DEPARTMENT OF ARTIFICIAL INTELLIGENCE
IIT KHARAGPUR

Hierarchical Label Embedding for Xtreme Multi-Label Classification (XMC)

Gourav Pathak
Under the supervision of Dr. Mahesh Mohan MR

Date: 29/11/2024

CONTENTS

Intro to Xtreme Multi-Label Classification (XMC)

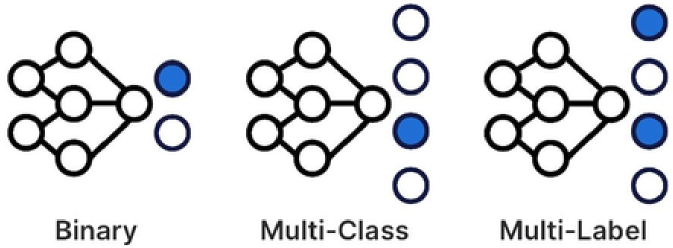
The objective in extreme multi-label learning is to train a classifier that can automatically tag a novel data point with the most relevant subset of labels from an extremely large label set.

There are several real world application to this task and for major corporate business such as:
Amazon: Automatically tags a new product like "Wireless Earbuds" with labels such as "Electronics > Audio > Headphones > Bluetooth Earphones."
Flipkart: Classifies a customer review like "Great camera but poor battery life" into sentiment labels such as "Positive (Camera)" and "Negative (Battery)."
Google: Predicts YouTube video labels like "Tech Reviews," "Smartphones," and "Unboxing" for a video about the latest iPhone.

Multi-Class vs Multi-Label Classification

Multi-label classification extends traditional classification tasks by allowing each data point to belong to multiple categories simultaneously. For instance, in a movie recommendation system, a single movie could be tagged with labels like comedy, drama, and romance.

This is different from multi-class classification, where each data point is associated with exactly one category from a set of mutually exclusive classes (e.g., categorizing animals as dog, cat, or bird).



Unlike multi-class classification, where predictions are independent, multi-label classification requires models to consider the relationships among labels to improve accuracy and relevance.

Transformer as a seq-to-seq model for XMC

T5 treats all tasks as text-to-text problems, converting inputs (like product descriptions) into outputs (like hierarchical labels). This makes it flexible for tasks like XMC, where both input (product details) and output (labels) are sequences.

The encoder processes the input sequence (e.g., a product description) to extract meaningful representations.
The decoder generates the output sequence (e.g., hierarchical labels), one token at a time, capturing label dependencies naturally.

Fine-Tuning for XMC:
T5 is pre-trained on massive datasets and fine-tuned for XMC by training it to predict structured outputs, like "details_Brand: XYZ L0_category: Electronics ..." given product details.

Implementing T5

- Introduction to the Amazon dataset that we are working with
- The Input Features are shown in Table 6.1 :

S. No.	File_name	Product_ID	Title	Price	Store	Manufacturer
1	Automotive	B000C9FJ1Y	GM 15-8535 Heating and Air Conditioning Blower...	NaN	GM	ACDelco
2	Electronics	B075HRNB8K	Polaroid PIF-300 Instant Film - Twin Pack	NaN	Polaroid	Polaroid
3	Automotive	B07J9ZFLT8	CoolingCare Radiator for 1992-2004 Chevy GMC C...	NaN	Cooling Care	Cooling Care
4	Electronics	B086384SF5	RM-GD014 Remote Control Replacement for Sony ...	7.95	Elekpia	Elekpia Factory
...	...	...	...	...	...	...
1680013	Home_and_Kitchen	B004L8V95C	Urnex Cafiza Espresso Machine Cleaning Tablets...	7.34	Urnex	Urnex

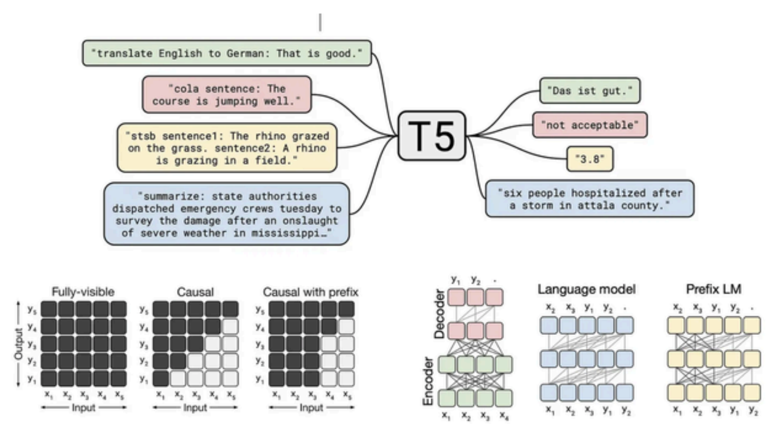
- Output Required:

1	GM (1934)	Automotive (484633)	Replacement Parts (253381)	Engine Cooling & Climate Control (18199)	Heating (2081)	Blower Motors (1718)

The fine-tuning task was framed as a sequence-to-sequence problem using a structured prompt. The model was provided with product-related information, such as the file name, title, store, and manufacturer details, as input.

Prompt:

Your task is to extract specific product details from the given information.
You need to identify and extract the following categories from the text:
- Brand of the product
- Primary category (L0_category)
- Secondary category (L1_category)
- Tertiary category (L2_category)
- Quaternary category (L3_category)
- Quinary category (L4_category)

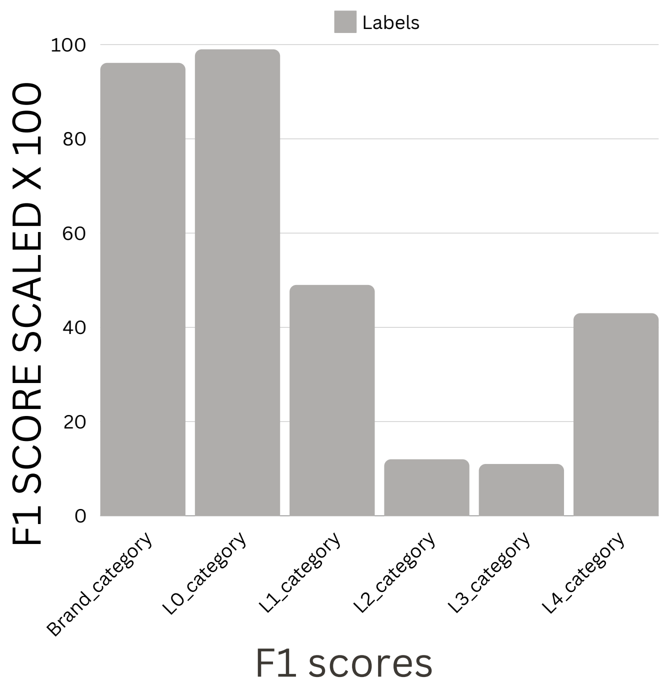


Input:
<product details here>

Output:
details_Brand: <brand> L0_category: <category_0> L1_category: <category_1> L2_category: <category_2> L3_category: <category_3> L4_category: <category_4>

Results

- **Tail Labels** are labels which have less than or equal to **25** training samples in the dataset.
- The high F1 score for Brand_category is attributed to its **strong correlation** with the Manufacturer attribute in the input data.
- The elevated F1 score observed in the L4 category suggests that the model has **predominantly** learned to output "NA" for this label, rather than demonstrating genuine classification accuracy.



Tree and Embeddings based approach

- **FastXML** is a **tree-based** algorithm designed for extreme multi-label classification.
- It achieves high accuracy for tasks requiring fast predictions, such as web searches or recommendation systems.
- But tends to lose fine-grained label relationships due to its reliance on hard partitioning and fixed tree structures.

- **SLEEC (Sparse Local Embeddings for Extreme Classification)** is an embedding-based approach that learns low-dimensional label embeddings.
- SLEEC is particularly accurate for datasets with intricate label dependencies.
- It struggles with scalability and computational efficiency as embedding learning can be resource-intensive. Additionally, it lacks the interpretability provided by tree-based classifiers like FastXML.

Hierarchical Embedding and Cluster Search for XMC
(proposed)

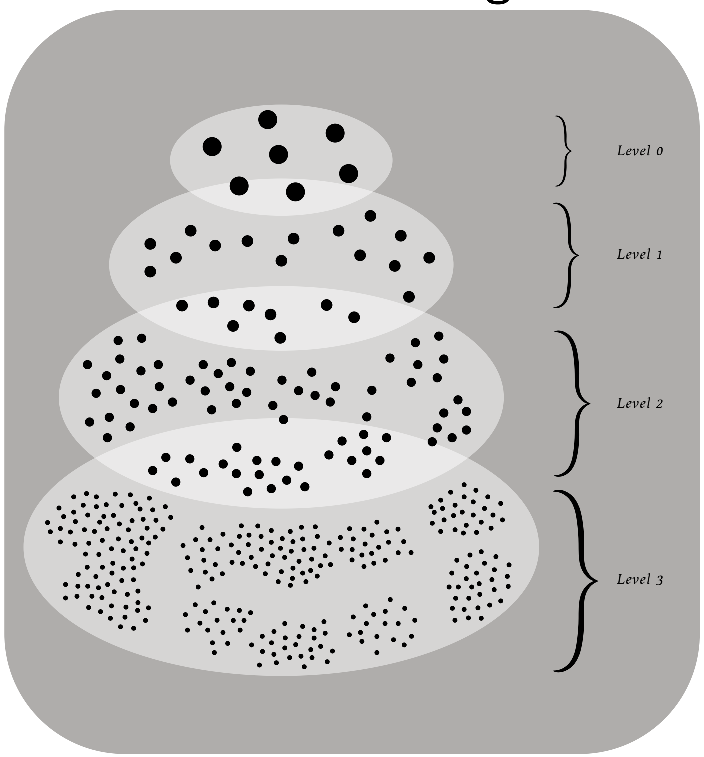
To combine the strengths of FastXML and SLEEC, we propose a Tree-Embedded Label Space, where labels are organized hierarchically, and embeddings are learned for each level of the tree. The parameters and approach for this setup are:

1. Parameters for Tree Embedding:

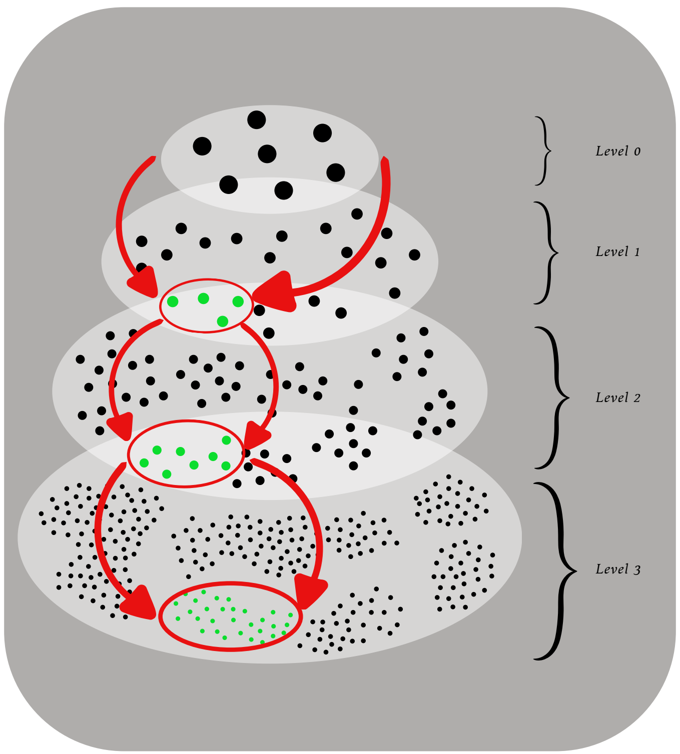
- Embedding Dimensions: Root-level embeddings (L0) have the smallest dimension, representing general labels (e.g., "movies"). As we move deeper into the tree (L1, L2, ...), embedding dimensions increase, allowing more specific labels (e.g., "horror") to be represented with higher granularity.
- Projection Matrices: Each parent node has a matrix that projects its embeddings into the space of its child nodes.
- Tree Structure: A predefined or learned tree hierarchy determines label relationships. This can be data-driven, reflecting semantic or categorical relationships.

2. Loss Function:

- Prediction Loss: For each data point, predictions are made at every level of the tree. The loss is computed as the sum of classification losses (e.g., cross-entropy) across all levels, weighted by the importance of each level.
- Consistency Loss: Child embeddings are learned as projections of parent embeddings. A consistency loss ensures that embeddings at different levels maintain hierarchical coherence.
- Regularization Loss: To prevent overfitting, regularization terms constrain embedding dimensionality and projection matrices.



Schematic of Multi-Level Target Labels representation



Schematic of Cluster Search on different dataset levels

- In summary, the **Tree-Embedded Label Space** bridges the gap between **FastXML** and **SLEEC** by introducing hierarchical interpretability and embedding flexibility. (can also be embedded in hyperbolic space)

- This hybrid approach **could** outperform both in tasks requiring high precision, scalability, and the ability to capture complex label relationships.

- This approach is supposed to bring the best of the above mentioned two methods and perform better where they individually lagged.

- Although this method is currently in the ideation phase, we would continue doing more work on this and come up with a working model with promising results and accuracies on several benchmarks.

- We have been chosen for the **poster presentation** for our work in this field at the **IndoML Graduate Forum 2024** happpeening on 22nd December 2024.

Link to poster:

https://www.canva.com/design/DAGSTgh_5Qw/hxCgKoI4FXWAdlt6O_-ONQ/edit?ui=eyJEIjp7IlAiOnsiQiI6ZmFsc2V9fX0

BIBLIOGRAPHY

- Prabhu, Y., & Varma, M. (2014, August). Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 263-272).

- Bhatia, Kush, et al. "Sparse local embeddings for extreme multi-label classification." Advances in neural information processing systems 28 (2015).

- Kharbanda, S., Gupta, D., Schultheis, E., Banerjee, A., Hsieh, C. J., & Babbar, R. (2024). Learning label-label correlations in Extreme Multi-label Classification via Label Features. arXiv preprint arXiv:2405.04545.

- Nickel, Maximillian, and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations." Advances in neural information processing systems 30 (2017).

THANK YOU