Inclusivity in Job Recommendation: Recommending Jobs for Differently-Abled Individuals through Feature and Transformer based Approaches

Thesis Stage 3 Presentation

June 20, 2022

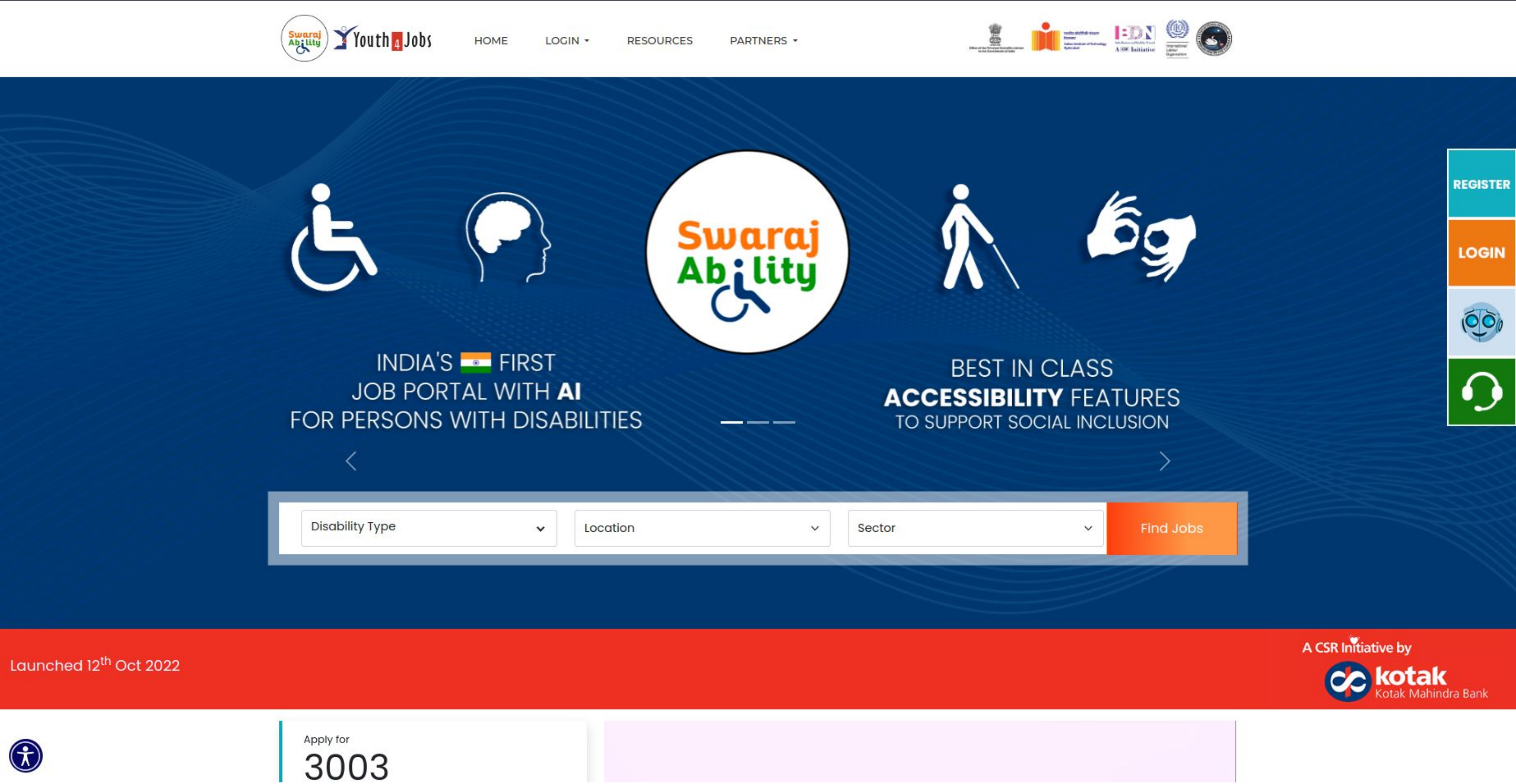
Kamal Shrestha, CS21MTECH16001

Thesis Supervisor : **Dr. Maunendra Sankar Desarkar**

Department of Computer Science and Engineering (CSE)
IIT Hyderabad

Overview

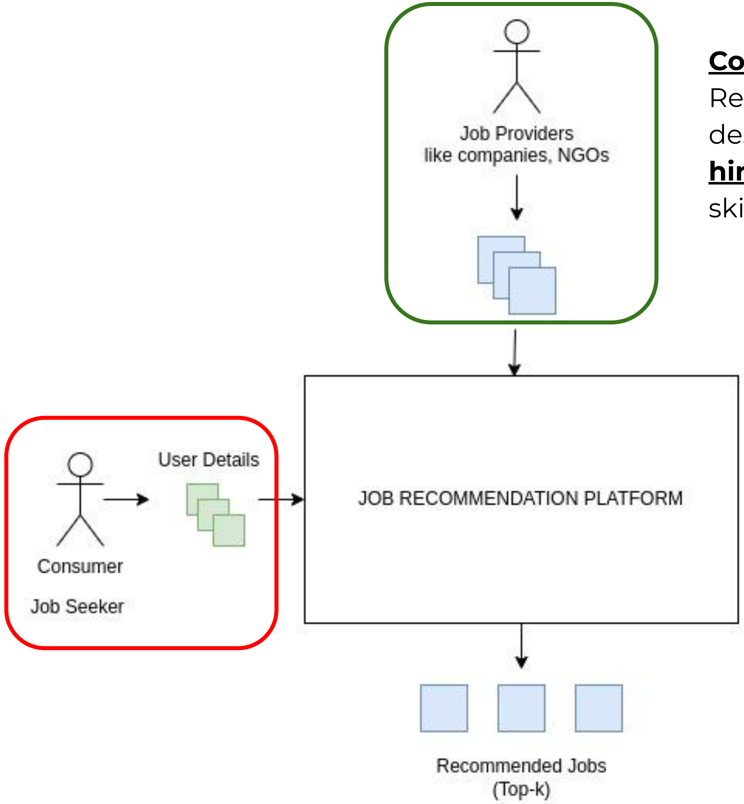
Overview



SWARAJIBILITY.ORG

C2C Based Jobs Platform connecting PwDs with Job Opportunities

Companies or other job providers
Register for odd jobs (like working on front desks, assistants, guards, grassroot fellows) **to hire** people based on their requirements and skills



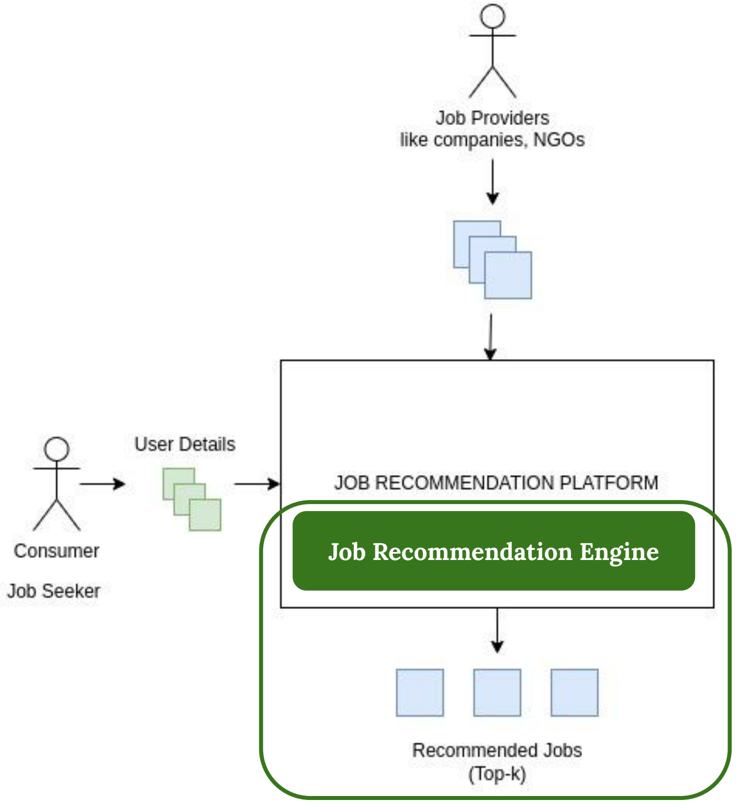
Consumer or job seekers
Find appropriate jobs based on multiple features and preferences like disabilities, skills, locations, experiences and training **to get hired**

Given a job-providing platform that focuses on connecting individuals with disabilities to odd jobs, **this work focuses developing a recommendation system**

Reliability

Quick Response Time (Low Latency)

Dynamic



Incremental

Minimal Space Requirements

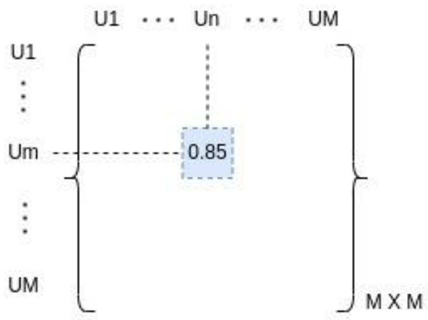
Automated

Overview

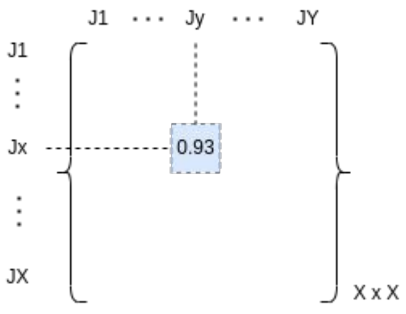
Hybrid Recommendation System (HRS) with pre-computed similarity matrices

CandidateDetails	
FIELD NAME	**DATA TYPE**
CandidateId (Primary Key)	int identity
DateOfBirth	datetime
Address	nvarchar
PreferredJobLocation	int
DisabilityTypeId	int
Trainings	nvarchar
SkillsCovered	nvarchar

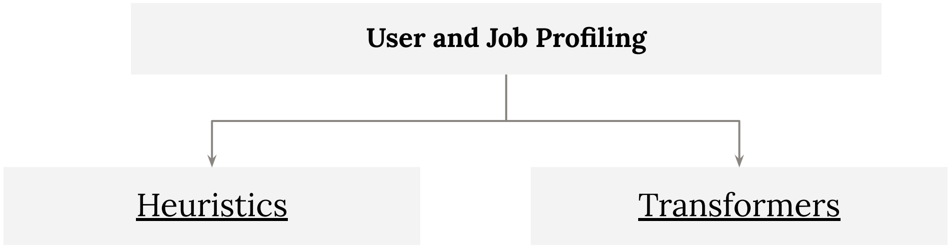
JobOpenings	
FIELD NAME	**DATA TYPE**
JobId (Primary Key)	int identity
JobTitle	nvarchar
JobDesc	ntext
JobType	tinyint
Experience	tinyint
JobLocation	int
NoOfVacancies	smallint
DisabilityTypeId	tinyint
startdate	datetime
enddate	datatime
SkillSet	nvarchar



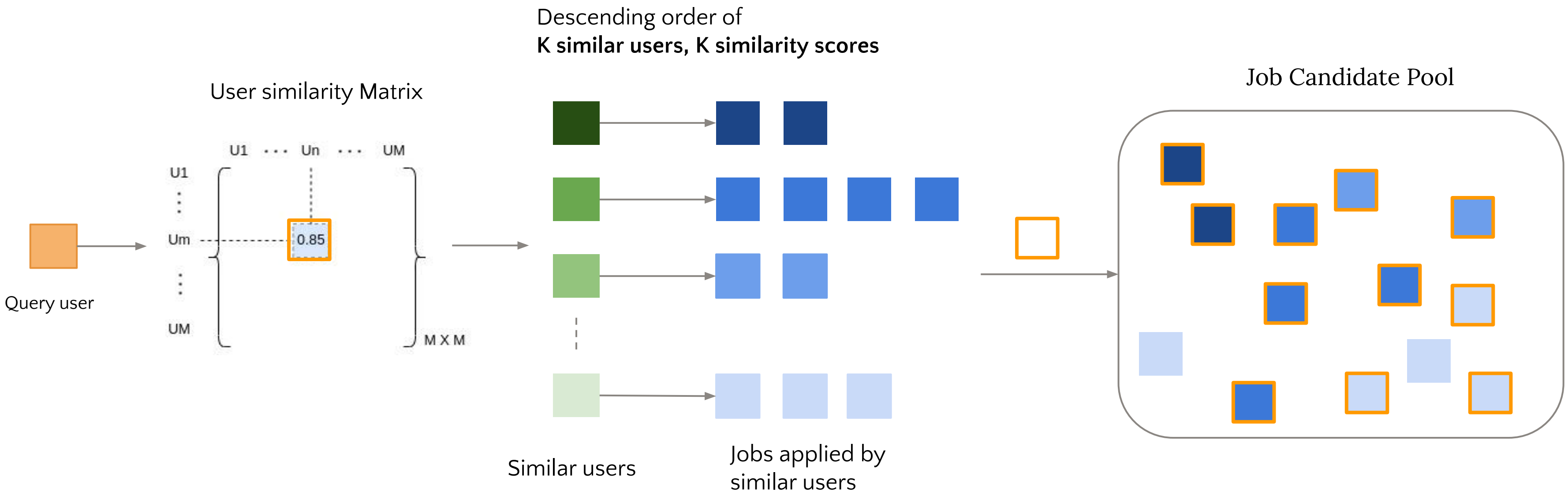
User - User Similarity Matrix



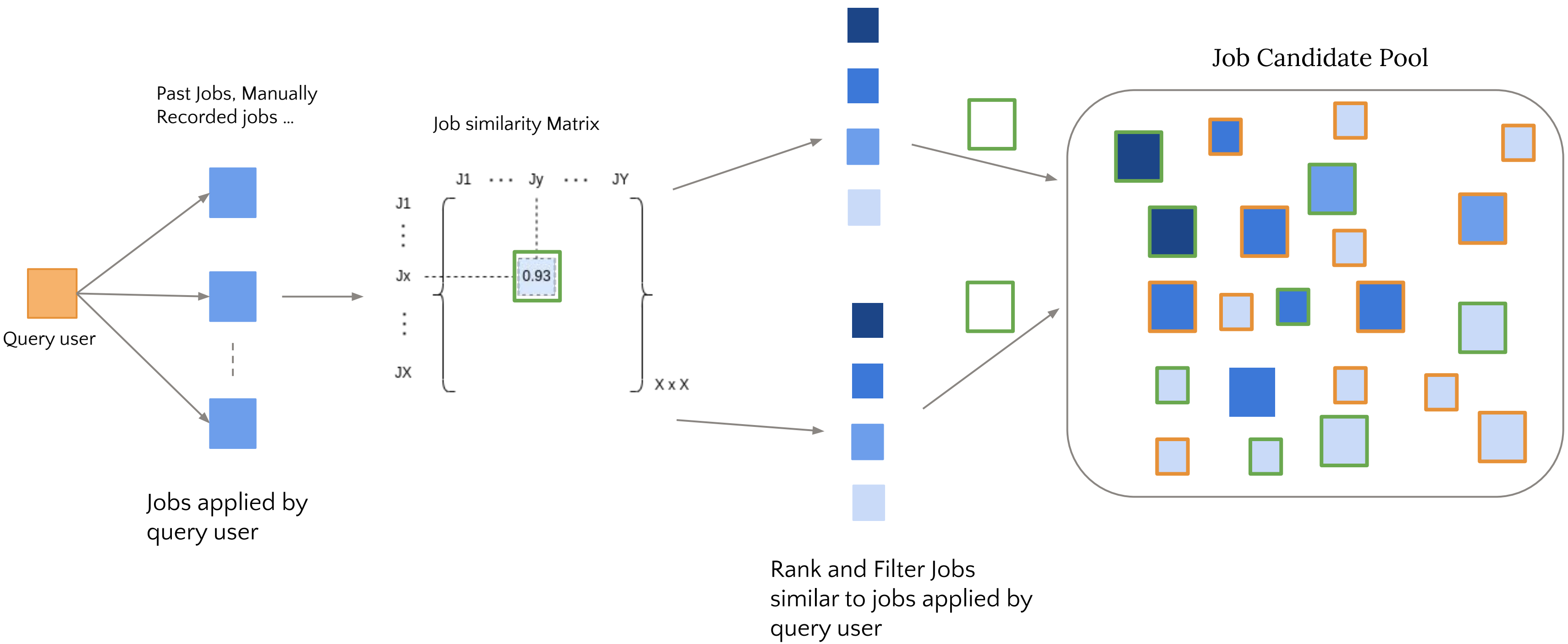
Job - Job Similarity Matrix



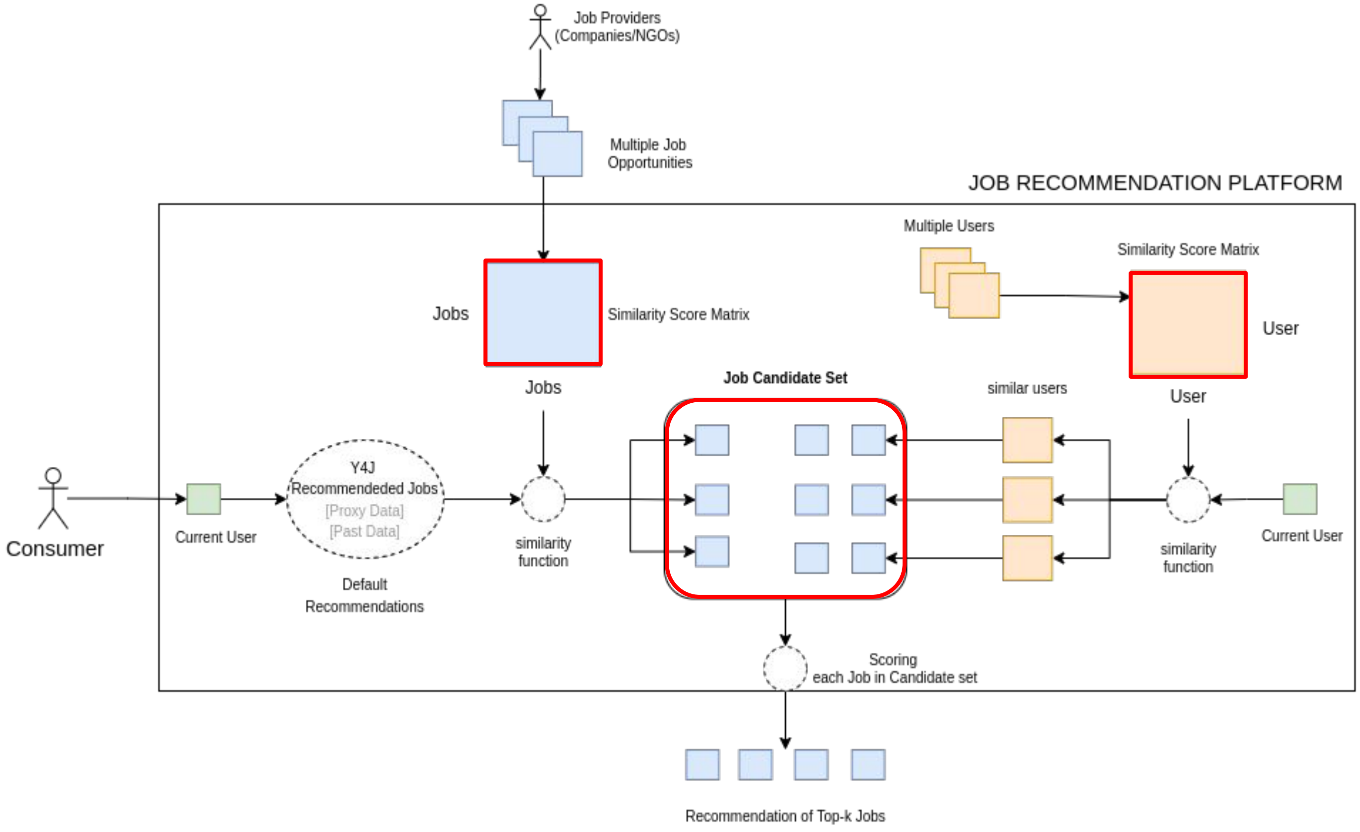
Candidate Job Pooling based on **User similarity Matrix (U2U)**



Candidate Job Pooling based on Job similarity Matrix (J2J)



Combine candidate pool and recommend Top-K jobs



Job Recommendation Engine V1.0

❑ **Input:** User ID (q), No. of Recommendations (k)
❑ **Output:** k recommendations

❑ **Factors considered in recommendation:**

- Disability(ies)
- Location
- Skills
- Age
- Jobs applied to by similar users
- Jobs similar to jobs that are applied to by the same user

❑ **Sample API call:**

❑ Recommendation: [URL]/recommend?q=ID&k=num

❑ Addition of Users: [URL]/addUser

❑ Addition of Jobs: [URL]/addJobs

Synthetic Input Data
Based on statistics

[ARRAYS and LISTS]

Heuristics Approach
Similarity Matrix Calculation

API Endpoints
System Integrations

Similarity calculations based on individual features in heuristics approach

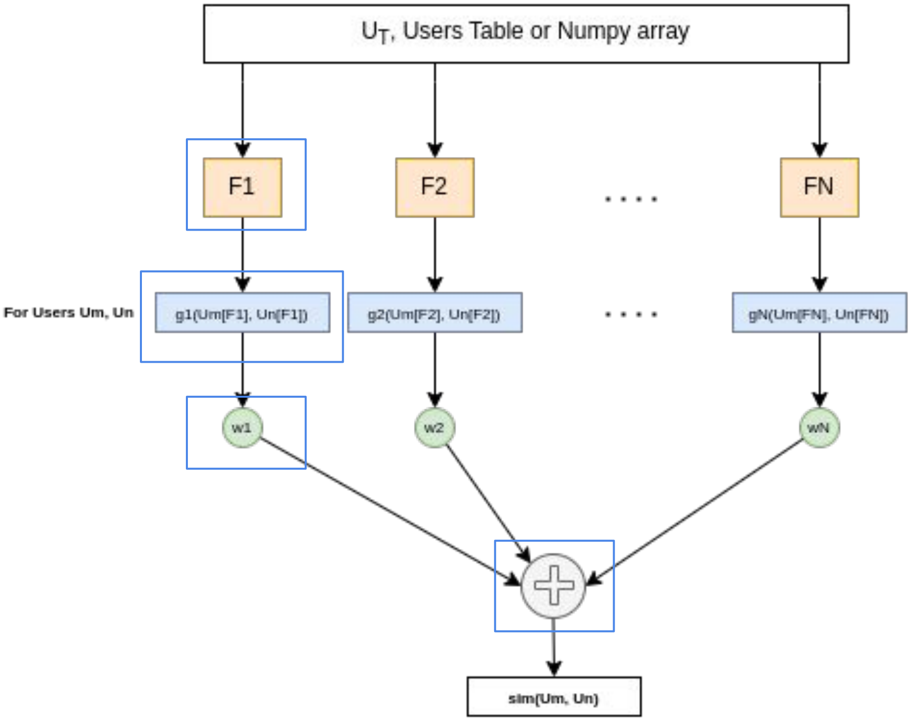
Algorithm 1 Heuristics Based Similarity Matrix Calculation Algorithm

Require: Users, U = $\{U_1, \ldots, U_M\}$
Require: Columns (field names) for similarity calculation b/w users
Require: Column weights, $W_c = \{w_{c1}, w_{c2}, w_{c3}, \ldots\}$

```
matrix = {}
for each User, U_i do score = 0
    for each User, U_j : i ≠ j do
        for each column, c_k in Columns do
            score += W_c[c_k] * SimilarityFunction(U_i[c_k], U_j[c_k])
        end for
        Normalize score by dividing it by the number of columns
        matrix[U_i][U_j] = score
        matrix[U_j][U_i] = score
    end for
    matrix[U_i][U_i] = 1
end for
```

// Similarity function differs based on each column; Jaccard Similarity is used for columns with multiple values, like disability, skills, and talents.

// For columns with only one text value, Fuzzy String Matching with Token Sort Ratio is used.



$$sim(U_m, U_n) = \sum_{f in F} w_f * g(U_T[U_m][f], U_T[U_n][f]) \forall (m, n) \in [U_1, U_M]$$

Job Recommendation Engine V1.0

- **Input:** User ID (q), No. of Recommendations (k)
- **Output:** k recommendations

- **Factors considered in recommendation:**
  - Disability(ies)
  - Location
  - Skills
  - Age
  - Jobs applied to by similar users
  - Jobs similar to jobs that are applied to by the same user

- **Sample API call:**
  - Recommendation: [URL]/recommend?q=ID&k=num
  - Addition of Users: [URL]/addUser
  - Addition of Jobs: [URL]/addJobs

Synthetic Input Data
Based on statistics

[ARRAYS and LISTS]

Heuristics Approach
Similarity Matrix Calculation

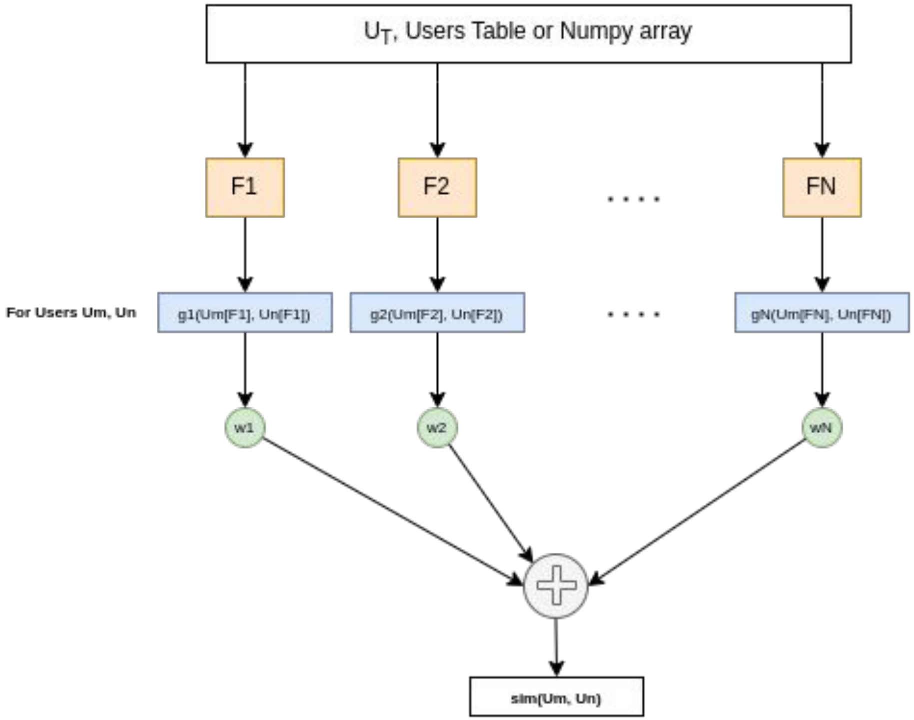
API Endpoints
System Integrations

Observations in Heuristic Approach

Zero similarity:
Handle Missing values
OR
Missing values resulting in 0 similarity score

Fixed Feature Weights: Might lead to feature bias

Limited Personalization: Each feature is independent
Does Not consider the correlation between features like disability and skills



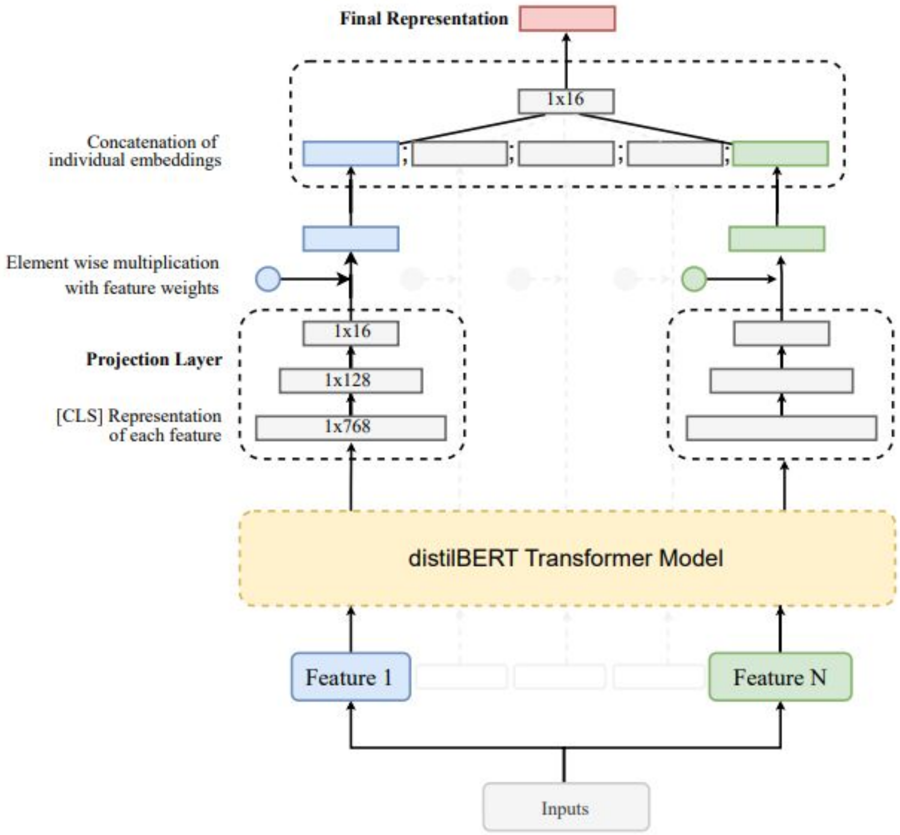
Overview

Transformer (Custom-distilBERT) based learning approach

Algorithm 2 Transformer Based Similarity Matrix Calculation Algorithm

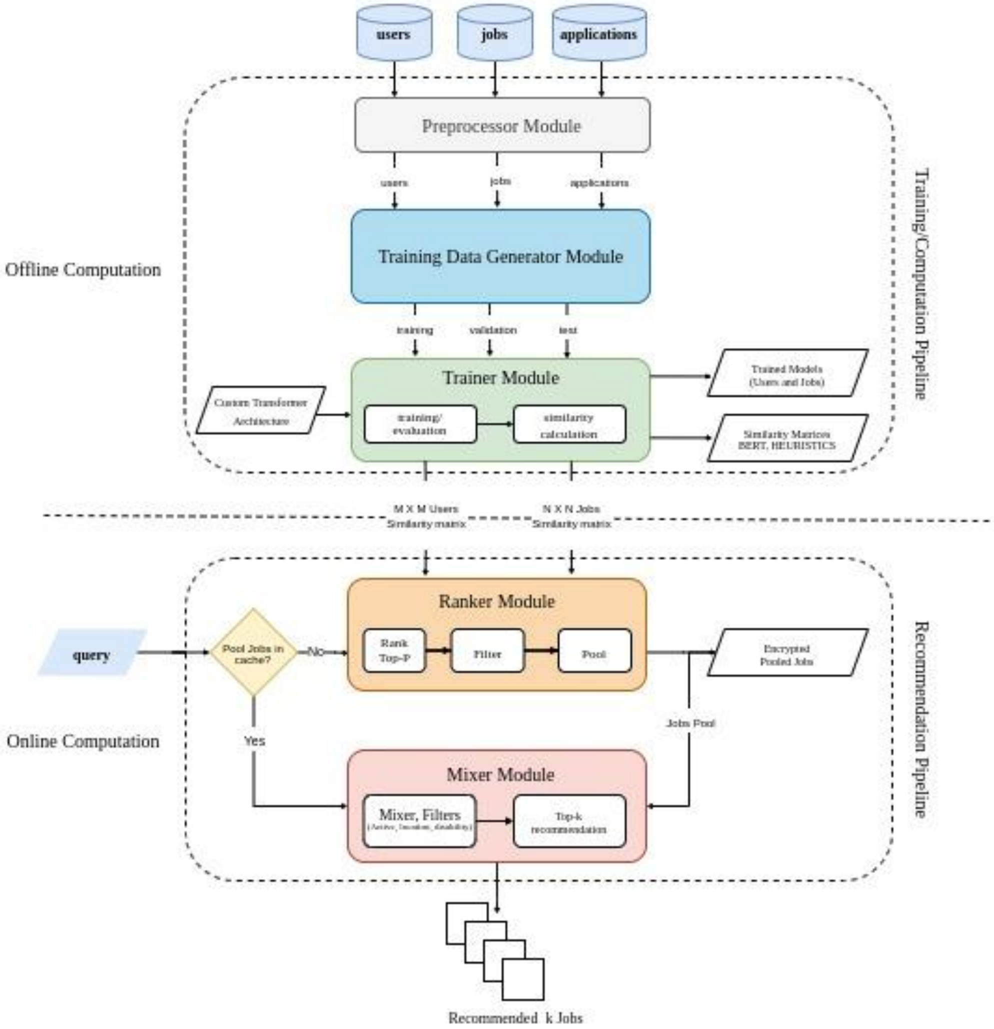
Require: Users, U $= \{U_1, \ldots, U_M\}$
Require: Columns used for similarity calculation between users
Require: Trained custom architecture (CustomBERT) model weights

set matrix = {}, embeddings = {}
Load the saved trained model weights *// Should match with custom architecture*
// Generate the embeddings for all users
for each User, U_i **do**
  for each column in Columns **do**
    Tokenize column, c_k
  end for
  Combine the column tokens to get T_i for user, U_i
  Pass T_i to the CustomBERT & get the embeddings for user, U_i
  set embeddings[U_i]
end for
// Calculate the similarity matrix
for each User, U_i **do**
  for each User, $U_j : i \neq j$ **do**
    score = CosineSimilarity b/w embeddings[U_i] and embeddings[U_j]
    set matrix[U_i][U_j] = score
  end for
  set matrix[U_i][U_i] = 1
end for

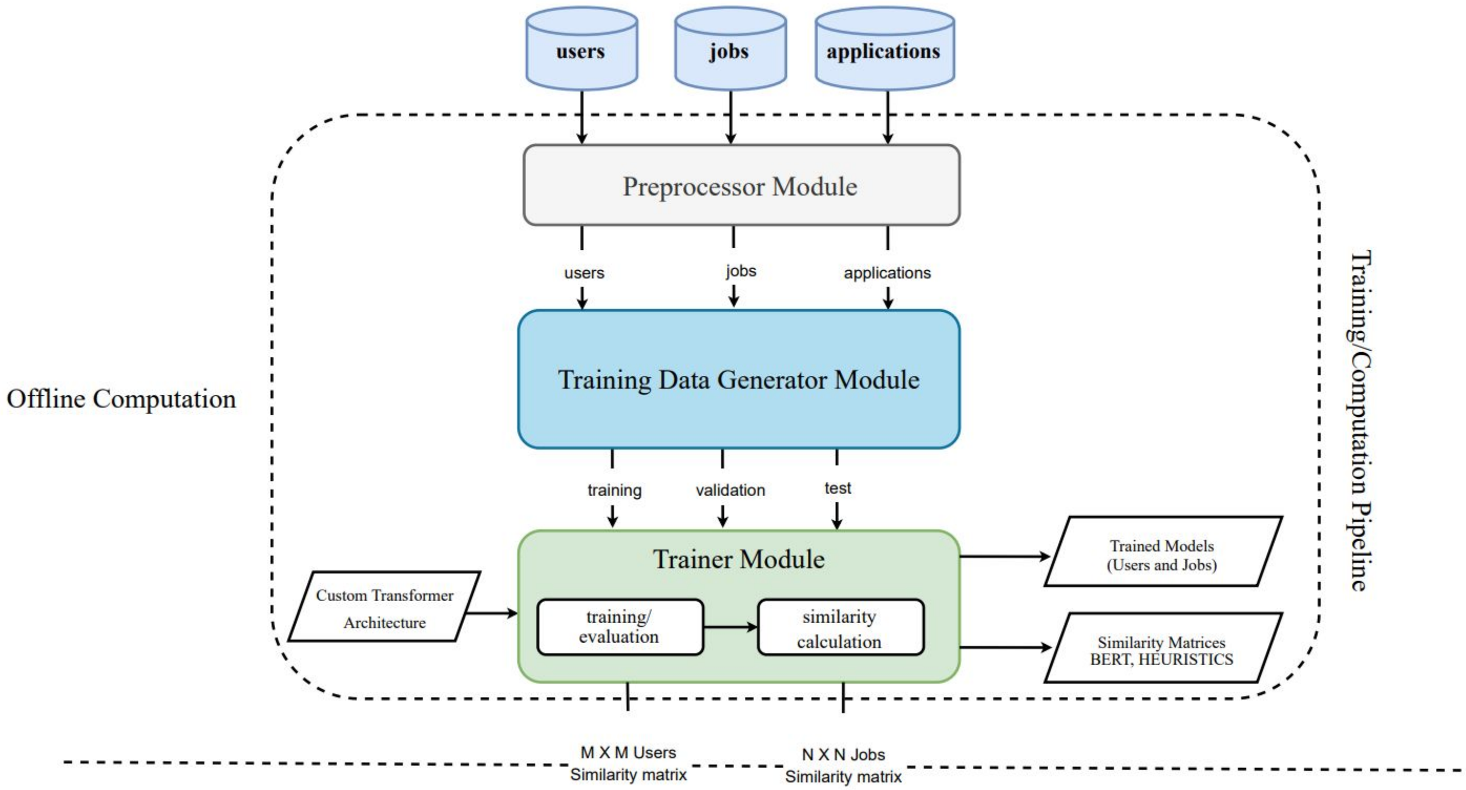


Overview

System Architecture



Training and Computational Pipeline - OFFLINE

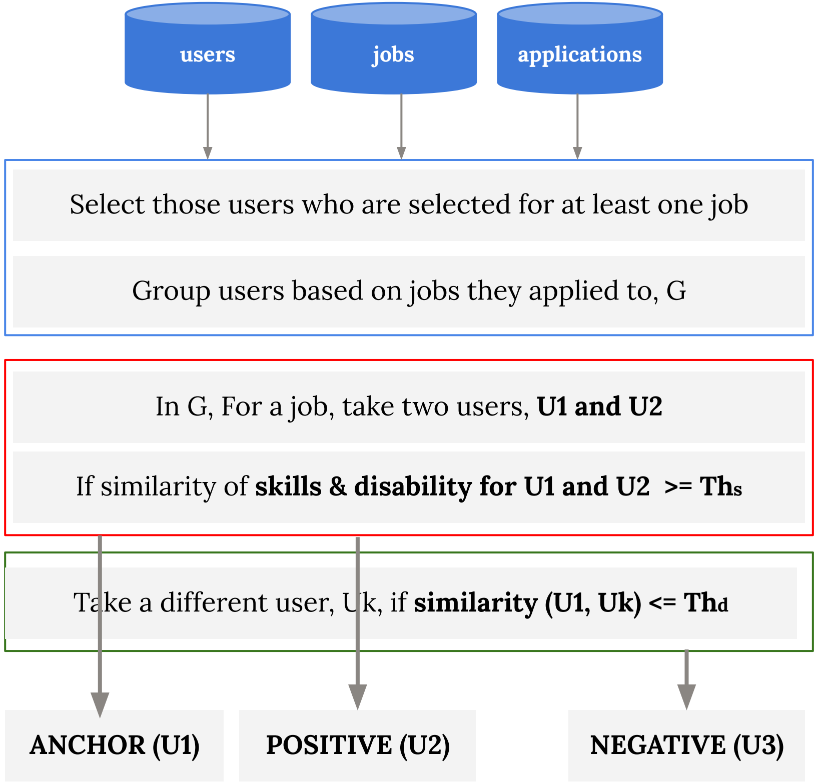


Training Data Generation - Triplet Pairs

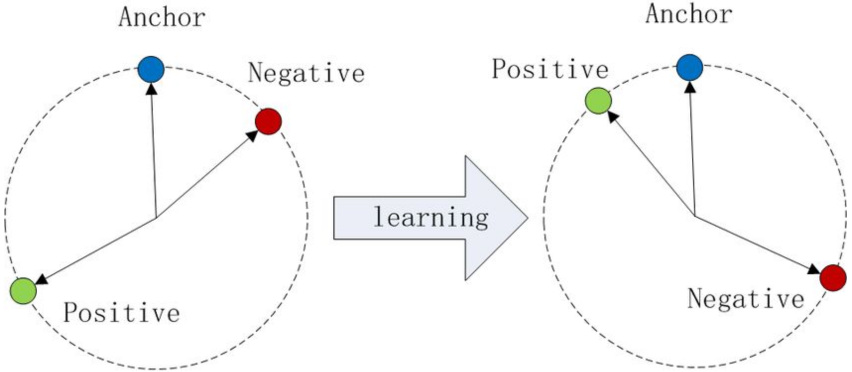
Algorithm 4 Training Data Generation

Require: Users, U = $\{U_1, \ldots, U_M\}$, Applications, A *// from database*
Require: $S_s = 0.4$, $S_d = 0.6$, $th_s = 0.9$, $th_d = 0.1$
Ensure: Preprocess Users, Applications

From Applications, filter, only those users that are **selected for at least one job**, $U_{selected} = \{U_1, ..., U_m\}$
Group the users based on the jobs they applied, $G = \{J_1 : \{U_{11}, U_{12} \ldots\}, \ldots, J_n : \{U_{n1}, U_{n2} \ldots\}\}$
for Job, J_i in G **do**
  Get all the users that applied to $J_i : \{U_{i1}, U_{i2}, \ldots\}$
  Create a 2-user combination of all users for J_i, $C = [\ldots, (U_{ix}, U_{iy}), \ldots]$
  for (U_{ix}, U_{iy}) in C **do**
    set anchor as U_{ix}
    Calculate similarity of skills and disability between anchor, U_{iy}
    if $S_s *$ similarity(Skills) $+ S_d *$ similarity(disability) $\geq th_s$ **then**
      set positive as U_{iy}
      for other users, $U_k : U_k \neq$ (anchor or positive) **do**
        Calculate similarity of skills, disability between anchor, U_{ix} & U_k
        Calculate similarity of skills, disability between positive, U_{iy} & U_k
        Calculate the average similarity of these two similarity scores
        if average similarity $\leq th_d$ **then**
          set negative as U_k
          Add triplets [**anchor** (U_{ix}), **positive** (U_{iy}), **negative** (U_k)] in training data
        end if
      end for
    end if
  end for
end for



Triplet Margin Loss Formulation



$$\text{Triplet Margin Loss} = \sum_{i=1}^{N} \max\left(0, \|\text{anchor}_i - \text{positive}_i\|_p - \|\text{anchor}_i - \text{negative}_i\|_p + \text{margin}\right)$$

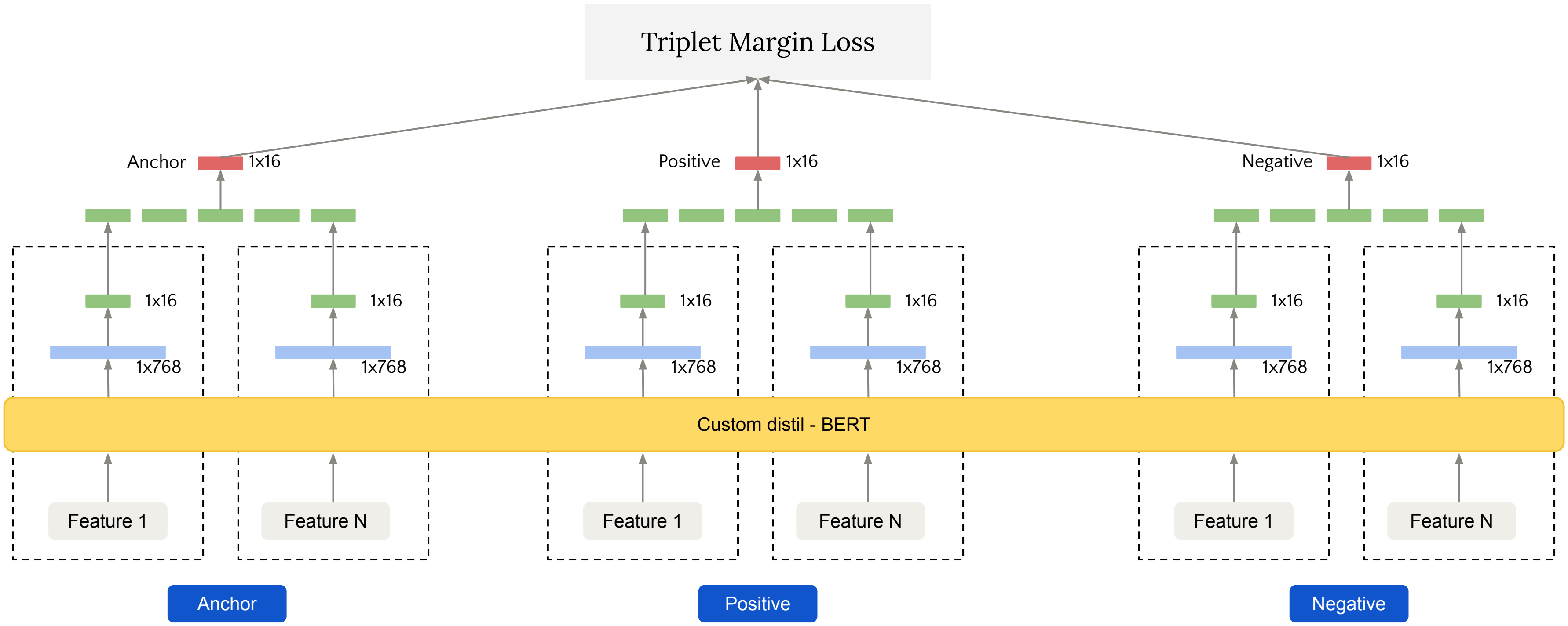
The distance between **(anchor and negative)** >= **distance (anchor, positive) + Margin**

Loss = 0 if Distance(anchor, negative) >= [Distance(anchor, positive) + Margin]
= x otherwise

Idea is to bring similar users close to each other, and take dissimilar users as far as possible

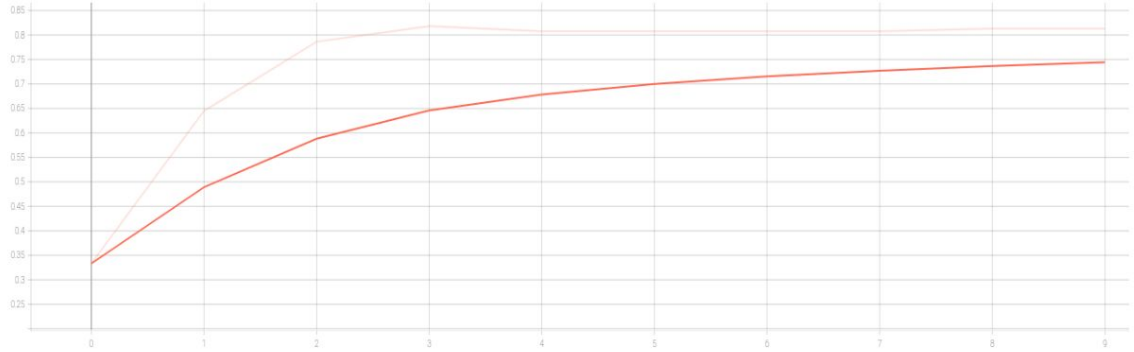
Learning local feature descriptors with triplets and shallow convolutional neural networks

Fine Tuning Custom-distilBERT with Triplet Margin Loss



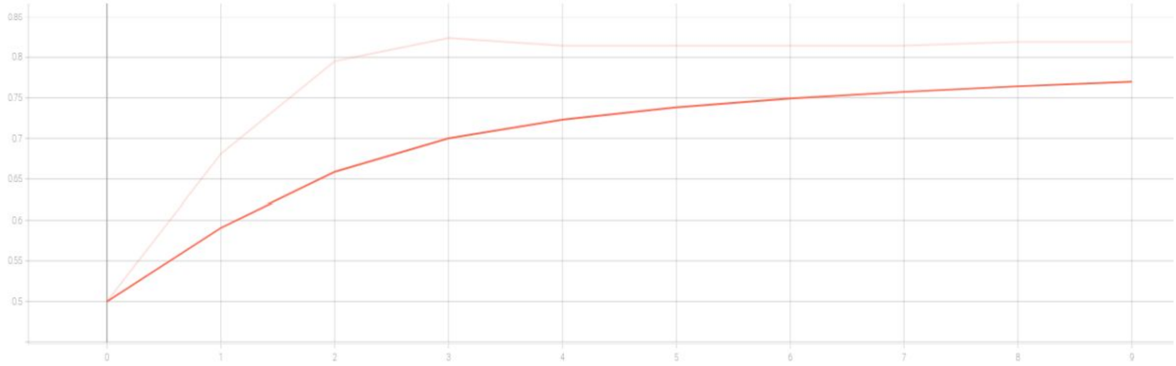
Validation Accuracy and F1 Score

Best F1 Score = **0.9389**



(c) Validation F_1 Score (Epochs Vs. F_1 Score)

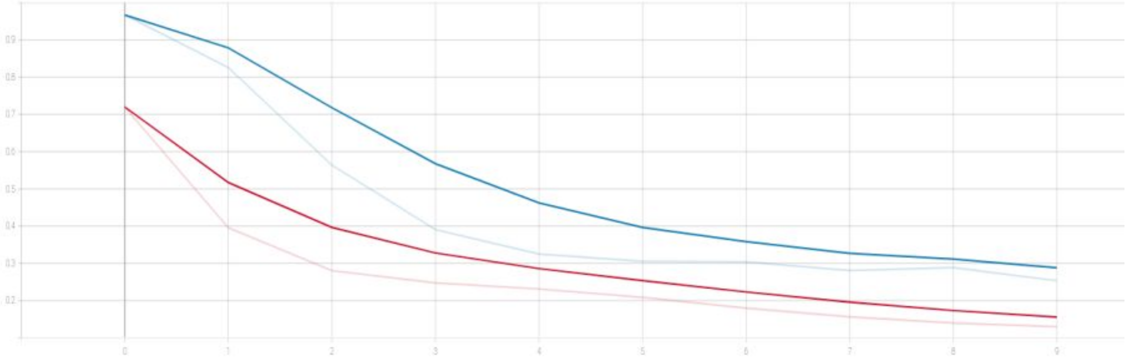
Best Accuracy = **0.9391**



(b) Validation Accuracy (Epochs Vs. Accuracy)

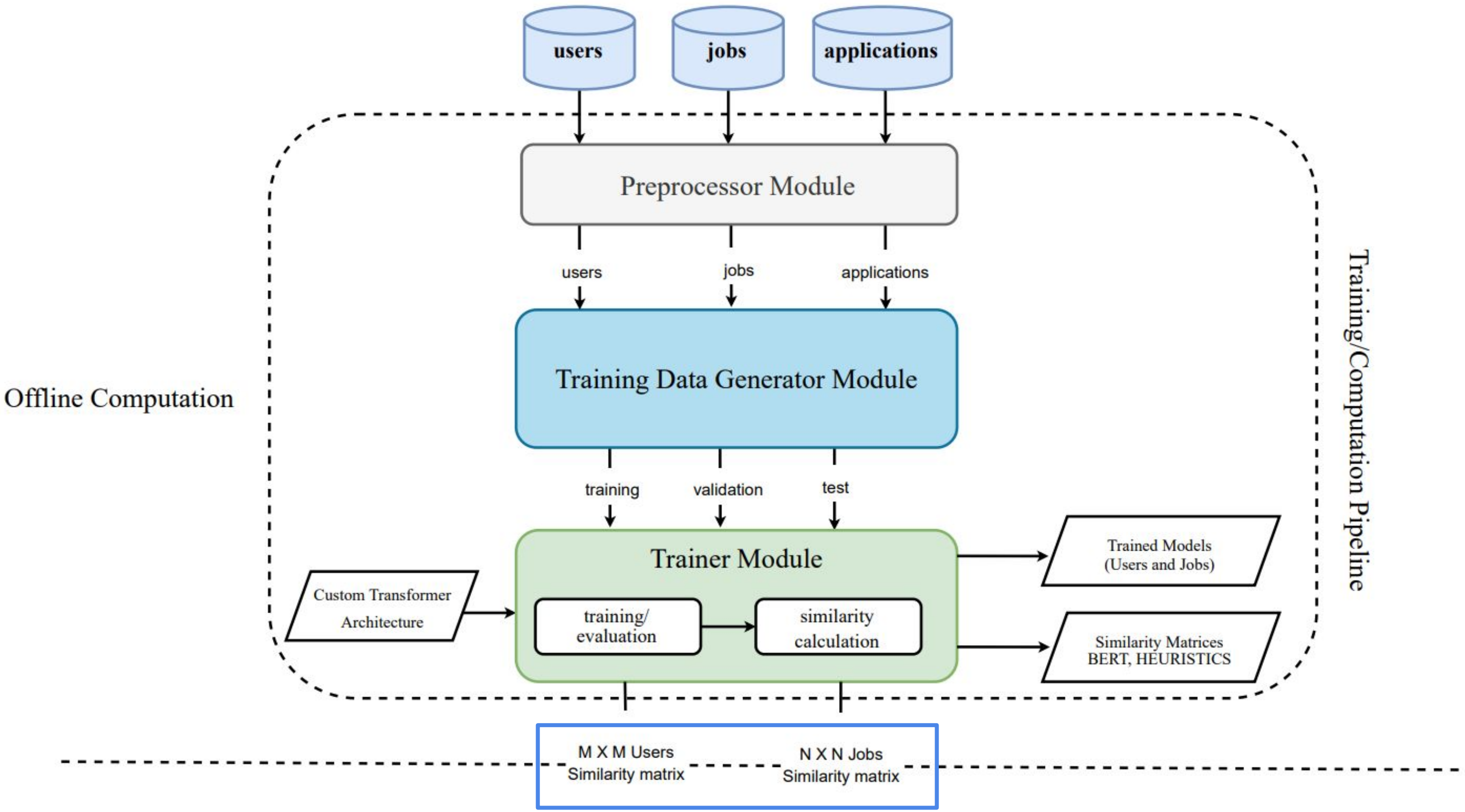
Triplet Loss Curves

(a) User Training Loss (Iterations Vs. Loss)

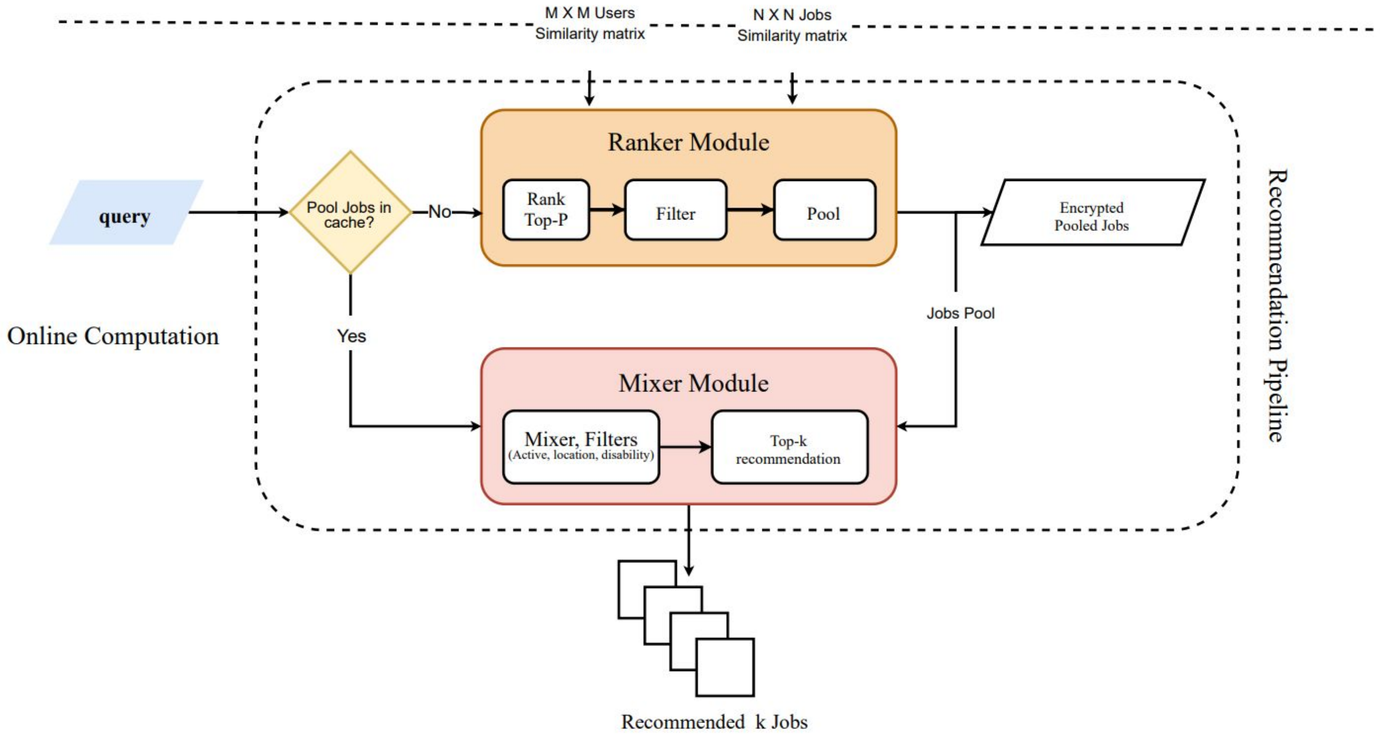


(d) Training and Testing Loss across Epochs (Blue = Training Loss, Red = Testing Loss)

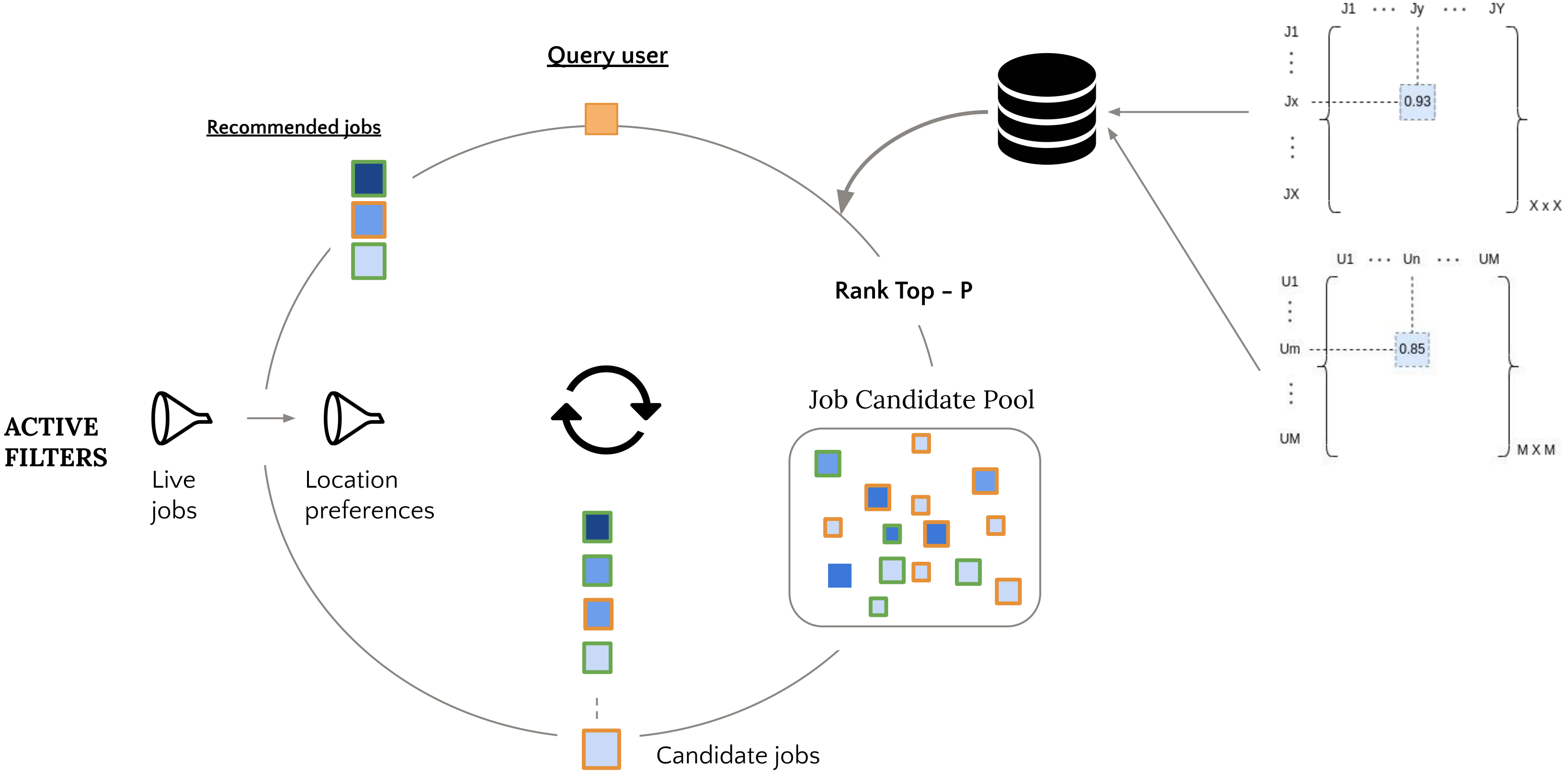
Calculate Similarity Matrices using Trained Transformer Model



Mixing and Recommendation Pipeline - ONLINE



Recommendation Pipeline - ONLINE

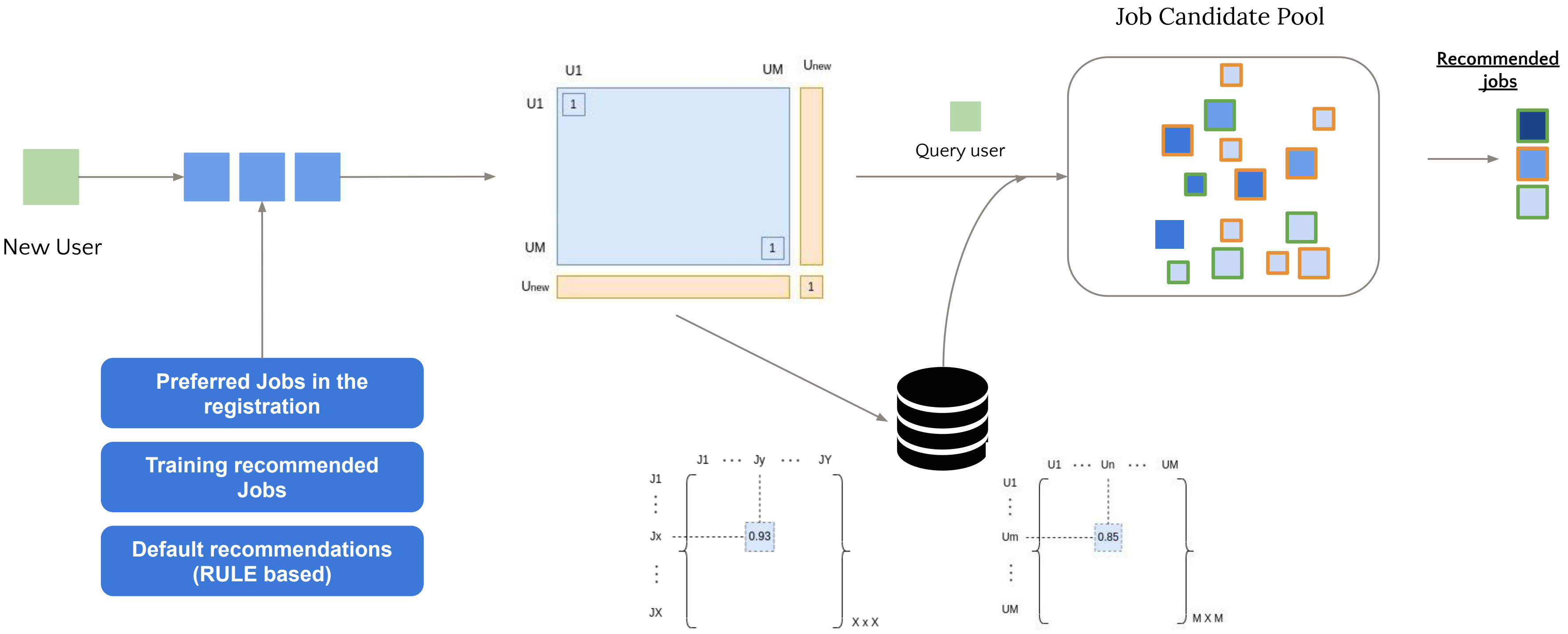


Execution Time - Cold Start and Warm Start

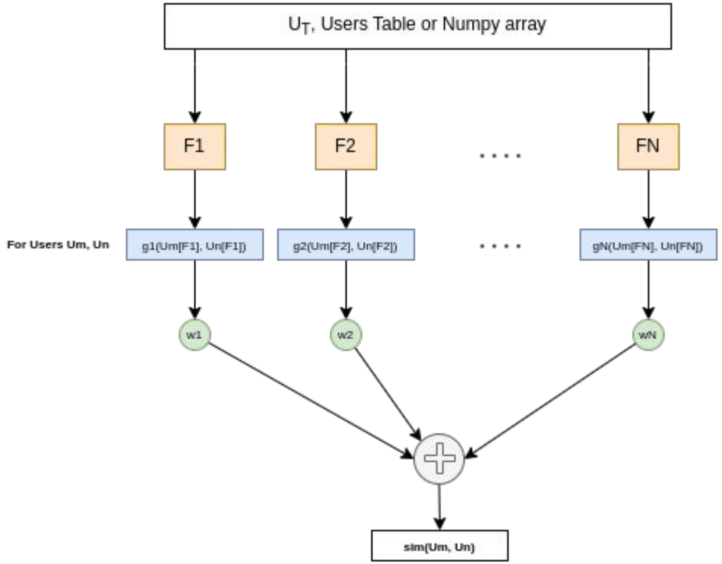
(a) Cold Execution of Recommendation Engine (No Job Pools Cache available)

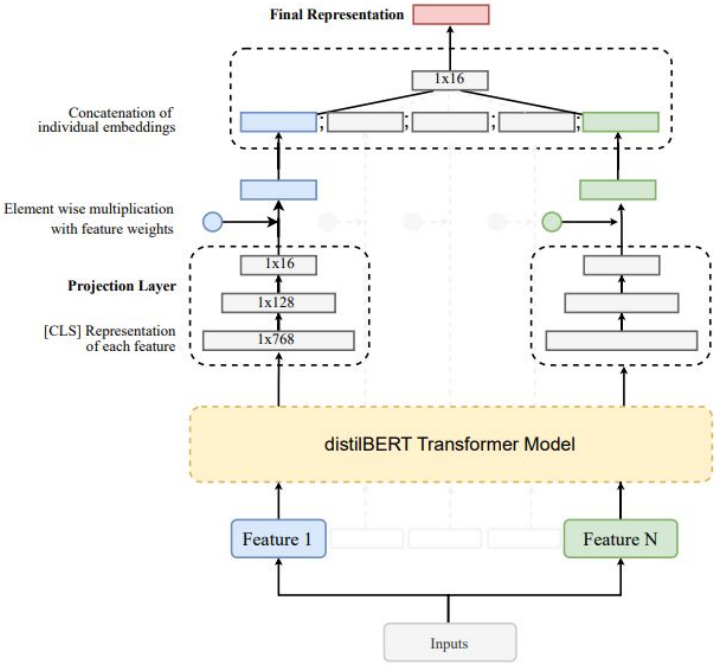
(b) Warm Execution of Recommendation Engine (Jobs pool available)

Incremental Updates on Similarity Matrices



Combination of both approaches yields best results





HEURISTICS

TRANSFORMER

Similar User Analysis - Both approaches

Type of User	user id	Disability Type	Skills	Talent	Score
QUERY	30292	LD(OH/ LD) - Upper Limb, Chronic Neurological Conditions	C++, Java, Asp.net, Ms Word, MS Excel, Data Entry, Computer Basics	Cooking	-
BERT	296	LD (OH/LD) - Trunk (Spinal Cord)	Computer Basics, Customer Service, Ms Word, Ms Excel, Java, Email And Chat, Data Entry, Organization	Drawing	0.97
	344	LD(OH/LD) - Upper Limb	Data Entry, Ms Word, MS Excel, Computer Basics	nan	0.959
	234	LD (OH/LD) - Upper Limb, LD(OH/LD) - Lower Limb	Ms Excel, Ms Word, Powerpoint Presentation, Social Media, Computer Basics	Video Creation	0.944
	56756	LD(OH/LD) - Lower Limb	Data Entry, Ms Excel, Computer Basics, Basic Math, Android, Social Media	Singing	0.943
	2343	LD(OH/LD) - Lower Limb	Computer Basics, Ms Word, MS Excel, Email And Chat, Data Entry	Art	0.937
HERUSITICS	5676	LD(OH/LD) - Upper Limb	Ms Word, MS Excel, Data Entry, Powerpoint Presentation	nan	0.571
	43567	LD(OH/LD) - Upper Limb	Basic Math, Android, Social Media	Cooking	0.548
	345	LD(OH/LD) - Upper Limb	Computer Basics, Data Entry, Email And Chat, Ms Excel, Ms Word, Powerpoint Presentation, Social Media, Technical Skills	nan	0.546
	7689	LD(OH/LD) - Upper Limb LD(OH/LD) - Lower Limb	nan	nan	0.531
	34534	LD(OH/LD) - Upper Limb	nan	nan	0.524

Job Recommendations for a query user

user id	Disability	Skills	Talents	Preferred Location	Experience
291001	LD (OH/ LD) - Lower Limb	Data Entry, Ms Excel, Ms Word, Powerpoint Presentation	nan	Visakhapatnam	3, 7

Job ID	Job Title	Type	Disability	Location	last Date
6273c47	Cognizant-Digital Marketing	Full Time	LD (OH/ LD) - Upper Limb, LD (OH/ LD) - Lower Limb, LD (OH/ LD) - Trunk (Spinal Cord), Dwarfism, Acid Attack Victim	Visakhapatnam	2023-06-28
6273c2f	Practitioner - Finance and Administration Delivery	Full Time	LD(OH/ LD) - Upper Limb, LD (OH/ LD) - Lower Limb	Delhi Visakhapatnam, Noida	2023-06-26
6273c4c	Top E-Commerce MNC-Billing Analyst	Full Time	LD (OH/ LD) - Upper Limb, LD (OH/ LD) - Lower Limb, Deaf and Hearing Impairment(HI), Low-Vision (LV)	nan	2023-06-17
6273c4a	Customer Support Officer	Full Time	LD (OH/LD) - Upper Limb, LD (OH/ LD) - Lower Limb, LD (OH/ LD) - Trunk (Spinal Cord), Dwarfism	Hyderabad, Bangalore	2023-06- 20
673b2a0	Sales Executive	Full Time	Physical Disability	Mumbai	2023-06- 21
6273c49	Top E-Commerce MNC-Senior Financial Operations Analyst	Full Time	LD (OH/ LD) - Upper Limb, LD (OH/ LD) - Lower Limb, LD (OH/LD) - Trunk (Spinal Cord), Dwarfism, Acid Attack Victim	Hyderabad	2023-06- 19
6273c4e	Practitioner/Senior - Finance and Administration	Full Time	LD (OH/ LD) - Lower Limb, LD (OH/LD) - Upper Limb	Noida	2023-06- 21
6273c31	Backend Sales	Full Time	LD (OH/ LD) - Lower Limb, Dwarfism, Acid Attack Victim	New Delhi	2023-06- 27

Overview

Conclusion

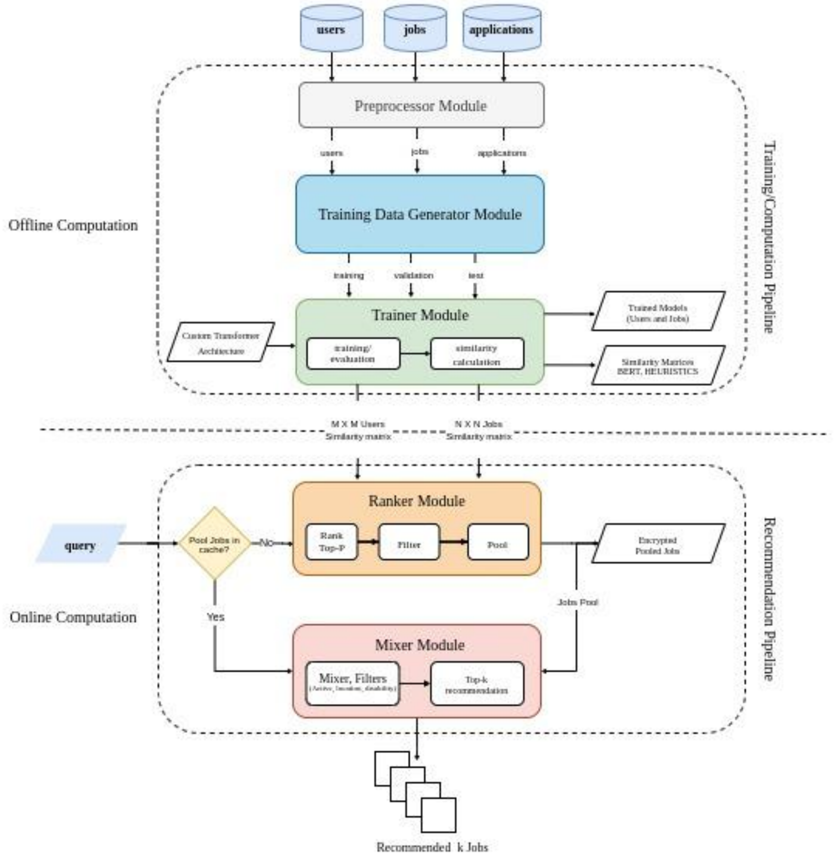
భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Hybrid Recommendation system with precomputed similarity matrices

An initial and innovative approach to address the challenges of personalized job recommendations by considering users' preferences, skills, and disabilities.

Automatic Triggers and Incremental Updates for adaptive and response to changes

Seamless integration with existing jobs platform with minimal space requirements



Future Directions

Not just structured data, Work with other user activity data like CTRs, Session Times

Not just job recommendations, extend to recommendation of training programs, certification, networking events

Skills Hierarchy to organize skills and better capture similarity

Better handle missing data - Required during Registrations

References

1. R. Kessler, N. Béchet, M. Roche, J.-M. Torres-Moreno, and M. El-Bèze. A hybrid approach to managing job offers and candidates. Information processing & management.

2. J. C. Valverde-Rebaza, R. Puma, P. Bustios, and N. C. Silva. Job Recommendation Based on Job Seeker Skills: An Empirical Study.

3. A. Gugnani and H. Misra. Implicit skills extraction using document embedding and its use in job recommendation. In Proceedings of the AAAI conference on artificial intelligence.

4. S. Ahmed, M. Hasan, M. N. Hoq, and M. A. Adnan. User interaction analysis to recommend suitable jobs in career-oriented social networking sites. In 2016 International Conference on Data and Software Engineering (ICoDSE).

5. Y. Zhang, C. Yang, and Z. Niu. A Research of Job Recommendation System Based on Collaborative Filtering. In 2014 Seventh International Symposium on Computational Intelligence and Design.

6. W. Lin, L. Shou, M. Gong, P. Jian, Z. Wang, B. Byrne, and D. Jiang. Transformer-Empowered Content-Aware Collaborative Filtering 2022.

7. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web.

Thanks!

Any questions ?

You can find me at:

- cs21mtech16001@iith.ac.in

Academic and Professional Updates at:

- shresthakamal.com.np