- `highest_score` $= 0.9$, `dist_nodes` $= [C]$, `paths` $= []$

  Find the next highest score:

- `Next highest_score` $= 0.7$, `dist_nodes` $= [A, C]$, `paths` $= [A \rightarrow C]$

- `Next highest_score` $= 0.6$, `dist_nodes` $= [A, C, B, E]$, `paths` $= [A \rightarrow C, A \rightarrow B \rightarrow E]$

  Note: In this step, the path $A \rightarrow B$ is already a substring of $A \rightarrow B \rightarrow E$, so it won't be added. If it was already in the paths, it will be removed.

  Find the next highest score:

- `Next highest_score` $= 0.5$, `dist_nodes` $= [A, C, B, E, D]$, `paths` $= [A \rightarrow C, A \rightarrow B \rightarrow E, A \rightarrow B \rightarrow D]$

  Repeat the above steps until k paths are extracted.

4.2.6 Reranking Model

Since, we will already have top 10 predictions in its path form, the reranking will be done based on that embedding model which will outperform rest of the models in the experiment carried out on top k similar labels.

4.2.7 Analysis of Role of Gold and Silver Nodes on Correct Prediction of Nodes

A detailed analysis on how the presence of gold and silver node influences the hierarchical path prediction will be carried out through correlation analysis and visualization.

4.2.8 Analysis of Performance of Model Across Similarity Thresholds for Silver Nodes

Since we have fixed the similarity threshold to be 0.85 for silver nodes extraction, our next step will be to analyze how the recommended model performs when silver node extracted considering different similarity thresholds. We will experiment on threshold values ranging from 0.45 to 0.9 with a difference of 0.05.

4.2.9 Analysis of User Defined k Value for Top k Paths

After introducing the recommended model, we will study how the user-defined k values lead to the depth of the tree and how this k value influences the performance of the result. Since the average number of labels per dataset is 5, we will experiment for k value ranging from 3 to 15.

Chapter 5. Experiments and Results

5.1 Top k Similar Paths

Table 5.1: recall@k score for extracted top 100 similar L1234 across different embedding models and supervised baseline model.

k	nasa	mpnet-all	text-ada-embedding	instructor	NeuralClassifier
1	0.1776	0.1781	0.3206	0.1878	0.3608
3	0.2236	0.2190	0.4167	0.2319	0.481
4	0.2419	0.2374	0.4462	0.2486	0.509
5	0.2578	0.2543	0.4682	0.2645	0.526
10	0.3180	0.3212	0.5374	0.3225	0.563
15	0.3580	0.3652	0.5799	0.3601	0.573
20	0.3849	0.3980	0.6119	0.3898	0.575
30	0.4250	0.4475	0.6695	0.4361	0.576
50	0.4826	0.5072	0.7321	0.4921	0.576
100	0.5615	0.5931	0.8170	0.5748	0.5761

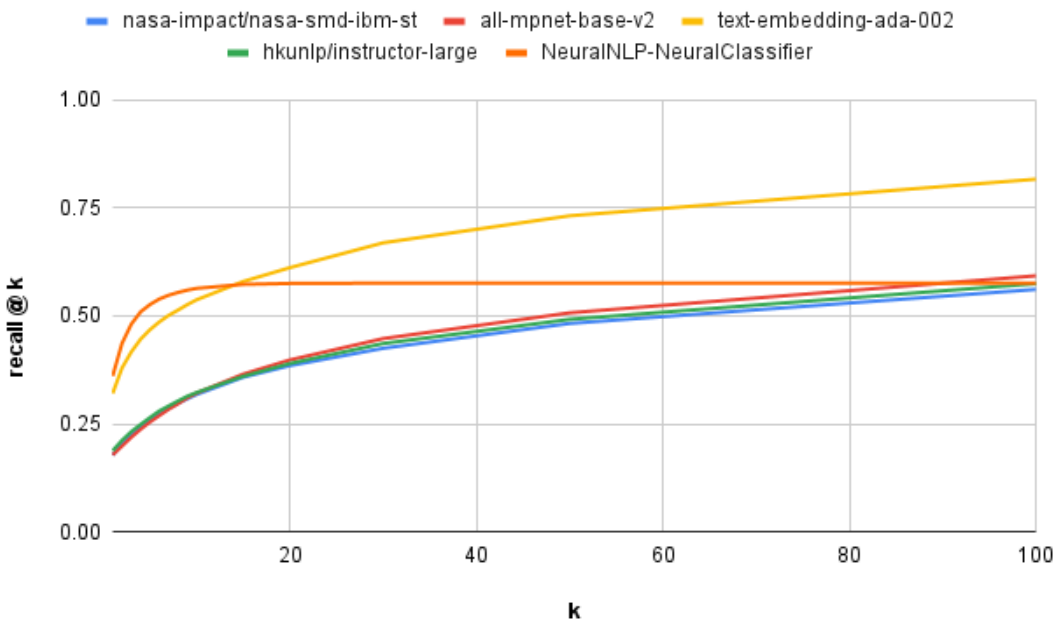


Figure 5.1: recall@k score for extracted top 100 similar L1234 across different embedding models and supervised baseline model.

From the Figure 5.1 and Table 5.1, we can see that text-embedding-ada-002 outperforms all the other embedding models as well as NeuralNLP-NeuralClassifier at every k values greater than or equal to 15 when similarity search was carried out considering ancestral node in the label of child node that belongs to Level6. The performance of the "nasa-smd-ibm-st" and "instructor-large" models was found to be comparable, with "instructor-large" slightly outperforming "nasa-smd-ibm-st" by a small margin. Another thing evident from this experiment is that the "text-embedding-ada-002" model is able to capture the labels with its ancestral information better than the other models. This fact is evident in the recall @ 100, which is measured at 0.8169, demonstrating supe-

rior performance compared to all other models. Even though our proposed model contains embedding to be done by the same model, the outperforming nature of text-embedinng-ada-002 in this experiment will also motivate us to experiment on hybrid models.

5.2 Experiments on Different Embedding Models

5.2.1 text-embedding-ada-002

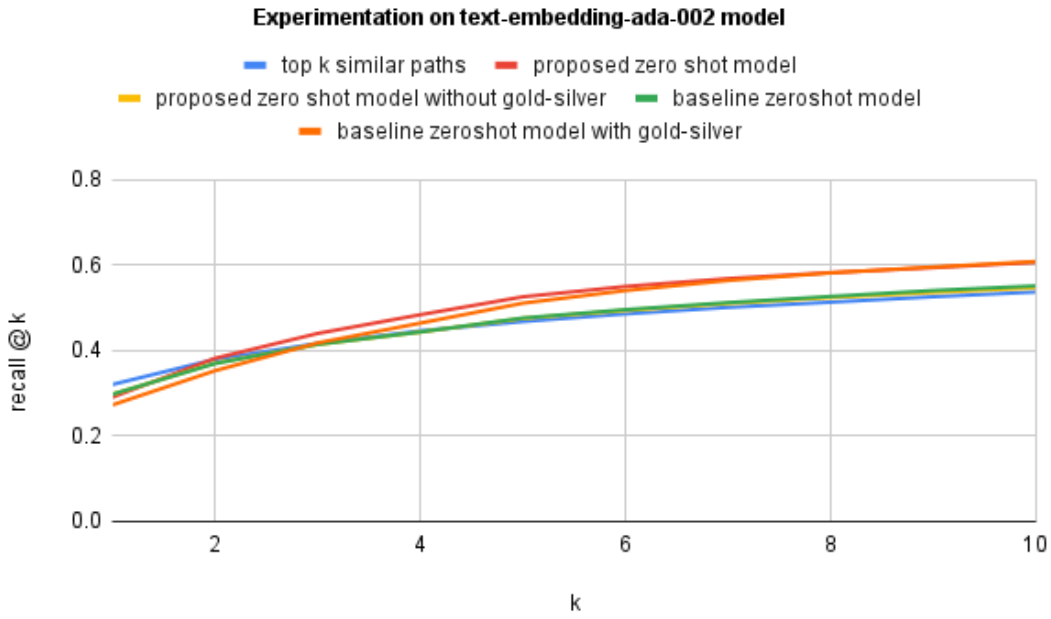


Figure 5.2: Experimentation on different versions of proposed zero shot model using text-ada-002 model.

From Figure 5.2, we can say that our proposed model using text-embedding-ada-002 outperforms the baseline zero shot model by a significant margin. Likewise, it is apparent that the concept of gold and silver nodes has played a signif-

icant role in our proposed zero shot model. Furthermore, there is no significant impact observed from the top k similar paths concept in the proposed zero-shot model. This becomes evident when experimenting without gold-silver, in comparison with the baseline zero-shot model. An experiment with baseline zero shot model which only considers gold and silver in its USP verifies the uselessness of top k similar paths in the text-ada-002 embedding model.

5.2.2 nasa-smd-ibm-st model

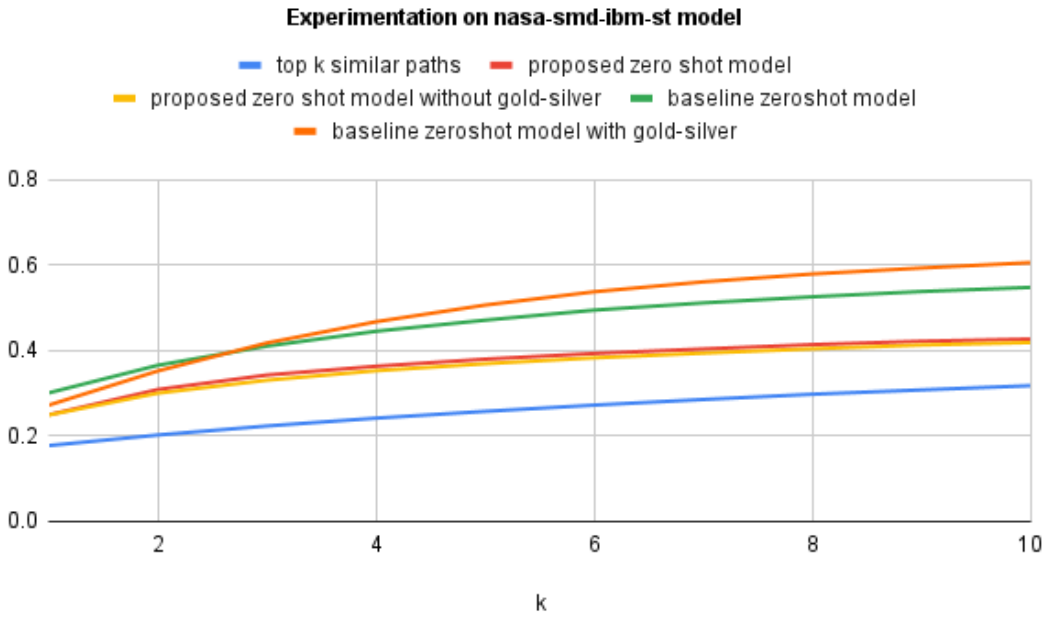


Figure 5.3: Experimentation on different versions of proposed zero shot model using nasa-smd-ibm-st model.

In the case of the nasa-smd-ibm-st model, the baseline zero-shot model with gold-silver labels easily surpasses the proposed model. Another noteworthy

observation is a significant difference between the proposed zero-shot model and the baseline zero-shot model with gold-silver labels, with recall @ k values of 0.41 and 0.60, respectively. This discrepancy is acceptable, given that the recall @ 10 for top k similar paths was found to be 0.56, compared to around 0.81 in the case of the text-embedding-ada model.

5.2.3 all-mpnet-base-v2 model

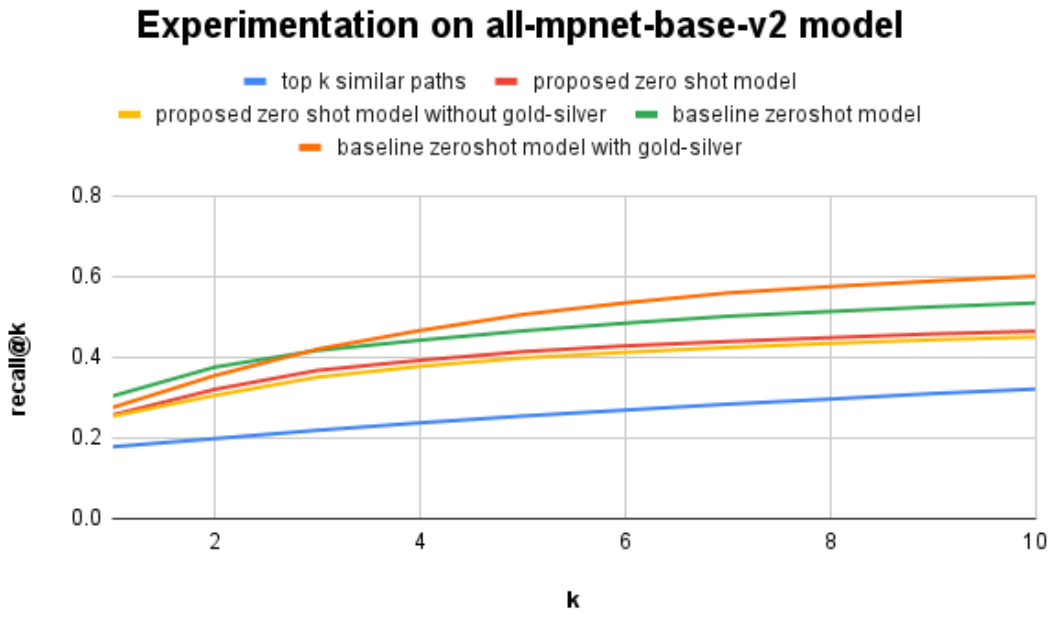


Figure 5.4: Experiment on different versions of proposed zero shot model using all-mpnet-base-v2 model.

In this case, baseline zero shot model with gold and silver nodes is found to outperform the proposed zero shot model. This is pretty evident with lower recall

@ 100 for top 100 similar paths. Here, top k similar paths is found to negatively influence the proposed zero shot model.

5.2.4 hkunlp/instructor-large

The result for instructor embeddings can be seen in Figure 5.5.

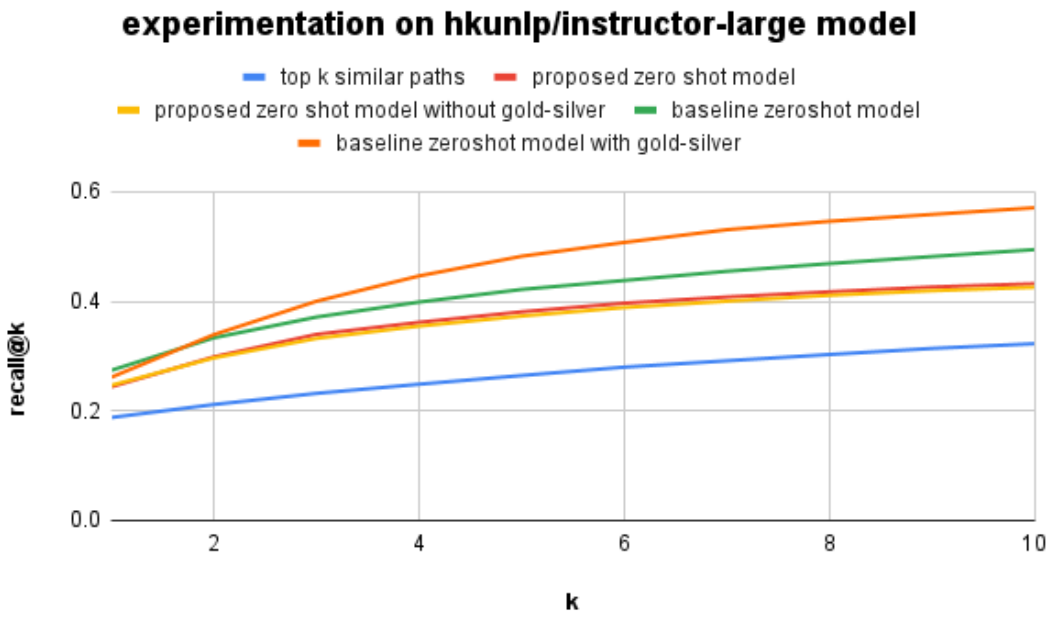


Figure 5.5: Experiment on different versions of proposed zero shot model using hkunlp/instructor-large model.

The baseline zeroshot model with gold and silver labels, and baseline zeroshot model outperform the different versions of proposed zeroshot model. One thing worth noting in case of recall@k values for proposed zero shot model and proposed zero shot model without gold-silver is that, the obtained top 100 values were not noticeably representative of gold and silver labels, that is why the re-

call@k values in these two cases were not so different. This fact shows that even though recall@k scores for instructor-large model and nasa-smd-imb-st in the case of top_k similar paths were found to be similar, the label's path which includes ancestral information is not truly representative of the gold and silver in case of instructor-large model.

5.3 Comparison of Different Models

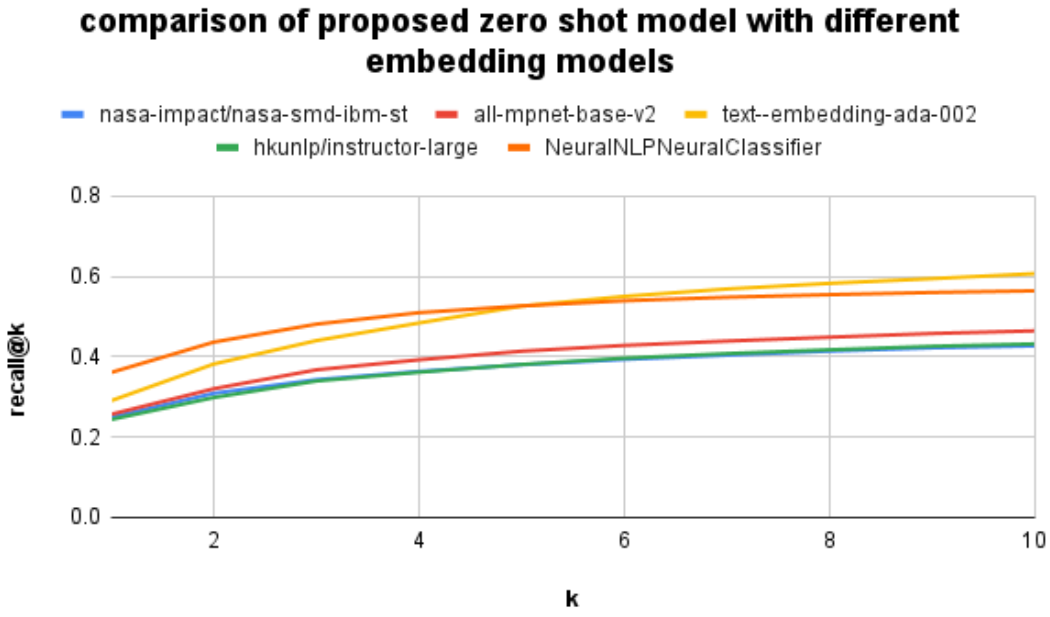


Figure 5.6: Comparison of proposed zero shot model with different embedding models.

From the above figure, we can see that the text-embedding-ada-002 model outperforms all the embedding models in the case of the proposed zero shot model with k¿=6. The performance of nasa-smd-ibm-st and instructor-large was found to be in the same range.

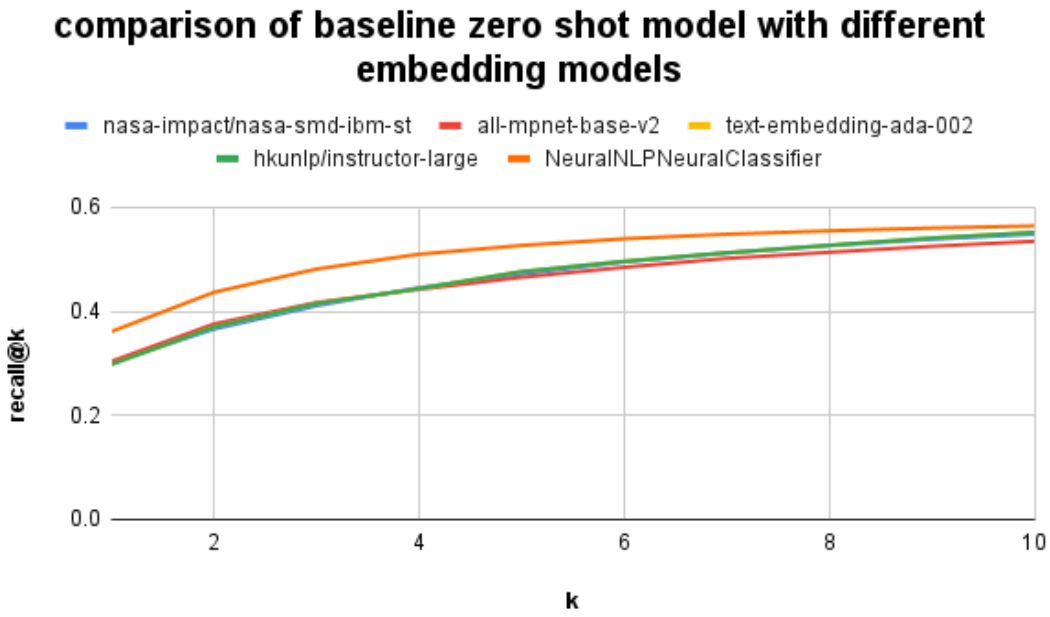


Figure 5.7: Comparison of baseline zero shot model with different embedding models.

From Figure 5.7, we can see that the performance of all the embedding models were found to be similar in case of baseline zero shot model.

Likewise, from Figure 5.8 we can see that recall @k values for baseline zero shot model integrated with gold and silver labels were found to be the same in case of all-mpnet-base-v2, nasa-smd-ibm-st, and text-ada-embedding-002 where text-ada-embedding-002 model outperforms these two at recall@10 by very small margin. Even though the recall@10 value for instructor-large model increased from 0.490 to 0.57 on baseline zero shot model when its integrated with gold and silver labels, it's still lagging with other embedding models with the same k value.

From Figure 5.9 we can see that the reranked result of baseline zero shot model integrated with gold-silver labels shows improved recall@k values

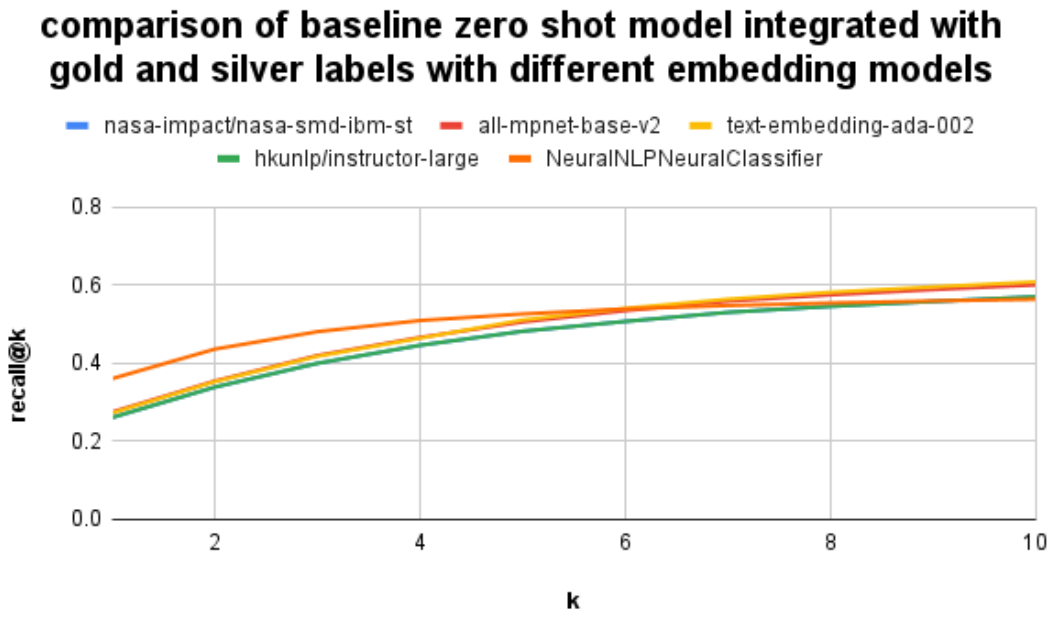


Figure 5.8: Comparison of baseline zero shot model integrated with gold and silver labels with different embedding models.

with k¡=5. This shows that the reranking approach implemented using text-ada-002 model is performing better as expected. Likewise, with the increasing value of k, reranked result and its baseline form is able to beat the NeuralNLP-NeuralClassifier model in terms of performance. One of the major reasons in which NeuralNLP-NeuralClassifier was able to achieve good results in lower k values might be those labels which are frequently used where the average number of labels in the dataset was about 5. When it comes to increasing k values, it might not be able to capture those labels which are highly imbalanced, in which the zero shot model is found to perform better. Validation of this hypothesis will be carried out in the next section.

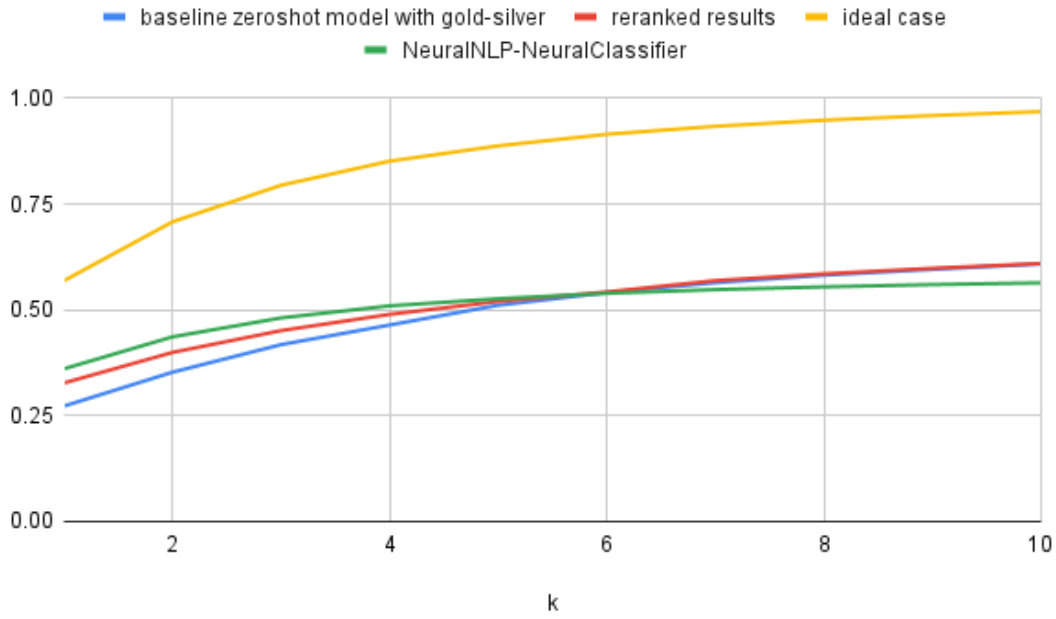


Figure 5.9: Comparison of reranked result with recommended model, ideal case, and NeuralNLP-NeuralClassifier.

5.4 Analysis of Depth of Labels Across Different Models

Figure 5.10 illustrates that the majority of true label paths are representative up to Level 4, followed by Level 3, and only a few labels extend to Level 6. Similar observations can be observed in the case of predictions made by the NeuralNLP-NeuralClassifier model. This fact shows that NeuralNLP-NeuralClassifier even though its able to leverage hierarchical information in its multi label text classification, it is only able to make predictions on those labels which are provided as a part of training data, rather than sets of potential labels that can be formed using the taxonomic hierarchy. Due to this reason, if the

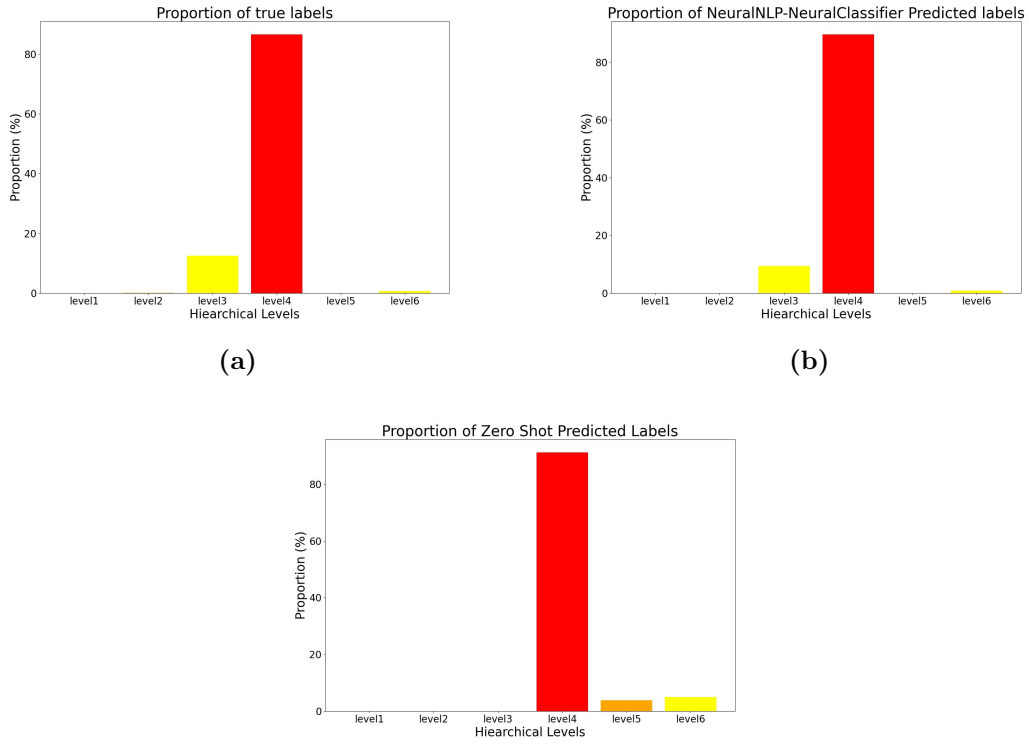


Figure 5.10: Proportion Analysis of labels in terms of hierarchical level across different models.

NeuralNLP-Classifier is not able to predict till node in Level6 when it should be Level6, it cant confidently predict it till Level5 node in its path. This drawback is solved using our proposed zero-shot model. It predicts the label upto which it is confident in hierarchically. This result can be observed in Figure 5.10 (c). The reason the zero shot model is not able to predict distinct paths till Level1, Level2, and Level3 is due to the fact of how our algorithm works. If a predicted path L123 is a substring of another predicted path L1234 while extracting top 10 labels, then the first one is removed by prioritizing to extract one more potential

path. The objective is to extract the path at its deepest level whenever possible. This analysis aligns with one of our goals, which is to extract labels belonging to the document up to the depth at which the model expresses confidence.

5.5 Analysis of Influence of Gold and Silver Nodes on Correct Predictions

The analysis conducted on the relationship between the presence of gold and silver nodes within datasets and the accuracy of predictions reveals insightful correlations. The implementation involves considering datasets with varying compositions of gold and silver nodes, with a threshold of 0.85 utilized for extracting silver nodes. The mere presence of number of gold and silver nodes does not linearly translate to higher prediction accuracy. This is also due to the fact that a single gold node can influence the whole node of the single true label and 10 gold nodes can also do the same if its a single label. Likewise, 4 gold nodes can form a single true label path and still give the same result. So, correlation of the number of gold and silver nodes and the proportion of correct prediction of nodes will not give the information that we desire. This is why focus is given on the proportion of correctly predicted nodes in the presence of correctly predicted gold/silver nodes, as it provides a more meaningful metric of model performance. We have encoded ancestral information of gold and silver nodes in our list of gold and silver nodes as its going to influence the score when USP is carried out. The correlation is carried out considering proportion of correct prediction of

gold/silver/gold-silver nodes and the proportion of correctly predicted nodes for that data.

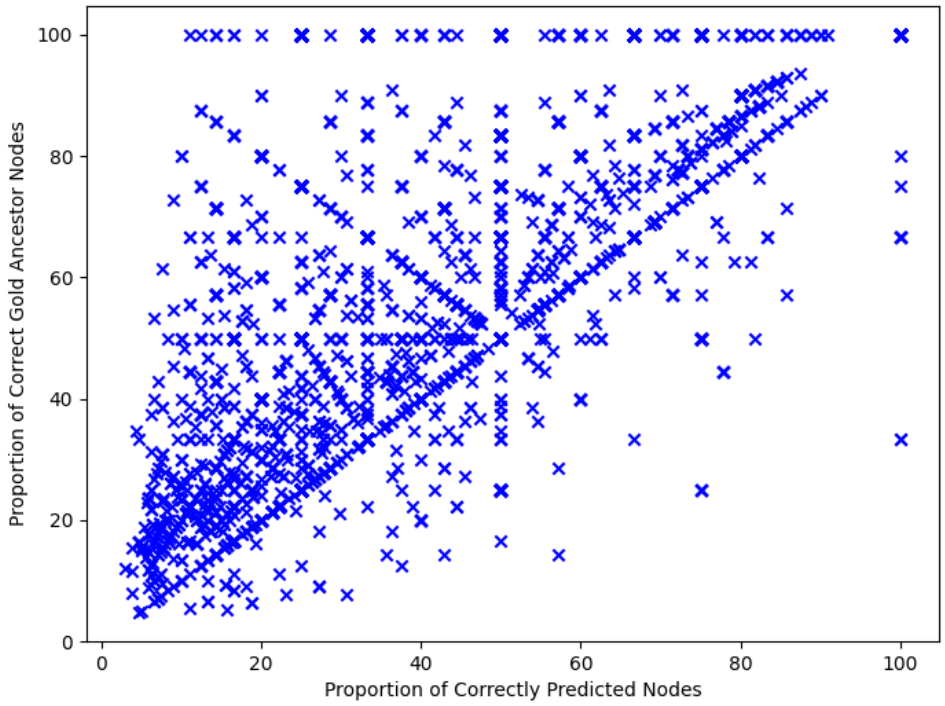


Figure 5.11: Plot of Proportion of Correct Gold Ancestor Nodes and Proportion of Correctly Predicted Nodes.

Gold Nodes Analysis: For this analysis, we have only considered those datasets which have at least one gold node. The correlation coefficient of 0.729 suggests a positive correlation between the presence of correct gold nodes and the proportion of correctly predicted nodes. The average proportion of correctly predicted nodes stands at 62.57%. This observation indicates that while the presence of gold nodes may offer some predictive value, it does necessarily guarantee higher

prediction accuracy. But this exceeds the recall@10 value of the recommended model and baseline zeroshot model for the whole dataset which was around 0.60 and 0.55 respectively.

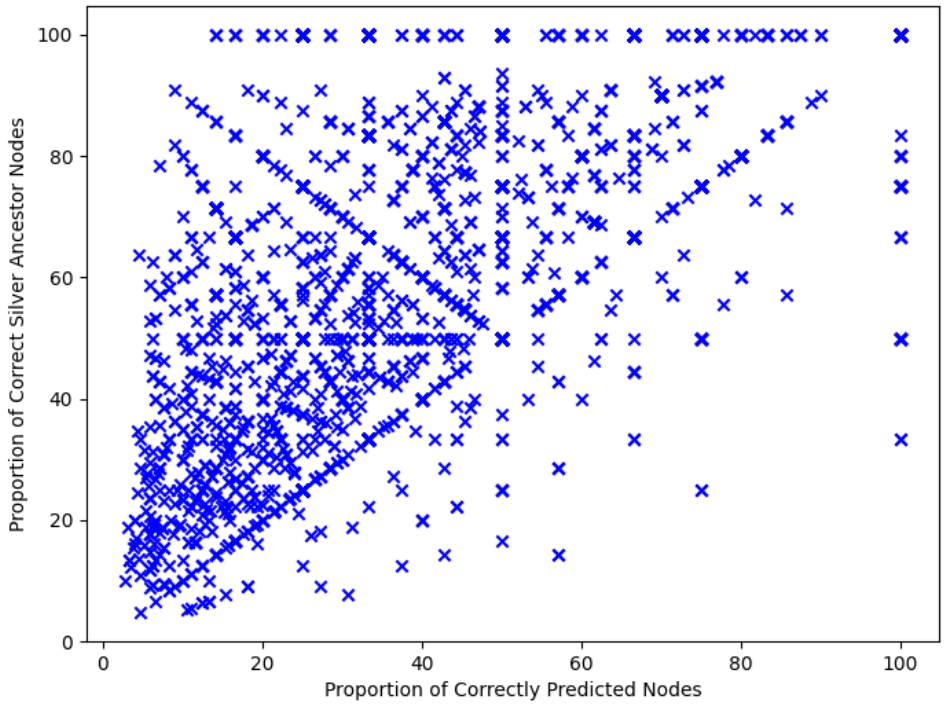


Figure 5.12: Plot of Proportion of Correct Silver Ancestor Nodes and Proportion of Correctly Predicted Nodes.

Silver Nodes Analysis: For this analysis, we have only considered those datasets which have at least one silver node. In contrast, the correlation coefficient for silver nodes is lower at 0.612, indicating a positive relationship with prediction accuracy but gold nodes are found to me more correlated than the silver nodes. However, the average proportion of correctly predicted nodes is higher at 65.5%.

This finding suggests that despite the weaker correlation than gold nodes, the presence of silver nodes contributes more consistently to accurate predictions. One thing worth noting is that the dataset obtained by filtering gold node and dataset obtained by filtering silver nodes are different.

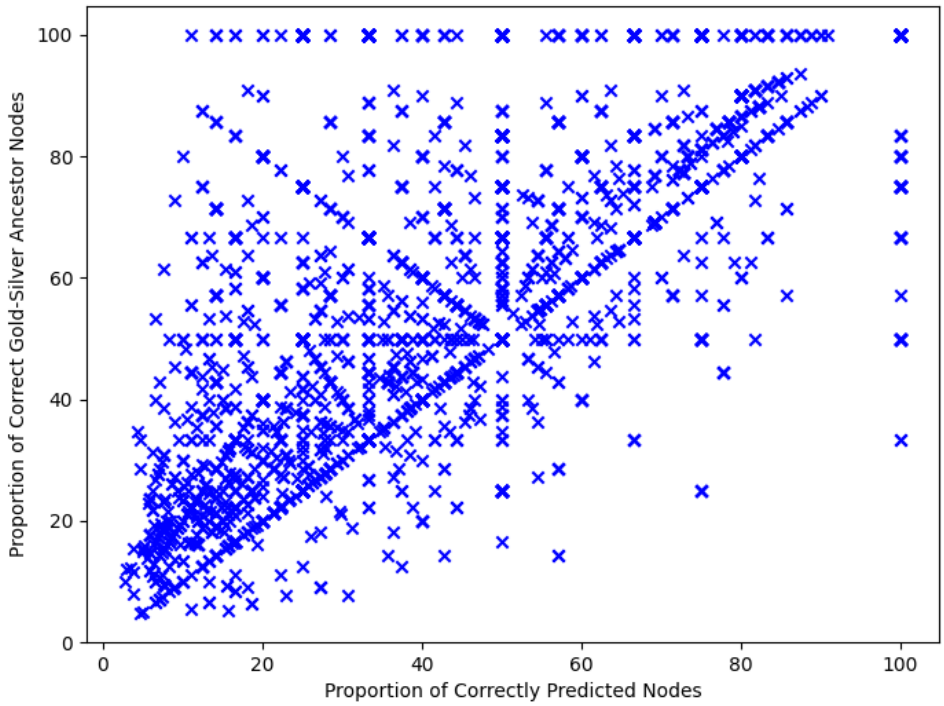


Figure 5.13: Plot of Proportion of Correct Gold-Silver Ancestor Nodes and Proportion of Correctly Predicted Nodes.

Gold-Silver Analysis: For this analysis, we have only considered those datasets which have either a gold node or a silver node. When considering datasets containing either gold or silver nodes, the correlation coefficient increases slightly

to 0.717 which is still slightly lower than that of gold node. The average proportion of correctly predicted nodes remains consistent with that of gold nodes at 62.57%.

5.6 Analysis of Varying Threshold Values for Silver Nodes Extraction

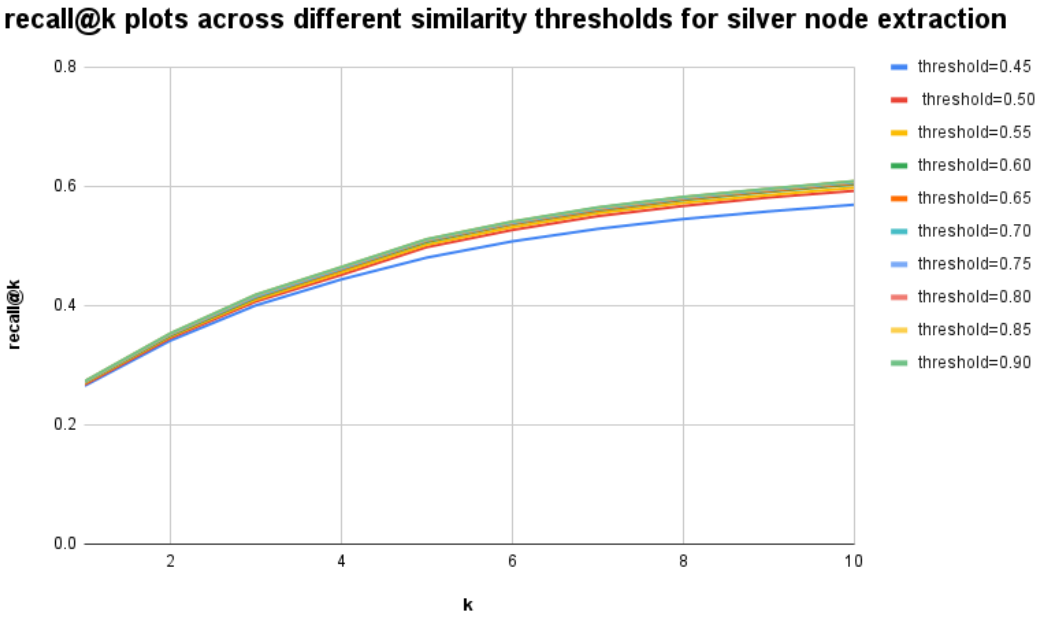


Figure 5.14: Comparison of recommended model's performance with various similarity threshold for silver nodes extraction.

As the similarity threshold is increased, there is a noticeable rise in its recall@10 score. This trend is apparent when comparing the recall@10 score for a threshold of 0.45, which stands at approximately 0.569, to that of a threshold of 0.9, which reaches its peak at 0.687. The threshold utilized in our research aligns closely with this observation, as we also employed a threshold comparable to 0.9.

5.7 Analysis of Varying m Values for m Paths Extraction

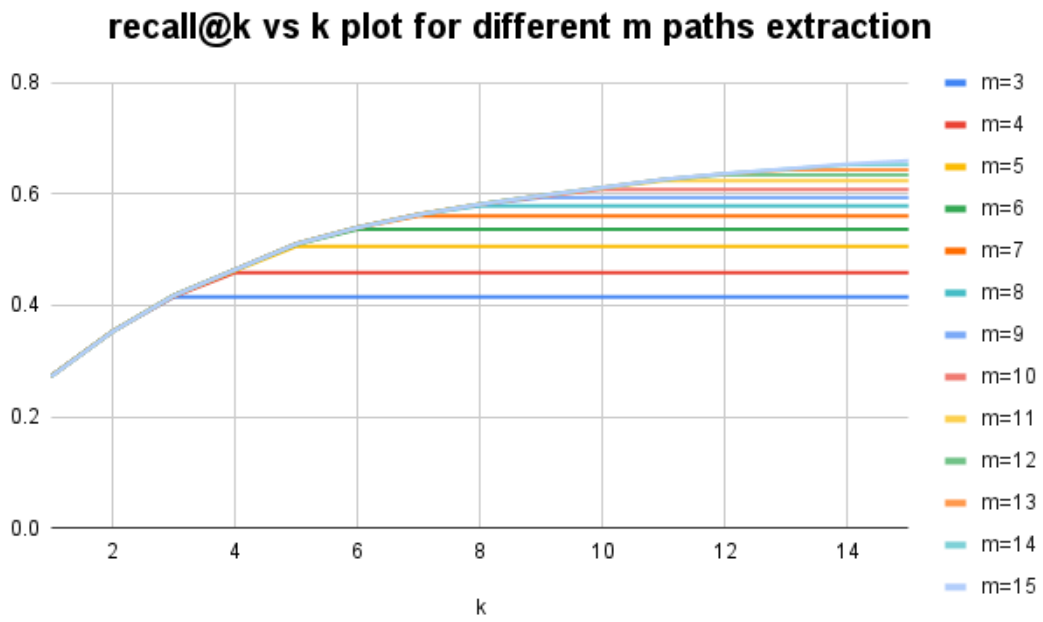


Figure 5.15: Comparison of recommended model's performance with m values for top m paths extraction.

Figure 5.15 shows that as the value of m which indicates the number of paths to extract keeps on increasing, the value of recall@k keeps on increasing. Since our data contains an average of 5 labels, we have set m to 15 in this analysis. Another thing worth noting is that the value of recall@k is slightly better when we extract k+1 paths than the value of recall@k that we obtained by extracting k paths. But this difference is not visible in the figure.

5.8 Analysis of Correct Prediction of Highly Imbalance Dataset

In this analysis, we have considered the top 10 paths from the reranked model and all the possible predictions from NeuralNLP-NeuralClassifier. The objective is to assess the efficacy of our proposed model in effectively managing highly imbalanced labels when they represent true labels. Observing Figures 5.16 to 5.20 reveals that the baseline zero-shot model, when integrated with gold and silver nodes, adeptly recommends highly imbalanced labels, a task that proves challenging for the NeuralNLP-NeuralClassifier. Upon thorough analysis, it is evident that the high frequency values of certain labels in the NeuralNLP-NeuralClassifier result from the complexity of the respective node, *i.e.*, a node can belong to multiple paths. Despite the label being highly imbalanced in its hierarchical structure, the presence of multiple paths contributes to its frequency. Similarly, when evaluating paths that have occurred at least 250 or 400 times in the entire dataset, NeuralNLP-NeuralClassifier demonstrated confident predictions compared to its performance on imbalanced labels. It is evident in Figures 5.21 and 5.22. Notably, the performance of the baseline zero-shot reranked model remains comparable.

Figure 5.16: Frequency Plot of models with correctly classifying the lowest level of true path when its highly imbalance, count=1.

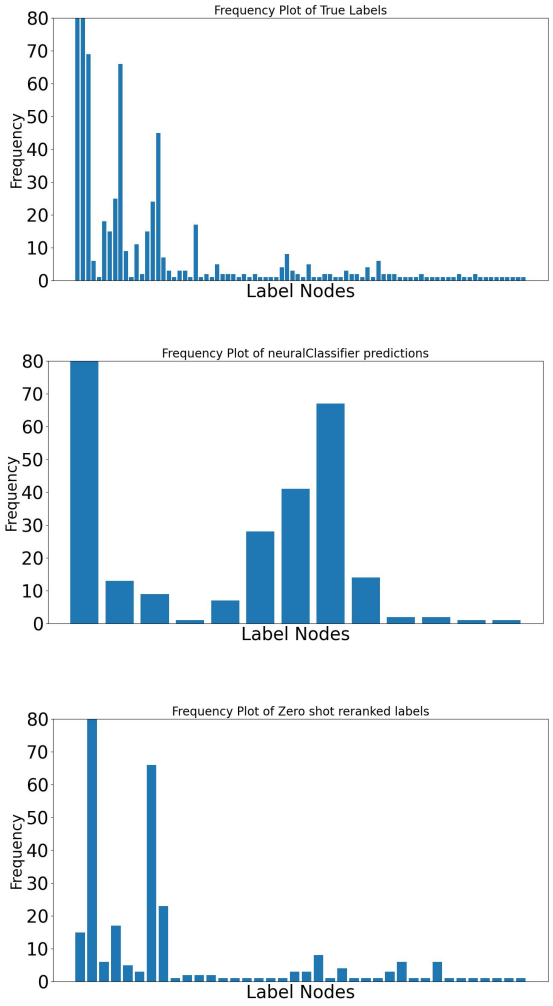


Figure 5.17: Frequency Plot of models with correctly classifying the lowest level of true path when its highly imbalance, count <5.

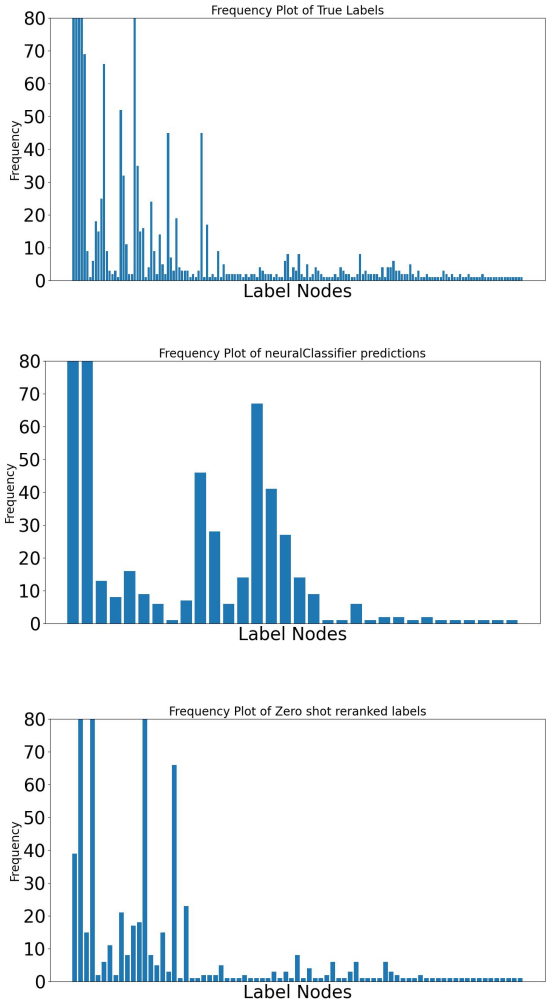


Figure 5.18: Frequency Plot of models with correctly classifying the lowest level of true path when its highly imbalance, count<10.

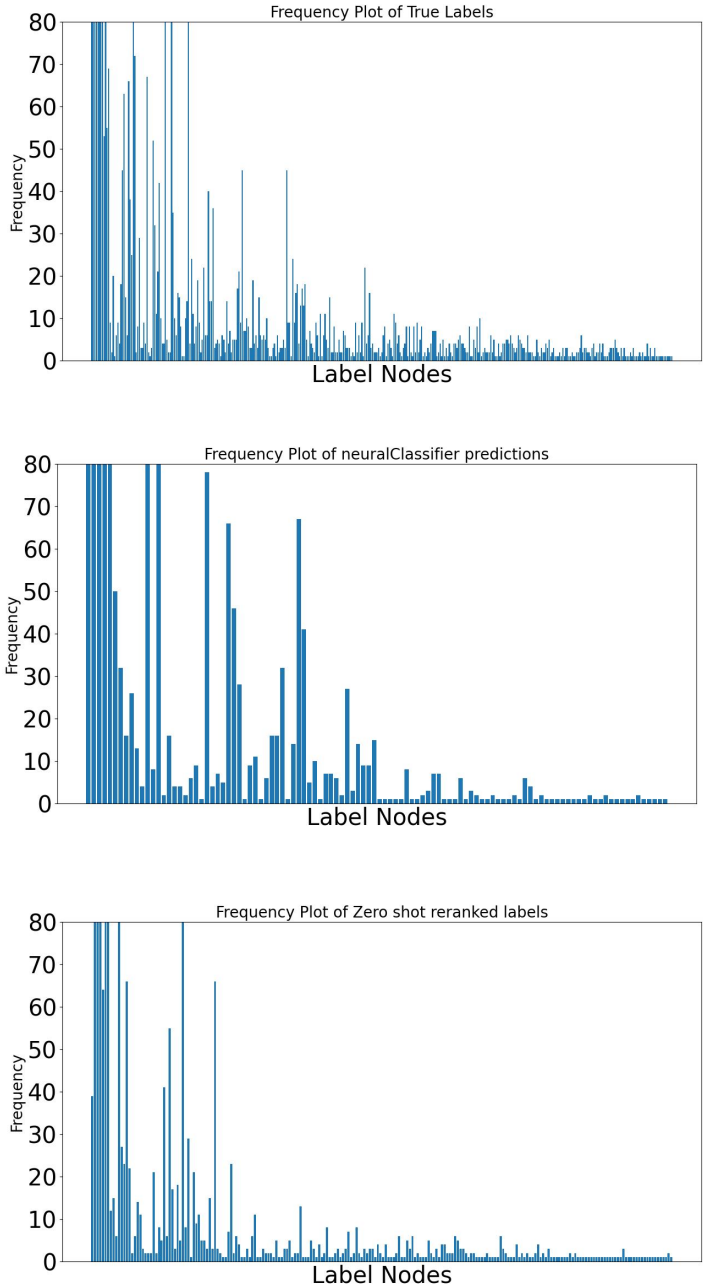


Figure 5.19: Frequency Plot of models with correctly classifying the lowest level of true path when its highly imbalance, count<30.

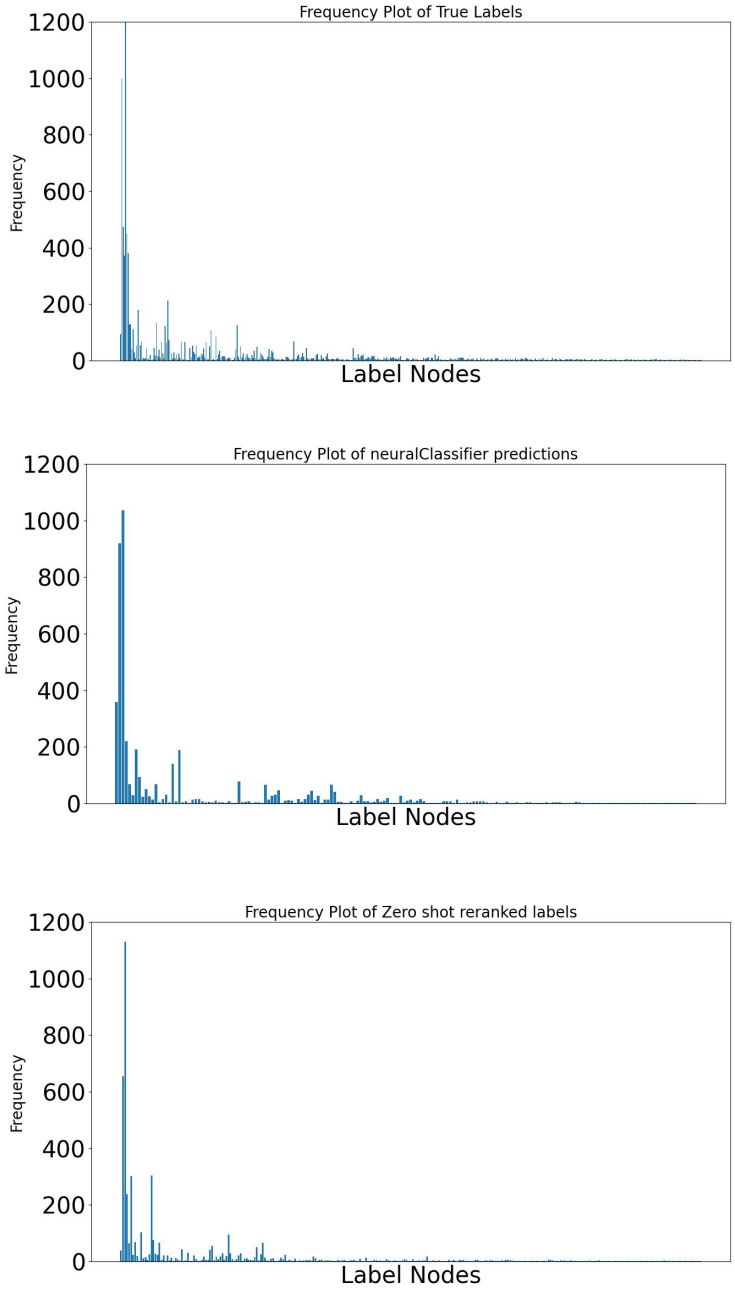


Figure 5.20: Frequency Plot of models correctly classifying the lowest level of true path in a highly imbalanced dataset with counts < 50.

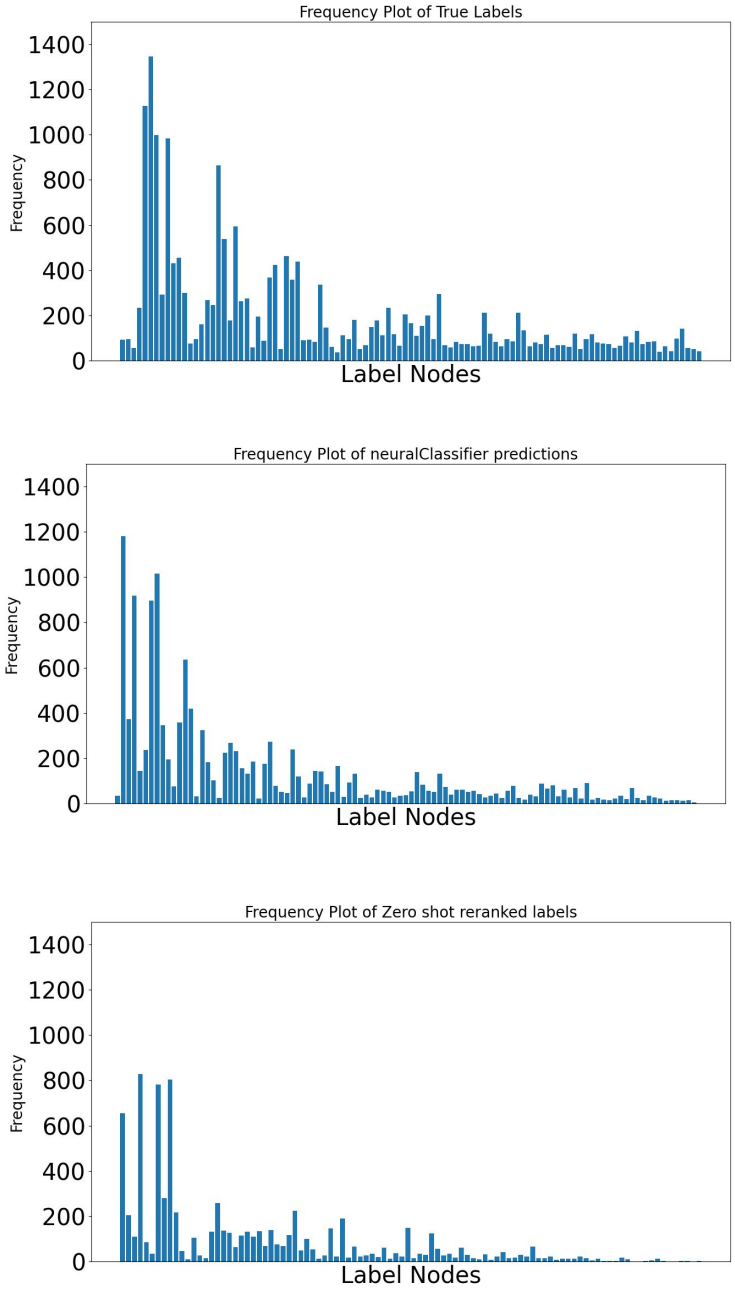


Figure 5.21: Frequency Plot of models correctly classifying the lowest level of true path in a highly imbalanced dataset with counts > 250.

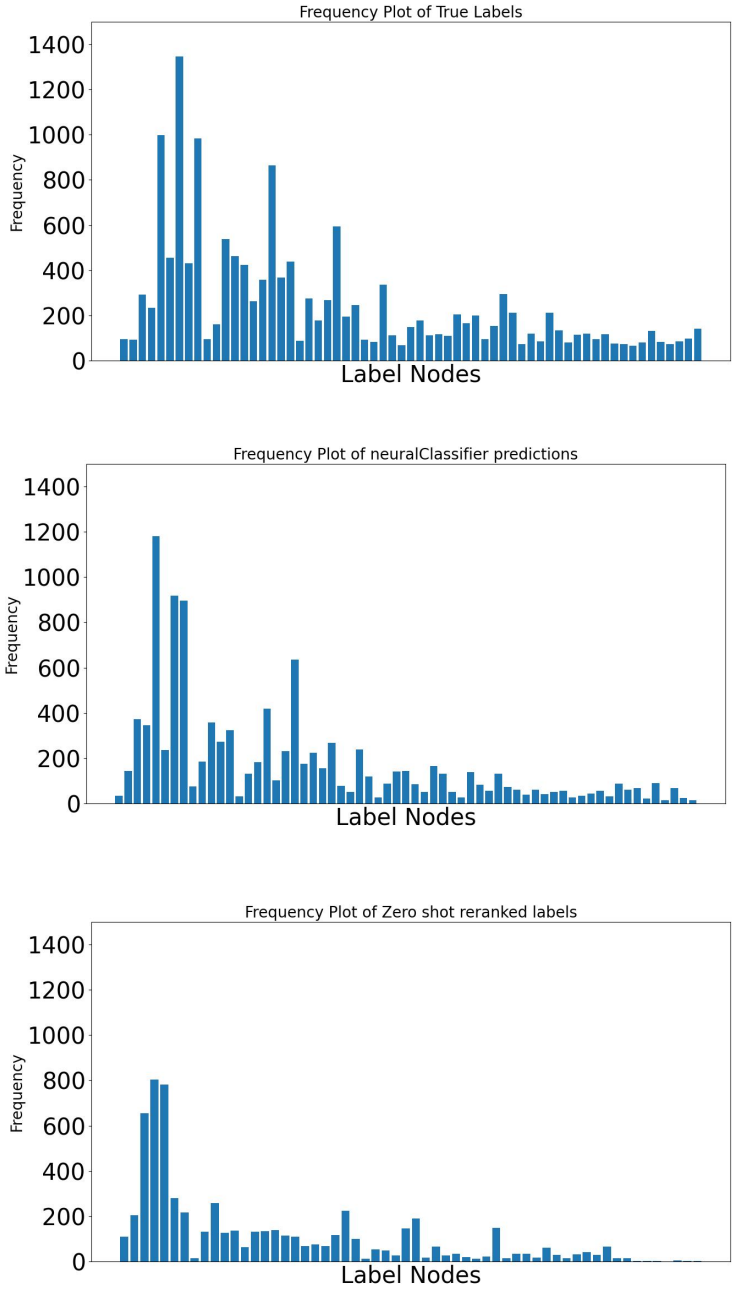


Figure 5.22: Frequency Plot of models correctly classifying the lowest level of true path in a highly imbalanced dataset with counts > 400.

Chapter 6. Conclusion and Future Work

This thesis proposes a novel framework for hierarchical multi-label text classification which is implemented in Earth science datasets having a fixed taxonomy. We can summarize following points from this research:

- The result shows that the baseline zero-shot model integrated with gold and silver nodes outperforms the NeuralNLP-NeuralClassifier and other variations of baseline zero-shot models.

- The result from the analysis shows that the text-embedding-ada-002 model surpassed other proposed embedding models in extracting top k similar labels. This success was particulary observed when leaf nodes were encoded with its ancestral information. It shows that the text-embedding-ada-002 model is suitable for reranking the obtained result from the proposed model.

- The analysis of the depth of labels across NeuralNLPNeuralClassifier and baseline line zero-shot model integrated with gold and silver nodes shows that the proposed model is indeed predicting labels up to the depth it is confident in, which fulfills one of the objectives.

- The analysis of correct prediction of NeuralNLP-NeuralClassifier and baseline zero-shot model integrated with gold and silver nodes on highly imbal-

anced labels shows that the proposed model was able to predict imbalanced labels more correctly than NeuralNLP-NeuralClassifier.

The proposed framework works best for the fixed set of taxonomy. The ever-evolving nature of Earth Science research may introduce new terminology, concepts, and relationships. Despite the changed taxonomy, the zero-shot model still works with the addition of new terminologies and relationships. However, NeuralNLPNeuralClassifier still needs data belonging to these labels and retraining from scratch. As this domain continually evolves, the significance of developing models capable of adapting to new terminology, concepts, and relationships becomes increasingly crucial. Overcoming these challenges will not only enhance information retrieval and knowledge discovery but also foster collaboration across diverse Earth science disciplines.

Since this experiment was carried out on a subset of a fixed set of GCMD hierarchy with a single root, future works could include assessing the model's performance on larger and more diverse datasets to validate its scalability and generalization capabilities. Likewise, experimenting this framework on datasets belonging to other domains such as biology, physics, medicine, etc. can be a next step to better understand the rigidity of the model. One thing worth noting is that the labels should be word representative to that particular domain and should be presented in hierarchical order. Not every domain has the dataset in the format we have in our data. So, a little bit of work needs to be done in this domain. Likewise, the recommended model in our research might not perform equally well in other domains. Therefore, we suggest experimenting on other potential

models introduced in this research and doing a bit of experimentation on different threshold values for extracting silver nodes to study performance of the model in that domain. This includes experimenting on different depth values, similarity threshold values, and embedding models to get best result before jumping straight into implementation.

References

[1] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12506–12515, 2020.

[2] Rohan Bhambhoria, Lei Chen, and Xiaodan Zhu. A simple and effective framework for strict zero-shot hierarchical classification. *arXiv preprint arXiv:2305.15282*, 2023.

[3] Lorenzo Bongiovanni, Luca Bruno, Fabrizio Dominici, and Giuseppe Rizzo. Zero-shot taxonomy mapping for document classification. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 911–918, 2023.

[4] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. An empirical study on large-scale multi-label text classification including few and zero-shot labels. *arXiv preprint arXiv:2010.01653*, 2020.

[5] Yuxiao Dong, Hao Ma, Zhihong Shen, and Kuansan Wang. A century of science: Globalization of scientific collaborations, citations, and innovations. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1437–1446, 2017.

[6] Dehong Gao, Wenjing Yang, Huiling Zhou, Yi Wei, Yi Hu, and Hao Wang. Deep hierarchical classification for category prediction in e-commerce system. *arXiv preprint arXiv:2005.06692*, 2020.

[7] Ryan Greene, Ted Sanders, Lilian Weng, and Arvind Neelakantan. New and improved embedding model, December 15 2022.

[8] Zied Haj-Yahia, Adrien Sieg, and Léa A Deleris. Towards unsupervised text classification leveraging experts and word embeddings. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pages 371–379, 2019.

[9] Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. Task-aware representation of sentences for generic text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3202–3213, 2020.

[10] Svetlana Kiritchenko, Stan Matwin, A Fazel Famili, et al. Functional annotation of genes using hierarchical text categorization. In *Proc. of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, 2005.

[11] Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. *arXiv preprint arXiv:2104.01666*, 2021.

[12] Liqun Liu, Funan Mu, Pengyu Li, Xin Mu, Jing Tang, Xingsheng Ai, Ran Fu, Lifeng Wang, and Xing Zhou. Neuralclassifier: an open-source neural hierarchical multi-label text classification toolkit. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–92, 2019.

[13] Rundong Liu, Wenhan Liang, Weijun Luo, Yuxiang Song, He Zhang, Ruohua Xu, Yunfeng Li, and Ming Liu. Recent advances in hierarchical multi-label text classification: A survey. *arXiv preprint arXiv:2307.16265*, 2023.

[14] Hamza Haruna Mohammed, Erdogan Dogdu, Abdül Kadir Görür, and Roya Choupani. Multi-label classification of text documents using deep learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4681–4689. IEEE, 2020.

[15] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072, 2018.

[16] Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. Hft-cnn: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 811–816, 2018.

[17] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020.

[18] Dominik Stammbach and Elliott Ash. Docscan: Unsupervised text classification via learning from neighbors. *arXiv preprint arXiv:2105.04024*, 2021.

[19] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

[20] Salma Taoufiq, Balázs Nagy, and Csaba Benedek. Hierarchynet: Hierarchical cnn-based urban building classification. *Remote Sensing*, 12(22):3794, 2020.

[21] Sappadla Prateek Veeranna, Jinseok Nam, Eneldo Loza Mencıa, and Johannes Fürnkranz. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proceeding of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: Elsevier*, pages 423–428, 2016.

[22] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In *International conference on machine learning*, pages 5075–5084. PMLR, 2018.

[23] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*, 2019.