GSoC 2025: UCSC OSPO

Medicinal Language Embeddings

Proposal: AI-Powered Graph Network & LLM for Enhanced Embeddings, Dimensionality Reduction & Clustering

- **Project**: **Medicinal Language Embeddings**
- **Organization**: UCSC OSPO
- **Mentors**: Oskar Elek, Kiran Deol
- **Applicant**: Ayush Shaurya Jha
- **Project Length**: 350 Hours

PERSONAL INFORMATION

CONTACT INFORMATION

- **Name**: Ayush Shaurya Jha
- **Primary Email**: shauryasphinx@gmail.com
- **Secondary Email**: ayush.2022ug3013@iiitranchi.ac.in
- **Github Profile**: https://github.com/jhaayush2004

- **LinkedIn**: Click to visit
- **My Resume**: Click to visit
- **Location**: Patna, Bihar, India
- **Timezone**: IST (UTC +5:30)

Relevant Experiences

I have extensive experience in integrating AI and harnessing the prowess of LLMs to increase accuracy of research works and daily tasks in biological and healthcare domain which has given me significant **domain knowledge of bioinformatics and how medicinal chemistry works**. And this makes me suitable for this project as I believe that along with technical knowledge, one also needs domain knowledge, which I have gained from numerous projects in the field of AI in healthcare. And below are the testaments to my technical and domain knowledge of bioinformatics and medicinal chemistry.

Personal project, experience and research works which gave me significant exposure to **domain knowledge of Bioinformatic and medicinal chemistry field as well as are testaments to my technical knowledge** of AI, LLMs, Deep Learning, Machine Learning, and Optimizations:

- **NeuroVisionAI:TumorMapperNet** is the project having two parts. The first part deals with classifying and segmenting brain tumor out of MRI scans of brain with the help of self-designed deep learning architecture. The second part classifies the type of tumor and suggest which molecules and compounds would be most beneficial for the treatment and thus uses the molecular embedding in finding out the medicines by embedding similarity calculation.

• **Generative AI intern (LLM Engineer intern)**, Tapfinity Technologies Pvt. Ltd.

- It's an AI healthcare startup dealing in **bioinformatics** as well as personalized AI health assistance and working here gave me extensive experience of medicinal and other **biological embeddings generation, visualization,** and **storage**.
- I also worked on a **Graph RAG** which consisted of scraped data of around 1200+ diseases and has around 9+ millions entity nodes and 14+ million entity relationships and deployed it on a neo4j instance, this solution is used to perform diagnosis of users.
- I designed a personalized **AI health coach** which involved exploring and analyzing and molding a whole bunch of data with about 1M+ instances.
- I worked on designing **LLM powered RAG** pipeline which involved embedding generations, its storage into database and then its retrieval based on similarity check.

- **Applied AI and Research intern, ISRO**(Indian Space Research Organization, NARL)
  - I worked on designing a robust and well optimized deep learning based model from scratch which is getting trained on Martian ionoshpheric data composed of frequency, time-delay and altitude numeric values.
  - The pipeline I designed till now is capable of precisely recognizing and segmenting out ionospheric echo (time-delay and frequency) out of numeric data and get the real-time plot with an accuracy of about 98.2% as of now, and continuous optimizations for better convergence and training are ongoing.
  - This automation has reduced the analysis time of the Martian ionosphere by about 35%. Apart from this, I'm also finetuning a LLM on the Martian data available till now along with doing changes in its architecture and quantization to make it memory efficient keeping its precision maintained.
  - I also implemented reranking and filtration techniques which reduced LLM hallucinations by about 40%.
- **AI ML Lead** at House of Geeks, IIIT Ranchi
  - I am appointed as Lead of AI ML club at House of Geeks which is technical society of my college where I work as project maintainer and guide passionate developers how to strive in the domain of AI and Machine Learning, guiding them in their project and Open-Source contributions.
  - Integrated the dynamic LLM and RAG powered bot into the official website of our college, fostering student collaboration and providing valuable resources for academic, technology and extracurricular activities.

My AI Research Publication:

- Deepfake Detection with Multi-Fusion and Residual Learning in Deep CNNs

  The paper, with me as the first author, has been accepted at the 5th International CVR conference 2025, NIT Goa (25-26 April,2025). It presents a novel Multi-Fusion Residual Deep CNN (MFuRe-DCNN) approach that enhances deepfake detection by optimizing feature retention efficiency and generalization.

STUDENT AFFILIATION

- **UNIVERSITY**: Indian Institute of Information Technology Ranchi
- **DEGREE**: Bachelor of Technology (B.Tech.)
- **MAJOR**: Computer Science and Engineering (with specialization in Artificial Intelligence and Data Science)
- **EXPECTED GRADUATION**: 2026

BRIEF BIO

I am Ayush Shaurya Jha, a third-year undergraduate student at the Indian Institute of Information Technology Ranchi (India) pursuing Bachelor of Technology in Computer Science and Engineering (with specialization in Artificial Intelligence and Data Science), with comprehensive experience in AI, Large Language Models (LLMs), Machine Learning, Deep Learning, Optimization, and NLP through various projects, research works, and internships and my experiences mentioned above are testaments to my words. My interests lie in exploring and implementing innovative AI-driven techniques to analyze and interpret complex data, particularly in bioinformatics and computational biology, where machine learning and AI can unveil hidden essence and complex patterns and enhance scientific discovery. I am highly fascinated by Mediglot's use of PolyPhy, a natural network-inspired approach which was originally designed for reconstruction of cosmic web, now being applied to medicinal embeddings and giving unexpectedly good results.

My skill set includes AI, Machine Learning, Large Language Models (LLMs), Deep Learning, NLP, Graph-Based Learning, Optimization, Python, C, C++, and JavaScript.

Due to my involvement in open-source projects, research works and personal projects, I have developed a deep understanding of need for well-documented, maintainable, and efficient code. The **Mediglot's Medicinal Language Embedding Project,** which focusses on refining medicinal embeddings and enhancing visualization and clustering using LLM and AI-driven techniques, aligns strongly with my expertise, interest and passion. I am very excited about leveraging the LLMs, advanced deep learning graph approaches, and novel similarity metrics to enhance clustering and representation of medicinal data. I look forward to contributing to this project by integrating cutting-edge AI methodologies to make Mediglot a more robust and comprehensive tool for biomedical research and visualization.

PROJECT DESCRIPTION

ABSTRACT

Mediglot is a web-based tool which has been designed for the better visualization of medicinal embeddings in a 3D space. However, its current approach depends fully on salt compositions for embedding generation, which limits the analysis and the interpretability of relationships between medicines. This project aims to advance Mediglot's embedding and clustering techniques by integrating salt composition along with molecular structure, Biomedical based LLM generated textual descriptions (like target symptoms, uses and so on), Pharmacological descriptions (like action mechanism, target molecule, effects and so on) and using unsupervised Graph based Neural Network (GCNN) technique to extract more refined embeddings followed by use of advanced novel similarity metrics. By using PolyPhy's network-inspired methods and incorporating LLM models, I have planned to enhance Mediglot's clustering, improve medicinal representations, and come up with a more meaningful exploration of medicinal embeddings.

BACKGROUND

It's important to understand and explore relevant relationships between medicines for drug discovery, similarity measurement, and personalized medicine. In the current times, Mediglot provides visualization of high-dimensional medicinal embeddings derived from only salt compositions of medicines, using UMAP and similarity analysis through the Monte-Carlo Physarum Machine (MCPM) metric. However, medicinal embedding lacks the inherent properties like molecular structure, action mechanism, target molecules, uses and so on, thus inducing inaccuracy and also making it difficult to extract much information of medicines from their respective embeddings or to calculate similarity between two medicines.

However, the existing approach has several limitations:

- **Limited Feature Representation**: All of functional, chemical and pharmacological properties of a medicine can't be captured by using salt composition alone.
- **Lack of Structural Awareness**: It may happen that medicines with same salt compositions but different action mechanism, use case and pharmacological properties may be grouped together leading to inaccuracy in the system.
- **Suboptimal Similarity Metrics**: The similarity metric currently in use, i.e, MCPM, may not always capture the insightful relationship and similarity between medicines based on their embeddings which induces the need to check it against various other novel similarity metrics.
- **No Recommendation system**: Currently, Mediglot doesn't has any recommendation Pipeline for similarity search based recommendation of medicines.

Thus, the idea proposes to enhance the robustness and comprehensiveness of medicinal embeddings by incorporating molecular structure, textual, and pharmacological descriptions (generated by biomedical-based LLMs) followed by unsupervised Graph-based Neural Network (GCNN) to get more refined and neighborhood (other medicines) aware medicinal embeddings to achieve more accurate and comprehensive similarity calculations (by evaluating various metrics using these embeddings) and clustering of medicines. Furthermore, we will explore various dimensionality reduction techniques and try to enhance visualization and interpretability of medicines. Thus, by advancing Mediglot's capabilities with AI and LLM-powered techniques and using PolyPhy's network-based approach, we would aim to develop a more comprehensive as well as insightful tool for medicinal data exploration and medicinal discovery.

PROJECT PROPOSAL

Overview

The proposal aims to advance Mediglot's medicine visualization and clustering techniques using novel Large Language Models (LLMs), advanced techniques in embeddings generation, graph-based neural networks, dimensionality reduction, and novel similarity metrics. I have divided the project into five phases to ensure structured development and traceability:

- **1. Robust Embeddings Generation –** Constructing high-quality medicine embeddings using transformer-based models.
- **2. Graph-Based Representation Learning** – Developing a similarity graph and training a Graph Convolutional Neural Network (GCNN) to refine embeddings.
- **3. Efficient Dimensionality Reduction, Clustering & Visualization**– Improving computational efficiency and interpretability by trying out and comparing various dimensionality reduction algorithms on high-dimensional medicinal embeddings to enhance visualization using 'Mayavi' and clustering.
- **4. Advanced Similarity Metrics for Retrieval –** Enhancing retrieval accuracy using novel similarity measures.
- **5. Final Optimization & Deployment** – Integrating the pipeline, optimizing performance, and documenting the project.

Technologies

The project will take into use deep learning, graph-based techniques, and statistical methods, using the following tools and technologies:

- **Programming Languages**: Python, JavaScript(vanilla)
- **Deep Learning Frameworks**: Transformers, PyTorch, TensorFlow, Keras, HuggingFace, SciPy, NumPy, pandas, Scikit-Learn
- **Chemical Informatics:** RDKit, PubChemPy
- **Graph-Based Learning**: Networkx (for graph operations, diagnostics and visualization), DGL (Deep Graph Library for experimentation and backup), Geometric (PyG) (primary)
- **Dimensionality Reduction & Clustering**: UMAP-learn, T-SNE, PCA, HDBSCAN, DBSCAN, Agglomerative Clustering, KMeans
- **Similarity Metrics & Retrieval**: FAISS, sentence-transformers, Tanimoto, Cosine, Pearson, Jaccard, Manhattan, Monte-Carlo Physarum Machine (MCPM) Metric and other similarity metrics
- **Visualization & Data Interaction**: PyPlot, Mayavi (for 3D visualization), BeautifulSoup, Selenium, Chromedriver

PLAN OF ACTION

Phase 1: Generating High-Quality Medicinal Embeddings

Objective

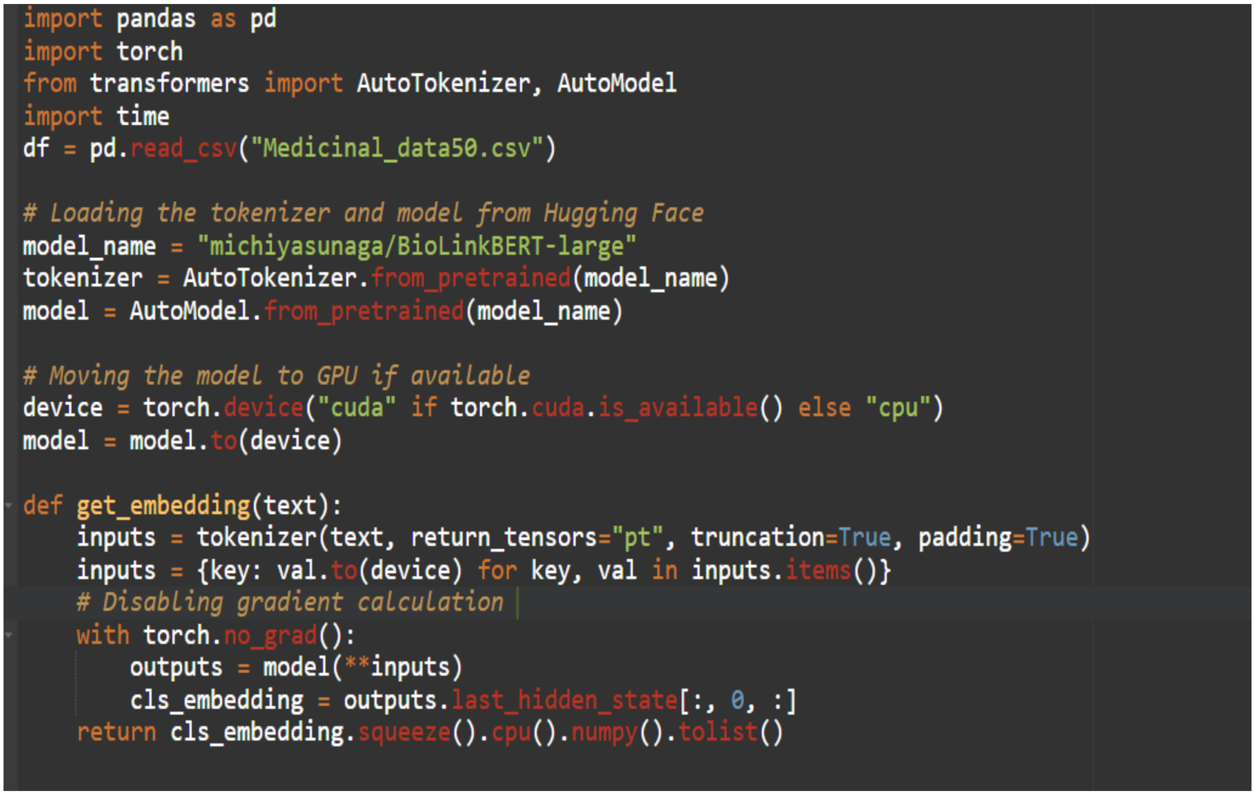
To make the representation of medicines more comprehensive, embeddings should not capture only salt compositions but should also incorporate additional relevant biomedical information like chemical structural information of medicine and its description, which may include action mechanism, chemical interaction, uses, target disease, etc.

Data Sources & Features to Include:

- **Salt Composition** – The fundamental property used in Mediglot. Plans are to extract the salt composition and vectorize it using LLMs like **biomedical-based LLMs** such as **BioLinkBERT-large** (Primary choice), and **BioBERT, SciBERT, MedBERT** for experimentation purpose and then store the embeddings **(Embedding E1).**

- **Medicine Description** – Medicinal description will include action mechanisms, interactions, and target molecules. I would extract these descriptions in textual format from sources and would transform it into embeddings using transformer models **BioLinkBERT-large** (Primary choice), and **BioBERT, SciBERT, MedBERT** for experimentation purpose depending on the performance**. (Embedding E2).**

  Sample implementation using LLM for text embedding

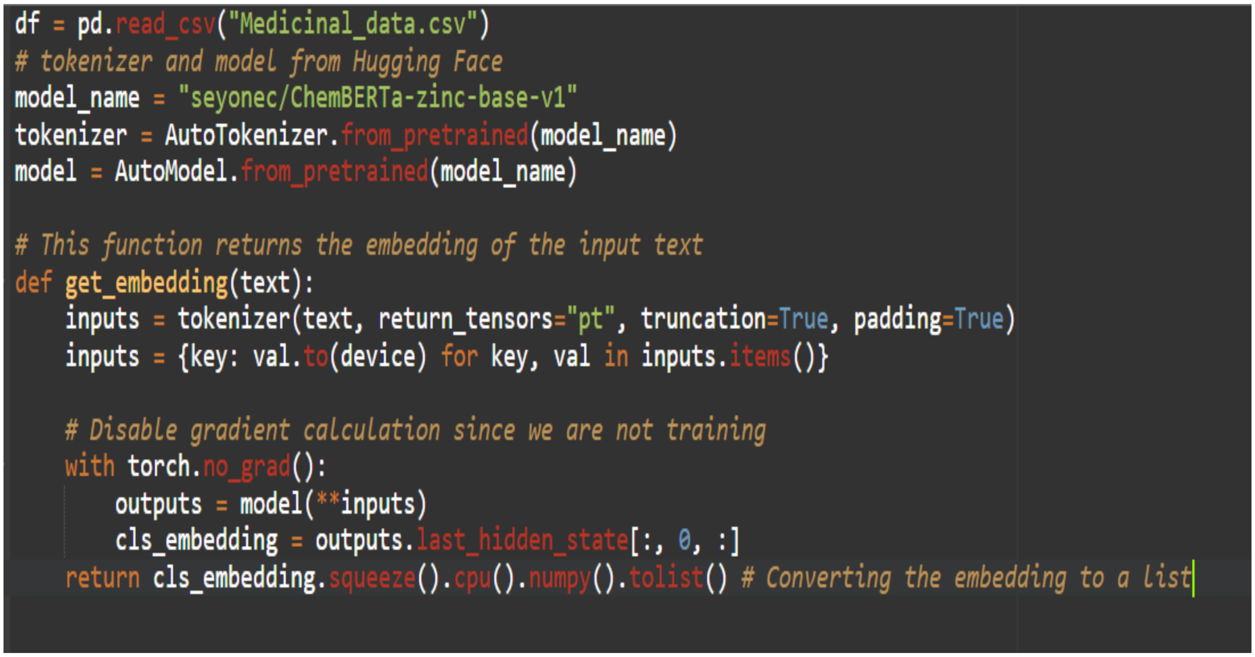


```
import pandas as pd
import torch
from transformers import AutoTokenizer, AutoModel
import time
df = pd.read_csv("Medicinal_data50.csv")

# Loading the tokenizer and model from Hugging Face
model_name = "michiyasunaga/BioLinkBERT-large"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)

# Moving the model to GPU if available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = model.to(device)

def get_embedding(text):
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    inputs = {key: val.to(device) for key, val in inputs.items()}
    # Disabling gradient calculation
    with torch.no_grad():
        outputs = model(**inputs)
        cls_embedding = outputs.last_hidden_state[:, 0, :]
    return cls_embedding.squeeze().cpu().numpy().tolist()
```

- **Pharmacological Data** – Plans are to extract pharmacological information of medicines from databases like `1mg`, `PubChem`, and `ChEMBL`, and embed this information numerically using LLMs **BioLinkBERT-large** (Primary choice), and **`BioBERT`, `SciBERT`, `MedBERT`** for experimentation purpose. I will test all possible combinations of these LLMs and compare with the current embeddings used in Mediglot. (Embedding E3)

- **Molecular Structure –** Molecular structure of medicines would be extracted using advanced biomedical based LLM ChemBERTa and embeddings would be generated using **SMILES**, which afterwards will be used. (Embedding E4)

Sample Code of Implementation



```python
df = pd.read_csv("Medicinal_data.csv")
# tokenizer and model from Hugging Face
model_name = "seyonec/ChemBERTa-zinc-base-v1"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)

# This function returns the embedding of the input text
def get_embedding(text):
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    inputs = {key: val.to(device) for key, val in inputs.items()}

    # Disable gradient calculation since we are not training
    with torch.no_grad():
        outputs = model(**inputs)
        cls_embedding = outputs.last_hidden_state[:, 0, :]
    return cls_embedding.squeeze().cpu().numpy().tolist() # Converting the embedding to a list
```

- Now, since my primary choice of LLM for embedding generation of **Salt Composition, Medicine Description and Pharmacological Data** is same (i.e., **BioLinkBERT-large),** so the shape of embeddings for these three information is (None, 1024) where as since I will use **ChemBERTa** for embedding generation of Molecular Structure, so it will have embedding shape of (None, 768) and thus, we will need to make embeddings to equal shape for efficient concatenation.

- So, I have plans to go with up-sampling the Molecular Structure Embeddings and not with down-sampling other three information embeddings because this may cause significant loss of information. Now, for oversampling I am not going for zero padding as it may add irrelevant information to final embeddings. So, I will go with

Multi Linear Perceptron (MLP) for up-sampling 768 dimensional structural embeddings to 1024-dimensional vector.

- MLP will apply non-linear transformation to increase dimension and will help us retaining high-level properties of the original 768-Dim Embeddings.
- Why MLP for up-sampling 768-Dimensional Embedding?
    - MLP will retain Global patterns and alignment across samples, so it preserves Distances between samples, Relative Cluster tendencies and general topology of the embedding space.
    - It creates a smooth and learnable transformation that easily map embeddings in way better manner than duplication or random padding by allowing continuous information flow through it is layers thus, making it more learnable.
- After implementing MLP based up-sampling, I will experiment same with Autoencoders and contrastive learning both. But MLP is much more than better for our task.
- All these stored and normalized embeddings will be normalized using `L2 normalization` and then concatenated to give the final informative and robust embeddings for each medicine.
- L2 normalization is used because it aligns naturally with cosine similarity thus, ensuring numerical stability and preserving directionality of embeddings, which is crucial for similarity-based tasks, clustering and GNNs that will be used in next phase for embedding refinement.

Combined and Normalized Embedding = Concatenation [E1, E2, E3, E4].

Novelty

- The idea of merging structural, pharmacological and compositional information of medicines will make the embeddings more comprehensive and biologically insightful.
- MLP based dimensional up-sampling has been considered in place of random padding or down-sampling which will prevent embeddings from information loss and addition of irrelevant information.
- Weighted concatenation instead of vanilla approach to maintain the actual significance of each information in the embedding instead of treating them of equal importance.

Why This Works Better?

- This ensures capture of both **molecular properties** (including salt composition and structural information) as well as **pharmacological descriptions** instead of relying only on salt composition.
- `SMILES` **+** `Transformers` provide a more comprehensive and informative representation of the drug.

Phase 2: Deep Learning Graph-Based Representation Learning

Objective

In this phase, we will ensure the refinements of the embeddings of medicines by unsupervised training of the GraphSAGE or GCNN, which will generate new robust embeddings for all medicines after going through multiple epochs or iterations. We would carry out this phase in two sub-phases. The *first sub-phase will be graph creation*, and the *second sub-phase will be extracting refined embeddings by training of the GraphSAGE(as decided by Pre-GSoC work)*.

2.1 Constructing the Medicine Similarity Graph

Objective

Our aim in this phase will be to create a network where medicines will be represented by nodes and edges will define relationships based on chemical and functional similarity.

Graph Construction

- Nodes -> Medicines.
- Edges -> Defined by similarity metrics.
- Edge Weights -> Degree of similarity.

All medicines would be placed as nodes, each having their attributes in the form of embeddings. Edges between medicines would be added based on their similar functions, chemical composition, structure, etc., and it would be executed using various similarity metrics:

Tanimoto Coefficient (Jaccard Coefficient (Jaccard for molecular fingerprints)) – can best measure structural similarity of medicinal embeddings.

Cosine Similarity – Mostly preferred for calculation of similarity in the high-dimensional embedding space.

Pearson Correlation – Can be used in understanding relationships between features.

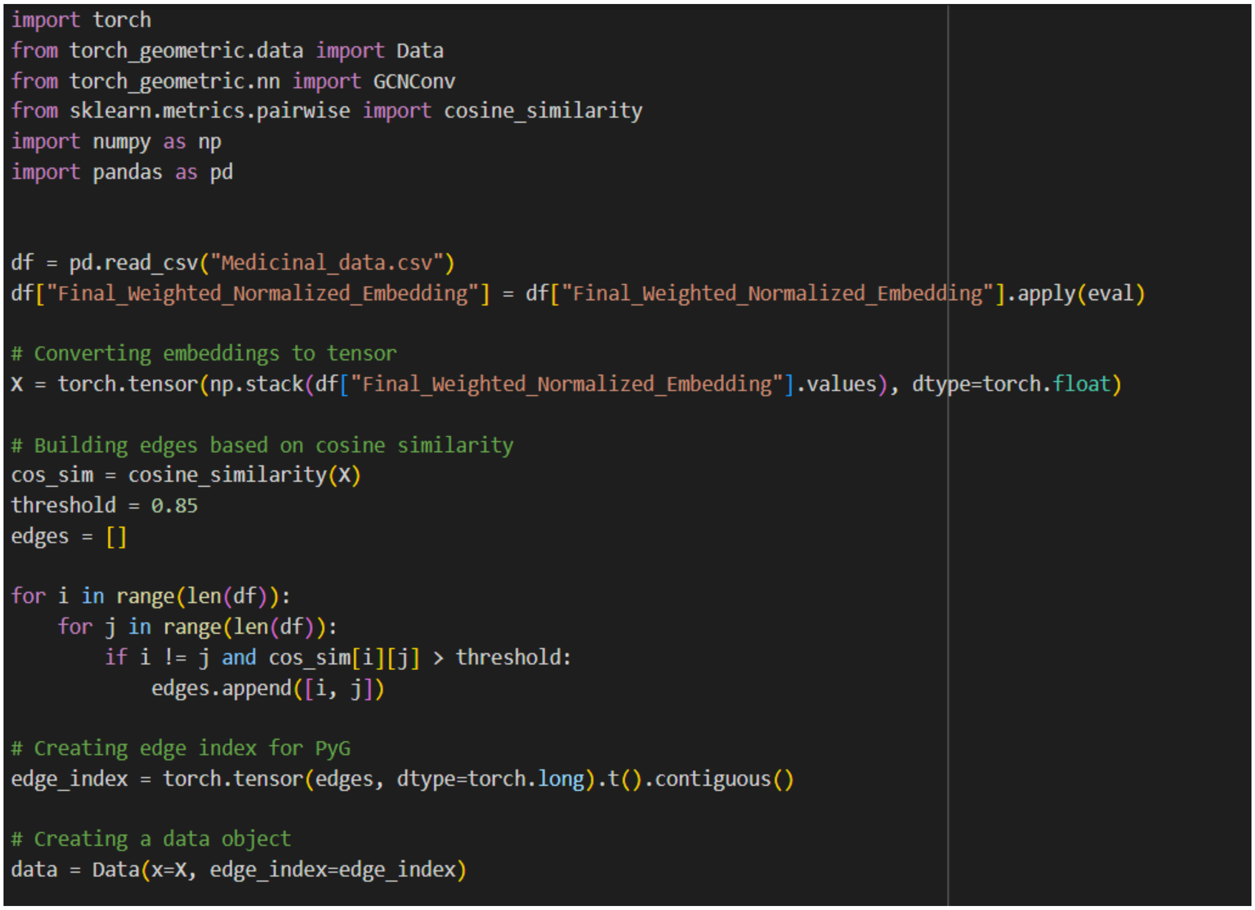
MCPM Metric – Mediglot's existing method for comparison of medicines.

All the above metrics has been experimented with in Pre-GSoC task and the best performing is Cosine Similarity, so we will go with it for graph creation which is to be feed to GraphSAGE.

Tools & Libraries:

- **PyGraph** –This will be used to create and analyze the medicine graph. We will use any one of them on the basis of better performance.
- **Scipy & Scikit-learn** – They would be used for similarity computations.

Sample Implementation Code



```
import torch
from torch_geometric.data import Data
from torch_geometric.nn import GCNConv
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np
import pandas as pd


df = pd.read_csv("Medicinal_data.csv")
df["Final_Weighted_Normalized_Embedding"] = df["Final_Weighted_Normalized_Embedding"].apply(eval)

# Converting embeddings to tensor
X = torch.tensor(np.stack(df["Final_Weighted_Normalized_Embedding"].values), dtype=torch.float)

# Building edges based on cosine similarity
cos_sim = cosine_similarity(X)
threshold = 0.85
edges = []

for i in range(len(df)):
    for j in range(len(df)):
        if i != j and cos_sim[i][j] > threshold:
            edges.append([i, j])

# Creating edge index for PyG
edge_index = torch.tensor(edges, dtype=torch.long).t().contiguous()

# Creating a data object
data = Data(x=X, edge_index=edge_index)
```

Why This Works Better?

- Captures relationships dynamically rather than relying on static similarity scores.
- Graph representation would help us in inducing neighborhood information (information about neighboring node medicines) incorporated and thus, help in better clustering. We can take an analogous case, suppose we have to find two

friends, then apart from their character similarity, we would also go for checking that how many mutual friends they have. Similarly, we get more about the similarity of two medicines when we know more about their cluster and groups. And, by this, we can aim to create comprehensive embeddings for all medicines with robust clustering.

2.2 Refining Embeddings Using Unsupervised Learning of Graph Convolutional Neural Networks (GCNNs)

In this sub-phase, I have earlier planned to train the Graph Convolutional Neural Network (GCNN) in an unsupervised manner and obtain refined embeddings for each medicine but now I will go on with using GraphSAGE as it performed better than GCNN in the test which I performed on a data sample.

Why is this step necessary?

- Vanilla embeddings do not incorporate neighborhood relationships.
- GraphSAGE always operate on graphs, where nodes (i.e., medicines) are connected by edges (which represent similarity relationships).
- The idea is to advance each node's embedding by aggregating information from its neighboring nodes.
- This aggregation ensures that the GraphSAGE learns the representations that capture the contextual relationships between medicines.
- Thus, in the similarity graph, the GraphSAGE ensures that medicines learn from their neighbors.
- Out of **GCN, GAT, GraphSAGE**, I will use **GraphSAGE** after examining our requirements and performance comparison between the three on a sample of our data.

Working of GraphSAGE

- **Message Passing:**
  - The GraphSAGE layer collects the embeddings of the neighboring nodes for each node.
  - These collected embeddings from neighboring nodes are then transformed linearly and then aggregated by averaging, summing, etc.
  - Then, finally, the node's own embedding is also included in the aggregation after getting transformed.
- **Aggregation and Update:**

  - The node's original embeddings are then combined with the aggregated embeddings.
  - This combined information is then introduced to an activation function (e.g., ReLU) to introduce non-linearity.
  - The result of these steps is the updated embedding for that node.

Unsupervised Training of GraphSAGE(GS)

- The unsupervised learning objective is to keep the essence of the original embeddings as much as possible, while still allowing the GraphSAGE to refine them based on the graph structure.
- **loss = F.mse_loss(out, data.x)** is the key, where **out** represents the GraphSAGE-generated embeddings and **data.x** are the original embeddings.
- Loss Calculation (MSE Loss): **Mean Squared Error (MSE)**
  - The MSE loss calculates the average squared difference between the GS's output embeddings (**out**) and the original node embeddings (**data.x**).
  - **F.mse_loss(out, data.x)** calculates this average squared difference.
  - The GS's gets trained to minimize this loss, i.e., it tries to make the generated embedding as close as possible to the original embeddings.
- Gradient Calculation (Backpropagation):
  - Backpropagation
    - After the calculation of the loss, the gradients of the loss with respect to the GraphSAGE's weights are calculated.
    - These gradients are easily and accurately calculated by back propagation algorithm by passing the error signal backward via the network.
  - Gradient Information:
    - The gradients give us the measure about the contribution of each weight in GraphSAGE network in the loss.
    - A positive-gradient points that increasing the weight would increase the loss too while negative gradient indicates that increasing the weight would reduce the loss and vice-versa.
  - Weight Updates:
    - The **optimizer.step()** line updates the GCNN's weights using the calculated gradients.

    - The **optimizer.zero_grad()** line does the task of clearing each gradient before each update.
    - The aim is to converge the loss function to the minimum loss point or the optimal position by adjusting the weights in the direction of low loss.

Why GraphSAGE extracted embeddings are Better:

- **Contextual Embeddings**:
    - The aggregation process in GraphSAGE refines the initial embeddings to generate new embeddings that are aware of context of all other medicines in the similarity graph.
    - This contextual information truly important for getting the essence of complex relationships between medicines.

- **Improved Clustering**:
    - GraphSAGE helps to group similar medicines more accurately by taking into account neighborhood information.
    - Medicines which are close in the graph (highly similar) will have their embeddings pulled closer together, and this will make them easier to cluster.
- **Enhanced Retrieval**: Since the embeddings are now equipped with more information about the medicines relationships, so similarity searches get improved as a result.
- **Robustness**: GraphSAGE is robust to missing data as well as if present in the graph, as they have ability to learn from the overall structure of the network.

Novelty

Applying GraphSAGE on the final concatenated embeddings which were retrieved in the last phase for making the embeddings more comprehensive and insightful to enhance quality of clustering and similarity measurement by making embeddings aware of neighborhood information and complex relationships of the embedding with other medicines (i.e., embeddings).

Phase 3: Comparing Traditional & Advanced Similarity Metrics with themselves and Monte-Carlo Metric

Goals

- We will be implementing traditional as well as some more novel similarity metrics in order to enhance the accuracy of retrieval and similarity measurement of medicines.

- After comparing multiple advanced similarity metrics and benchmarking them against Monte-Carlo similarity, we will choose the best one.

- We will use FAISS with best similarity metrics selected to search for efficient, fast and seamless recommendation of medicines based on similarity calculation.

Tasks

Our task here is to carry out the comparison of the various similarity scores with respect to each other as well as Monte-Carlo Physarum Machine (MCPM) Metric and thus, select the best out of them for future task and phases. Now below are some of these traditional and advanced similarity metrics (with small description) that I will consider for the task.

- **Cosine Similarity**: It gives more importance to angle between the embeddings rather than magnitudes, i.e., focusses more on how their pattern are.
- **Euclidean Distance:** It just captures absolute distance between the points in the feature space. Traditional but not much insightful.
- **Manhattan Distance:** This metric sums the absolute distance between each feature of the embedding.
- **Tanimoto Coefficient:** Its an extension of Jaccard similarity metric, and is used mainly in cheminformatics and molecular similarity related tasks.
- If time allows, I intend to once check out any one graph-based similarity metrics out of **graphlet similarity (**Probably**)** and **random walk similarity**.

How to Compare which is Best?

- I have planned to take up a manually designed set consisting of similar types of medicines as its subset. Now, I will take up each subset at a time and measure the similarity scores using each of the above similarity metrics and will compare each with results using Monte-Carlo Physarum Machine (MCPM) Metrics.
- Example->
  Set = {{Emb_Med A1, Emb_Med B1, Emb_Med C1}, {Emb_Med A2, Emb_Med B2}}
  Above is set of medicine embeddings having subset of manually grouped embeddings of similar medicine. Now, I have planned to calculate the similarity between medicines of a subset using mentioned similarity metrics to be considered for comparison and finally it will be averaged depending on the number of similarity scores calculated for each subset and it will be done for each method and compared. One having the highest averaged out similarity score will be selected as the best similarity Metrics and would be applied and used for future phases.

- ## Comparison of Similarity Metrics & Setting of Retrieval system powered by the best selected similarity metrics

  - I will use MongoDB database to store all the embeddings in the desired format as required by the database.
  - We can easily use any custom similarity metrics, but here I will go on with the all the similarity metrics one after other that I have to compare.
  - A medicine embedding will be selected manually and passed to the retrieval pipeline which will extract the most similar medicines to it on the basis of similarity search among all embeddings stored within it and it will be done by leveraging the selected similarity metrics which I have provided.
  - One by one, we will use all the metrics and will collect the list of similar medicine mapping each to the similarity metrics used to extract it.
  - Now, all the similarity metrics which retrieved us the most similar medicine to the initially provided medicine embedding will be ranked on the basis of their performance.
  - Finally, considering this ranking and the past method of comparison using set will be considered to finally compare and decide the best metrics.
  - On the basis of this, a retrieval system powered by MongoDB will be built to easily retrieve the most similar medicines by providing a particular medicine's embedding.

- ### Evaluation & Benchmarking

  - From the last step, we already have with the basic comparison report on the basis of relevancy of most similar medicines retrieved but I will enhance the process further.
  - I will further assess retrieval accuracy using Precision, Recall, F1-score and NDCG (Normalized Discounted Cumulative Gain)
  - I will create a detailed comparison report of all these similarity metrics on the basis of three-level comparison and benchmarking against MCPM metric used in Mediglot as all the above testing will be applied to MCPM also.

Novelty

- A robust and comprehensive three-tier designed evaluation and comparison pipeline which will ensure to test metrics on all aspects and deliver the best result.
- MongoDB powered similar medicine Recommendation System powered by the best selected similarity metric to add a feature of Medicine Recommendation in the Mediglot on basis of symptoms, disease, molecules or medicine provided by the user.

Deliverables

- Optimized Similarity Search Algorithms: We will have best-performing similarity metric based on evaluation by comparison and benchmarking against Monte-Carlo similarity metrics.

- MongoDB-Powered Retrieval Pipeline: Efficient and fast recommendation of similar medicines leveraging power of best selected similarity metric.

- Performance Benchmarking Reports: It will have detailed comparison of similarity metrics on all the three level evaluation which they were tested on and comparison report with MCPM metric.

- After these improvements and enhancements, Mediglot will have robust medicine retrieval ability to measure almost error less similarity scores between medicines with even more high confidence than before and will also give accurate recommendation of medicines (on the basis of user query) by leveraging MongoDB based faster retrieval.

Phase 4: Efficient Dimensionality Reduction, Clustering & Visualization in 3D space

Goals

- We will aim to reduce dimensionality of embeddings generated till now to make the process computationally efficient along with less storage requirements.

- We would leverage novel techniques to group similar medicines into meaningful and interpretable clusters.

- Our top priority while reducing embedding's dimensionality would be to maintain high accuracy while reducing only redundant features in embeddings and this may be achieved by calculating covariance scores for various features and also leveraging advanced dimensionality reduction methods.

- We will use MAYAVI framework to embed our medicines as instances in 3D space for deep and scientific visualization with multiple features.

Tasks

- **Applying Dimensionality Reduction Methods and Comparing results with UMAP**

  Dimensionality reduction is important for improving computational efficiency and visualization while keeping the essence and maintaining the intrinsic structure of the original higher dimensional embeddings.

Some Methods which I have planned to experiment with are as follows along with their specialities:

- **t-SNE (t-Distributed Stochastic Neighbor Embedding)** for better non-linear learning and visualization.
- **PCA (Principal Component Analysis)**: Mostly used for linear reduction of dimensions.
- **UMAP (Uniform Manifold Approximation and Projection)** helps to preserve local as well as global structures and context effectively.
- **Autoencoders** acta as a deep learning-based alternative for the non-linear compression of high dimensional embeddings.
- **Isomap** helps in capturing global geometric structure.
- **PHATE (Potential of Heat-diffusion for Affinity-based Transition Embedding)** It captures local as well as global context while denoising and being robust, especially to biomedical data.

After comparing and selecting best out of them, we will apply it to our embeddings and will reduce them from 1024 to lesser dimensions (not 3D for now as 3-D is required for visualization and not for clustering) as to calculate similarity score between medicines and help users get similar medicines by faster retrieval.

How to Compare which is Best?

- I have planned to again take up a manually designed set consisting of similar types of medicines as its subset. Now, I will take up each subset at a time and measure the similarity scores using MCPM metric as well as the best similarity metric selected in the last phase.
- After its done, I will average out the similarity scores for each subset for each of the above dimensionality reduction methods, and will compare that which method has obtained highest similarity score.
- Example->
  Set = {{Emb_Med A1, Emb_Med B1, Emb_Med C1}, {Emb_Med A2, Emb_Med B2}}
  Above is set of medicine embeddings having subset of manually grouped embeddings of similar medicine. Now, I have planned to calculate the similarity between medicines of a subset using MCPM metric and the best selected metric from phase 3 for each of the dimensionality reduction method to be considered for comparison and finally it will be averaged depending on the number of similarity scores calculated for each subset and it will be done for each method and compared. One having the highest averaged out similarity score will be selected as dimensionality reduction method for all the data embeddings.

Enhancing Clustering by using some Advanced & Traditional Methods

Clustering targets to group and bring together similar medicines on the basis of their embeddings and at the same time filters out the noise. It helps us to identify groupings, complex relationships and underlying patterns within the dataset which may not get observed through manual inspection.

Unlike traditional clustering methodologies, which assumes that clusters are well separated and has well defined hard boundaries between them, advanced techniques (like GMM) mostly use better approaches to have clusters having more varied shape and soft boundaries.

Below are some of the methods (with short description of them) which I have planned to experiment with and figure out the best from them.

- **Hierarchical Density-Based Clustering (HDBSCAN)** specializes in handling clustering of noisy data as well as variable density clusters.
- **K-Means** helps creating balanced and compact clusters by leveraging dynamic selection.
- **Agglomerative Hierarchical Clustering** can helps find out the hierarchical relationships between medicines.
- **DBSCAN**: It's a density-based clustering for identifying outliers and core points. We will try out this also.
- **Gaussian Mixture Models (GMM)** It's an unsupervised soft clustering algorithm which works by representing each cluster with gaussian distribution and points are assigned probabilities of belonging to clusters. It uses parameters like Mean, Covariance and mixing probability and these parameters get updated iteratively until these stops changing significantly.

- **Evaluation Metrics**: We will use traditional as well as some task specialized evaluation metrics to assess quality of clustering and effectiveness of dimensionality reduction.
  - After diving deep, I have planned to use **Silhouette Score** to measure compactness of clusters.
  - We will use **t-SNE** Visualization in 2D space to visualize structure which will give us how better points are grouped.
  - I will also go for **Davies-Bouldin Index (DBI)** to evaluate similarity between different clusters in the graph and it's like less similarity score means better clustering.

The combination of above evaluation metrics is capable of evaluating clustering methods on every aspect be it compactness or softness of boundaries. So, I will check each clustering techniques using above testing combination and will select the best.

- After choosing the best clustering algorithm, I will go on for applying the method on our embeddings and for better visualization, I will provide each cluster a color code and tags with human-interpretable names (e.g., "painkillers", "antibiotics", etc.). Though it has to be done manually, but I will definitely try to do it.

- **Plotting instances in 3D using MAYAVI:** I will use Mayavi framework to plot our embeddings in 3D space for better scientific visualization and engaging visual representation of medicines.
  - Embeddings would be converted in NumPy based desirable format as required by 'mayavi' for input.
  - Color Coding assigned to each cluster in above step will be incorporated for clear picture of clusters.
  - I will add Zooming, rotation and other engaging features to make it more insightful and efficient.

Benefits of Dimensionality Reduction & Clustering

- **Computational Efficiency**: High quality embeddings get generated while reducing computations and storage requirements at the same time.

- **Improved Interpretability**: Clustering helps in identification of groups of similar properties containing medicines which helps out in getting similar medicines on the basis of a particular feature of the cluster and it also helps medical professionals in finding out similar medicines.

- **Better Retrieval Performance**: Faster similarity searches take place due to low dimensional embeddings as well as searching the similar cluster instead of whole dataset making it perfect for real-time recommendation systems.

- **Noise Reduction**: Increases the robustness of the model by discarding or removing the noises.

Benefits of Tools & Techniques

- **UMAP & t-SNE**: These techniques are specialized in capturing the non-linear relationships obviously better than PCA, thus preserving local and global structures and contexts effectively.

- **Autoencoders**: It compresses the features and keeps only important features by leveraging the deep learning techniques.

- **HDBSCAN**: It helps in pointing out meaningful subgroups in the noisy data just by handling clusters with variable density.

- **Silhouette Score** & **Davies-Bouldin Index (DBI)**: It ensures optimal cohesion and separation of clusters. These are validation metrics that are used to evaluate clustering models.

- **Multiple Similarity Metrics**: We can identify best metric for clustering by using each Cosine, Euclidean, and Manhattan distances.

- **t-SNE Visualization**: It will help us get insights that we would be able to interpret (human interpretable insights).

Deliverables:

- **Optimized Embeddings**: Lower-dimensional representations of medicines with no to very less information loss.

- **Clustered Medicine Groups**: This will help us get similar medicines on the basis of feature of the other medicines in the cluster, thus helping in recommendation.

- **Visualization Plot:** An interactive and engaging 3D plot of all medicines representing a clear picture of clusters and relationships between medicines.

- **Performance Reports**: After all this, we would have with us best selected clustering technique along with detailed evaluation and comparison of clustering techniques and methodologies which will direct and mentor us in present coming phases and for future developments also.

Phase 5: Final Optimization, Testing & Integration

Goals

- Integration of all the pipeline for seamless visualization and similarity calculation of medicines.

- Optimization model performance in terms of speed, accuracy, and scalability.

- Final validation and testing will be conducted manually to ensure correctness and reliability.

- Documentation of the entire project and Comparison and Performance Report will be generated which will guide us in our future tasks and plans.

Tasks

- **Performance Optimization**

  - I will focus mainly on reducing the latency of the Recommendation system designed.
  - We will try out for FAISS powered Recommendation system leveraging the ANN search method for faster retrieval of medicines on the basis of similarity search.

- **Integration and Deployment**

- I will enhance original dataset with the extra pharmacological, structural, and other information collected about medicines for future uses.
- Improvement of the current visualization on the Mediglot if researched methods perform better.
- I will develop API using FastAPI for Recommendation System developed and will integrate it on the Mediglot page for users to directly get recommendations of similar medicines.
- Any other required web integration will be carried out as per the needs.

- **Wrap-up & Documentation**

  - I will design a well detailed and comprehensive documentation of all the above tasks, plans, and results achieved with some visual materials attached for better understanding.
  - A comparison report will be created on the performance of each dimensionality reduction methods and similarity metrics tried out with respect to UMAP and MCPM metric.

Final Deliverables

- Comprehensive dataset with added information of pharmacological (Uses, working mechanism), descriptional (Common description), structural (molecular structure) and compositional details (salts used) of each medicine by information achieved by extraction from various sources like PubChem, 1mg, ChemBL and so on.

- Highly-Informative and biologically much insightful and GraphSAGE refined medicine embeddings using Biomedical based LLMs like and transformer-based models.

- Robust clustering mechanism implemented for efficient grouping of medicines figured out after deep comparison and validation of traditional as well as advanced clustering methodologies, thus resulting in almost error less clustering of medicines.

- Implementation of best similarity metric selected using three tier testing and validation pipeline to ensure near-zero error in similarity calculations.

- Enhanced 3-D visualization for advanced and deep scientific study of medicine relationships and similarities.

- Accurate and High-speed retrieval and Recommendation system based on similarity score calculation by best performing similarity metric (selected after multiple aspects of comparison with MCPM) powered by FAISS for real-time medicine suggestions.

Comprehensive Documentation

- Well-structured project documentation covering implementation details, methodologies, and best practices.

- Well detailed and Comprehensive report of comparison between Mediglot's current dimensionality reduction and similarity metric with advanced and some traditional techniques and methodologies.
- Detailed future development roadmap outlining potential enhancements and scalability options.
- Performance benchmarking reports comparing different techniques used in the system.

Work done till now (Pre-GSoC)

- **I have been working on a sample subset of the original MediGlot dataset, consisting of approximately 50–100 data points, to conduct preliminary experiments and research. The objective of using this smaller dataset is to gain a clearer understanding of the optimal techniques, suitable LLMs, and reliable medical knowledge sources that will be used for the project. This exploratory phase will ensure that our approach is well-informed and effective before scaling to the full dataset.**

- First, I tried to extract **pharmacological** and **description data** of medicines from various sources like PubChem, ChEMBL, and so on but no place has data about specific medicine. So, I went for scrapping data from "**1mg**" by designing a pipeline to do so and I did it using **Selenium, BeautifulSoup** and **Chromedriver.**
- This helped me get **'Uses'** and '**working mechanism**' of the drug which I mentioned in two separate columns in dataframe.
- Then I went to **PubChem** to extract the **molecular structure** of the drug, but this was not possible as drug information were not there, so I extracted salt composition of each drug which is also present in our dataframe and then scrapped SMILES molecular structure of drugs (eg. : C1=CC(=CC(=C1NC(=O)C)[2H])O and added it to another column named **'Structure'**. Now at this, we have a comprehensive and insightful dataframe having '**Medicine_name', 'Salts', 'Working', 'Uses'** and **'Structure'.**
- Now at this point, I had a comprehensive and insightful dataframe having '**Medicine_name', 'Salts', 'Working', 'Uses'** and **'Structure'.**

  Notebook1 (Enhancing_Medicinal_Information): visit(Click)

- Now, main task is to **generate embeddings** but since I have many columns (features/information of medicine) in our dataframe, so each represent different type of information and has its own significance, thus, a single biomedical LLM can't be used to generate embedding for each. So, I selected different models for different features/columns.
- For columns:
  ----Uses, Working and Salts ---> "**BioLinkBERT-large**" (length of embeddings = 1024)
  ----Structure ---> "**ChemBERTa**" (length of embeddings = 768)

- After embedding generation of each feature, I made them go separately under **L2 normalization** and got stored in columns named "..._normalized"
- Now, since their length of embedding is different, so I projected them to same length of 1024 using **Multilayer Perceptron network** to avoid truncation or up sampling which may cause information loss and irrelevance information addition in embeddings**.** And normalization is done for each of four column (or feature) embeddings separately.
- Embeddings of all the features of medicines (salts, uses, working and structure) have been made to same dimension of 1024 and normalized for efficient concatenation. So, **Multi-Layer Perceptron (MLP**) has been used as replacement of zero padding which could have added irrelevancy in our embeddings. Now, since all embeddings are 1024 dimensions, so weighted concatenation has been implemented as all four features has different significance and contributes differently to the final embedding that will be generated, so below is the assigned weight which I chose:
  w1 = 0.35 for Salts
  w2 = 0.30 for Uses
  w3 = 0.25 for Working
  w4 = 0.10 for Structure (MLP Up-Sampled)
  Finally, I have concatenated embeddings of all medicines having pharmacological, structural and salt composition which makes its biologically more insightful and robust.

  Notebook 2 (Embedding_Generations) Additional work : **Visit(Click)**

- Now comes the Refining of final embeddings that ware generated in the last step after weighted concatenation of all the features of medicine. The idea is to advance each node’s embedding by aggregating information from its neighboring nodes. This aggregation ensures that the GraphSAGE learns the representations that capture the contextual relationships between medicines.
- As mentioned in proposal, Embeddings has been refined using Unsupervised Learning of Graph Convolutional Neural Networks (GCNNs) and then also compared with refinement using **GraphSAGE** and I found that, GraphSAGE is performing better as it seemed more related to the input embeddings on the basis of similarity measurements, so I considered GraphSAGE refined embeddings as final and discarded GCNN’s refined embeddings.
- I now have the refined and comprehensive embeddings of medicines having biological insights about molecular structure, salt composition, working mechanism, and uses and these embeddings are neighborhood aware as it is important for better clustering since we need to know the relation of medicines with other medicines (We can take an analogous case, suppose we have to find two friends, then apart from their character similarity, we would also go for checking that how many mutual friends they have)

  Notebook 3 GCNN_50_Refined_Embeddings): **Visit(Click)**

Now, next task is to scale the above steps to all the points in the Mediglot's dataset and then from here, I will continue with the full dataset having refined and comprehensive embeddings.

Timeline/Project Plan

After thoroughly reading the GSoC 2025 timelines, I have planned to carry out five phases of the project with a time allocation designed to guarantee the hassle-free completion of each phase within its time limit.

Time Frame	Start Date	End date	Task
Community Bonding	8th May	1st June	Discussion about the new ideas, suggestions, and getting into implementation details with mentors.
Coding begins Officially			
Phase 1	2nd June	3rd June	On the basis of work done till now, I will Scale the Data collection, Analysis, preprocessing to full dataset.
	4th June	7th June	Optimizing Pipeline built (as pre-GSoC work) for extracting Molecular and pharmacological information of medicines from sources such as 1mg, DrugBank, PubChem, and ChEMBL as well as from Biomedical LLMs like BioLinkBERT-large, ChemBERTa, etc. From the experiments performed as of now as Pre-GSoC work, I will directly go on with above mentioned sources and LLMs confidently.
	8th June	10th June	Just trying out some more LLMs like BioBERT, MedBERT(on full dataset) for embedding generation and comparing them with BioLinkBERT-large, ChemBERTa (which I selected for now).
	12th June	13th June	Moving on with ensemble method involving multiple Biomedical based LLMs finalized for embedding generation and weighted concatenation of them. Final comparison with MedBERT embeddings.

	14th June	15th June	Final embedding generation for all medicine having information like slat composition, molecular structure, pharmacological descriptions like action mechanism, uses, target molecules, and so on.
Phase 2	16th June	18th June	PolyPhy inspired network graph construction with medicines as nodes and relationships between them as edges (on the basis of embeddings) by leveraging Tanimoto coef., Person correlation and cosine similarity to capture structural and pharmacological information.
	19th June	22nd June	Designing GraphSAGE layers and some optimizations (as GraphSAGE is selected over GCNN in Pre-GSoC work.)
	23th June	24th June	Unsupervised training of GraphSAGE begins with graph as input to it.
	25th June	27th June	Altering the convolutional layers, trying out with various loss functions and optimizers to get better performance.
	28th June	29th June	Finally, getting all the refined embeddings from our trained and optimized GraphSAGE. Now, we have refined embeddings of all data instances.
Phase 3	30th June	3rd July	Implementation of all traditional and graph-based similarity metrics as mentioned.
	4th July	7th July	Comparison of all of them among themselves and with MCPM metric and choosing the best on the basis of manual supervision on self- designed three-tier evaluation bed.
	8th July	10th July	Storing all embeddings in MongoDB powered by the above selected similarity metrics and implementing retrieval pipeline on the basis of similarity score from the datastore.
	11th July	13th July	Evaluation of the above traditional and novel similarity metrics on various aspects and comparison with MCPM metric on the basis of retrieval quality which has to be done manually.

Midterm Evaluation	14th July	18th July	
Phase 3 Continues	19th July	21st July	Selecting the best similarity metrics and wrapping up this phase.
Phase 4	22nd July	24th July	Designing and unsupervised training of Autoencoder-Based Neural Network for dimensionality reduction.
	25th July	28th July	Trying out various traditional techniques like t-SNE, UMAP, Isomap and PHATE for dimensionality reduction and comparing them with themselves and our autoencoder based approach.
	29th July	30th July	Optimizing the Autoencoder and comparing and selecting the best performing dimensionality reduction method.
	1st Aug	4th Aug	Applying clustering techniques (like HDBSCAN, Agglomerative Hierarchical clustering, K Means, GMM) and comparing them over DBI and silhouette score, and t-sne visualization and to select out best of them. I will provide each cluster a color code as well as tags with human-interpretable names (e.g., "painkillers", "antibiotics", etc.). And this has to be done manually.
	5th Aug	6th Aug	Transforming and transferring the clusters into the desired format and using the MAYAVI Framework to plot the points in 3D space for better scientific and deep visualization of medicines.
	7th Aug	8th Aug	Introducing some cool features like zooming, rotation, color change, etc. into Mayavi visual space for better and engaging visualization.
Phase 5	9th Aug	10th Aug	Reducing the latency of Recommendation system designed and setting and comparing up FAISS powered by Approximate Nearest Neighbour (ANN) performance with our selected similarity metrics and selecting out best of them for Medicine Recommendation system.

	10th Aug	11th Aug	Developing FastAPI for Recommendation pipeline and integrating it with Mediglot page.
	12th Aug	18th Aug	Final validation and testing. Complete any follow-up or collaborative projects regarding it and any pending work or follow-ups task from the above tasks.
	18th Aug	25th Aug	Final comprehensive and robust documentation and comparison report generation with Mediglot's methodologies and techniques in use currently. Sharing Post GSoC plans with mentors and gathering suggestions.
Final Mentor Evaluation	26th Aug	1st Sep	

Week-Wise time distribution for better traceability

- **Week 1-2**: Data Collection and Preprocessing, LLM selection and embedding generation on the basis of Pre-GSoC tasks along with dimensionality up-sampling using MLP and then weighted concatenation(Phase 1).

- **Week 3-4**: Building similarity graph and training GraphtSAGE (decided on the basis of Pre-GSoC work) to get refined embeddings. (Phase 2).

- **Week 5-6**: Implementation and comparison of advanced as well as traditional similarity metrics among themselves as well as with MCPM metrics. (Phase 3)

- **Week 6**-7: I will set up MongoDB based and similarity metrics powered retrieval pipeline and will test these similarity metrics on the basis of retrieval quality. (Phase 3)

- **Week 8-9**: Apply and compare various listed dimensionality reduction techniques and would finally choose the best. After it, Clustering methods will be implemented and compared on the designed evaluation framework and best out of them will be selected. Now, I will go on with visually representing embeddings in 3D space using 'mayavi'. (Phase 4).

- **Week 10-12**: Setting up FAISS based ANN powered medicine Recommendation Pipeline for similarity metrics comparison with ANN and Final testing, detailed documentation and comparison of past and current performance of system in each aspect and then, wrap-up (Phase 5).

Commitments

In weekdays (Monday through Friday), I can easily put **5-6 hours** to project work, ensuring alignment with UTC, MDT (or any other, if required) working hours which will allow for efficient communication and collaboration. In order to compensate for the reduced working hours during weekdays, I'm happy to put extra hours on weekends (Saturday and Sunday) – around **7-8 hours** per day. And this extended weekend commitment will make sure that I meet the proposed timeline while keeping in mind the time zone difference.

Additional Information about the Timeline

- The timeline mentioned above is subject to change and is only an approximate outline of my project work. I will stick to or exceed this schedule and create a more detailed schedule during the pre-GSoC and community bonding phase.
- **I've no other commitments during the summer** and can dedicate **40 to 45 hours a week**. During the last month of the project, my college will begin, and I'll be able to commit a max of **25 hours a week**. Due to the same, I will do a significant portion of the work before this period.
- Time will be divided (according to workload) each week amongst planning, learning, coding, documenting and testing. All documentation will go hand in hand with the development.

POST GSoC PLANS

I am committed to the long-term sustainability and success of the MediGlot project. With the initial scope of this proposal primarily aims at developing a robust medicinal system with advanced embeddings, clustering, and retrieval techniques (i.e., similarity measuring), I recognize the potential for further enhancements and advancements.

- I will like to experiment with integrating real-time patient data to provide personalized medicine recommendations which will be based on the medical history and symptoms of the patients and these would be used to retrieve the most relevant medicine and would also like to build API and UI for real time medicine recommendation and comparison system.

- Additionally, I aim to refine the explainability module, ensuring that users receive clear justifications for recommendations by leveraging explainable AI (using LIME, etc.).

- Another potential improvement which I will like to do is developing seamless user-friendly web interface to make MediGlot more accessible to healthcare professionals and researchers.

By aiming for these enhancements, I plan and hope for the interpretability, scalability and high impact value of Mediglot beyond GSoC timeline.

Final Notes

I assure you that if I get selected to work with UCSC OSPO this summer, I will surely try my level best to make this project a great success and will love to continue working with USCS OSPO in other project even after the summer. I would really love to work under such great mentors of UCSC OSPO and it would be a golden opportunity for me to contribute to Open-Source community under the guidance of my mentors. This will provide me a platform to collaborate, brainstorm and work together with the world level developers who will be there guiding me as my mentor throughout GSoC 2025.

Also, for some reason, If I am not selected this year even then I will try to contribute to this and other projects as much as possible and retry again next year.

Looking forward to working with you

Thanks and Regards

Ayush Shaurya Jha