Generating User Embedding from Social Data Links

Md Shahbaz Siddiqi

Data Set

From the received data, we finalised that user embeddings should be generated considering following user aspect:

- Tweets/ Retweets posted by the user
- User personal about information
- A connection network that consist of individuals users tweeted about, mentioned or replied to.

The Data mainly consist tweets from various Health Care Practitioners along with people associated with them or users who tweet about topics in closely associated with the domain.

The Data we were working was closely related to Cardiology domain, specifically from users who mostly belong to North America region. Tweets so collected was from period 04-04-2020 to 23-04-2020.

User ID	User Meta Data	Tweets	AllConnection
0moba	['EM Physician | @Dartmouth MD MBA | Errand boy @digitalhealthNg & @webelievehealth | Passionate about Nigerian Healthcare systems | Iâ€™ll do it for Dodo']	['@docneto Ha my brother donâ€™t say this one. Patients are clotting through full prophylactic dosing of lovenox, then clotting through half dose heparin gtt, then having massive life ending spontaneous bleeds on full dose heparin gtt. No silver bullet here.', '@victorr_ugo exactly! In the first week of this pandemic here, a PCP lost 5 patients from non-covid conditions because they couldnâ€™t follow up / get the help they needed because all hands on deck. Those are even the people we should think about. The HF with no lasix, DM with no insulin']	['docneto', 'victorr_ugo', '0moba']
1600ronnell	['RN4LðŸ–¤']	['RT @fox5dc Coronavirus now killing more Americans than cancer, heart disease: report https://t.co/WouBrvHap8']	['1600ronnell', 'fox5dc']
1963Kelli	["Geek Mom/Feminist/Hiker surrounded by fellow working class people who've been convinced they're Republicans..help me! Floral/Event designer by trade, no DMs!"]	['RT @koolkaryn â€˜Imagine Having Ability to Do This All Along, and Choosing Not toâ€™: Eli Lily Cuts Cost of Insulin Amid Covid-19 Outbreak https://t.co/2SSYdwM3pT via @dailysoundnfury']	['dailysoundnfury', '1963Kelli', 'koolkaryn']
1EyeSmart1	['Male. Physician. Independent Thinker. Pro USA. Anti trump. #TexasDoctor. Debt & Taxes make the rich richer. Debt & Taxes make the poor &	["RT @BusyBrain_Very COVID-19 has now surpassed Heart Disease and Cancer as the number one cause of death in America per day And is now reported to be even more contagious than originally thought. Stay Inside. Stay Healthy. Protect Others. That's your part #COVID-19 #OneVoice1"]	['BusyBrain_Very', '1EyeSmart1']
1HopelesslyHope	['Makeup and Beauty Fanatic, Lover of Pretty Food, Wife, Mom, Pediatric Nurse, Superhero']	['RT @CNN Pharmaceutical company Novo Nordisk said it will offer insulin free of charge for 90 days to diabetes patients who lost health insurance coverage because they lost their jobs https://t.co/KsvrQtjFjN']	['1HopelesslyHope', 'CNN']

Approaches Explored

There is mainly two approaches used to generate embeddings **Single View and Multi View approach**.
In order to generate Single View Embedding, we used **User Doc and Post Doc** method for Text based user data and **Linear and non-linear algorithm for Network based user data**.

Post Doc Basic: was used to generate embeddings by utilising user text data in form of tweets and meta data by using document to vector model.
Post Doc modified: was used to generate embedding using pre trained models like BERT, Clinical BERT, Sci BERT and Bio BERT.

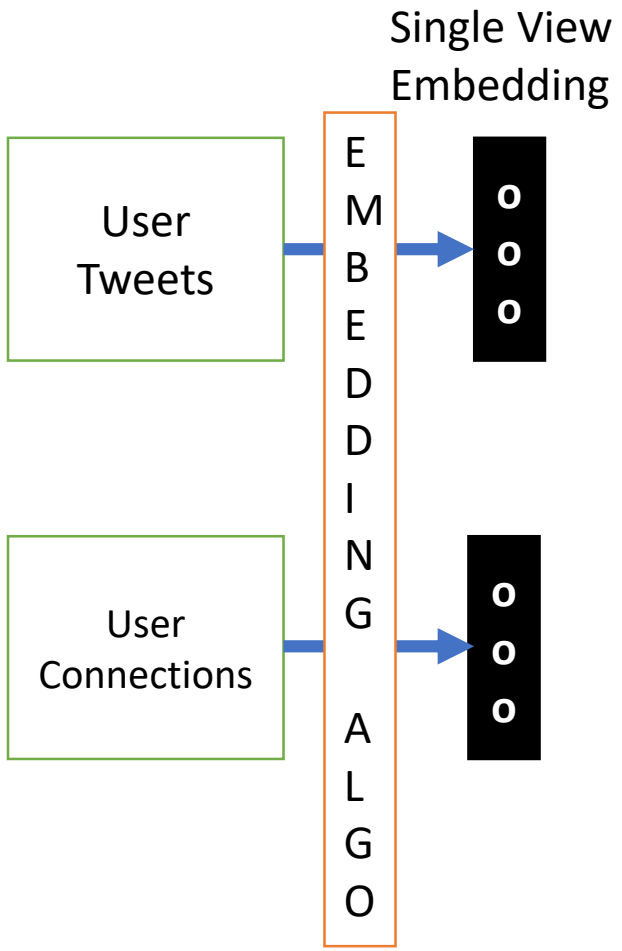
User Doc Basic: was used to generate embeddings from user meta data as well as user tweets, by considering only one tweet at a time. This further used the traditional paragraph to vector algorithm suggest by Thomas Mikolov in 2014.
User Doc modified: similar to the basic version in order to generate embedding pre trained models like Sci BERT, Clinical BERT and Bio BERT was used.

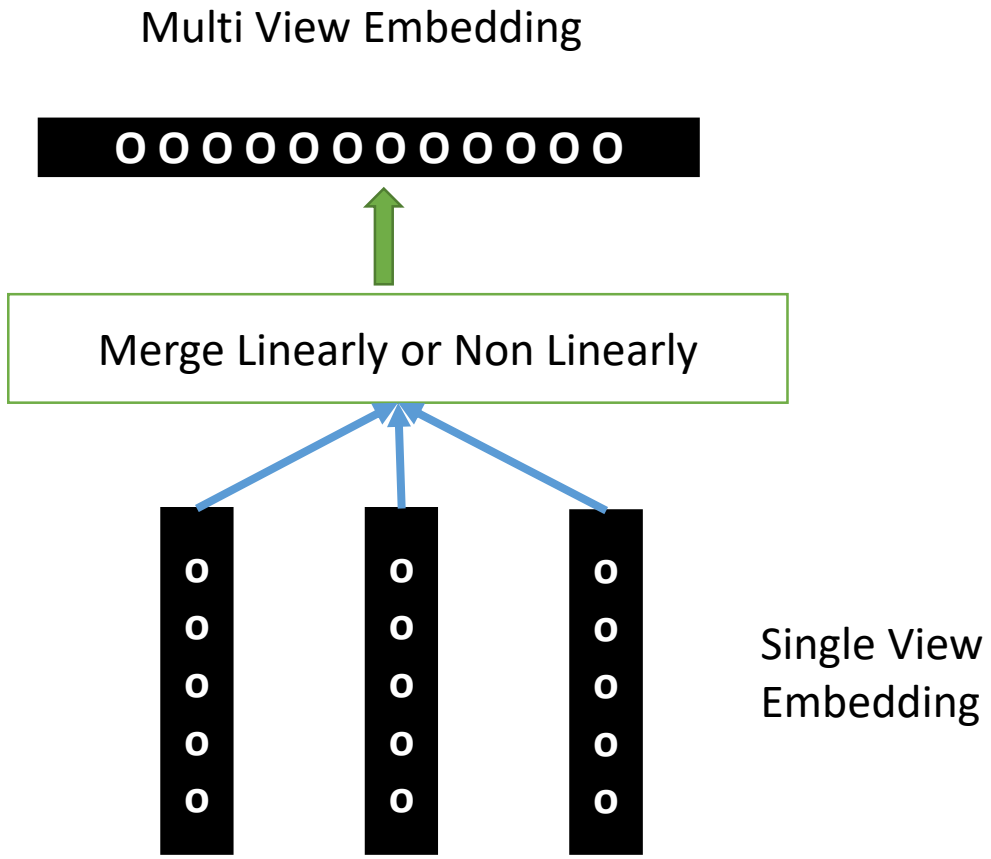
Network based user representation: Single View representation for user was generated, accounting every individual as a node of large graph and trying to create similar embedding for nodes who are connected. This was performed using linear method and non-linear methods like state of the art Deep Walk and Node2vec approach.

Multi View approach: was used to generate embedding by merging two or more single view embeddings.

Single View and Multi View Approach

Single View approach: User Embedding generated considering only single aspect of user data.
Multi View approach: User Embedding generated merging multiple single view embedding, either linearly or non-linearly

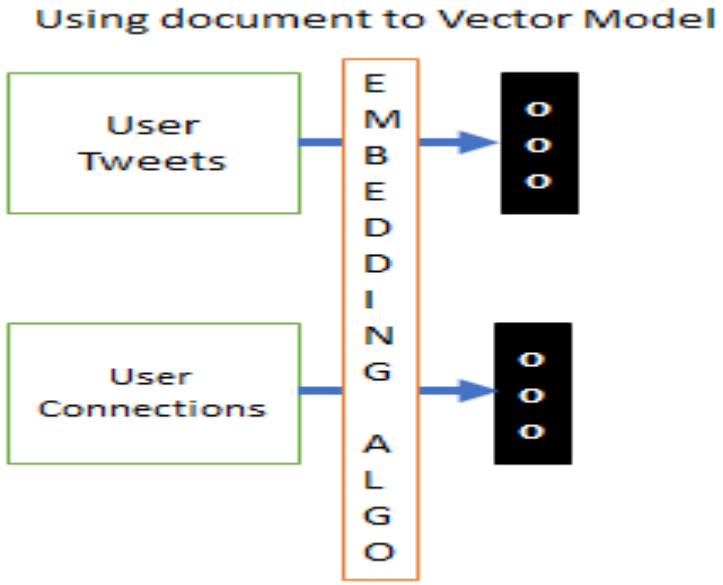


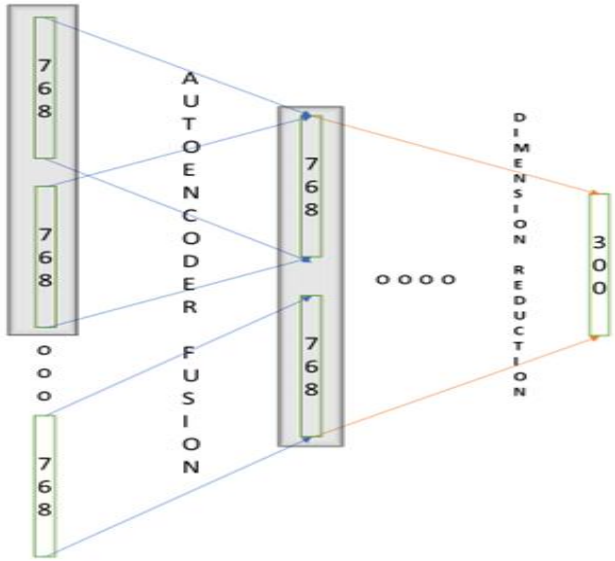


Single View Based Embedding

User Doc and Post Doc Method

POST DOC	USER DOC
• Post Doc method treats entire user data as a single corpus, which a document to vector model excepts as an input and generate a vector of pre-defined dimension. • Same algorithm is used to treat any type of text based user data.	• User doc method treats user tweet data separately by passing individual treat one at a time to paragraph to vector model, the vector so obtained is merged with other vectors of tweets to generate embedding for user tweet data. • User doc algorithm treat user meta data in a similar way as its treated by post Doc method, how ever user tweet data are treated in a different manner.





Experiment and Findings

While experimenting with some public data and then evaluating the results on post doc and user doc method, we confirmed following results.

	Adriande	Drake	Bruno Mars	Britney	Harry	Akshay	SrkKhn	Salman	SrBachn	Instagram	Twitter	Youtube	BBC	CNN	ESPN	NY Times	Sports Centre	Bill gates	Cr7	FC Barca	Real Mad	Virat	Obama	Trump	Nasa
Adriande	1	0.8192	0.8085	0.86	0.84	0.8223	0.857	0.7954	0.7754	0.8629	0.8299	0.8661	0.752	0.822	0.83	0.845	0.828	0.844	0.8	0.834	0.674	0.84	0.8406	0.837	0.87
Drake		0.9349	0.824	0.8576	0.81	0.8104	0.841	0.8475	0.7474	0.8464	0.8305	0.8554	0.752	0.848	8254	0.847	0.817	0.841	0.803	0.8002	0.688	0.82	0.8408	0.831	0.854
Bruno Mars			1	0.8584	0.83	0.8115	0.847	0.8466	0.7714	0.8392	0.8174	0.8571	0.759	0.836	0.822	0.84	0.814	0.851	0.8377	0.7854	0.684	0.83	0.8239	0.841	0.851
Britney				0.9314	0.89	0.8575	0.863	0.8533	0.7936	0.8881	0.8493	0.8933	0.78	0.876	0.84	0.872	0.836	0.875	0.8505	0.8622	0.702	0.86	0.8686	0.87	0.896
HarryStyl					1	0.8612	0.853	0.8305	0.804	0.8545	0.8481	0.8709	0.757	0.837	0.838	0.874	0.826	0.84	0.81	0.8176	0.678	0.84	0.8607	0.843	0.871
Akshay						0.9284	0.827	0.809	0.788	0.833	0.815	0.8541	0.747	0.825	0.823	0.861	0.818	0.829	0.8061	0.8215	0.677	0.83	0.8373	0.832	0.853
SrkKhn							0.98	0.8468	0.8095	0.8575	0.8516	0.9006	0.803	0.848	0.818	0.86	0.841	0.89	0.88	0.8233	0.697	0.89	0.8749	0.863	0.902
Salman								1	0.747	0.859	0.8411	0.8576	0.751	0.858	0.838	0.85	0.82	0.857	0.8205	0.7987	0.693	0.84	0.8535	0.846	0.863
SrBachn									0.8948	0.7924	0.7733	0.8131	0.731	0.766	0.792	0.792	0.772	0.816	0.796	0.7607	0.646	0.8	0.7882	0.789	0.82
Instagram										1	0.8414	0.879	0.796	0.862	0.824	0.867	0.815	0.863	0.838	0.8685	0.707	0.86	0.8735	0.86	0.899
Twitter											1	0.8684	0.74	0.844	0.82	0.843	0.821	0.855	0.802	0.805	0.68	0.84	0.8548	0.83	0.861
Youtube												0.9188	0.799	0.874	0.839	0.882	0.807	0.894	0.8772	0.848	0.72	0.88	0.88	0.807	0.9
BBC													0.902	0.75	0.77	0.784	0.76	0.78	0.81	0.75	0.64	0.77	0.78	0.77	0.811
CNN														0.899	0.82	0.85	0.83	0.861	0.82	0.82	0.701	0.83	0.8541	0.84	0.876
ESPN															0.842	0.84	0.82	0.857	0.83	0.814	0.694	0.84	0.84	0.833	0.86
NYTimes																0.964	0.831	0.85	0.831	0.831	0.69	0.85	0.87	0.86	0.882
SportsCentre																	0.844	0.852	0.8299	0.82	0.693	0.85	0.84	0.83	0.87
Billgates																		0.915	0.872	0.83	0.701	0.88	0.86	0.86	0.899
Cr7																			1	0.801	0.69	0.86	0.84	0.846	0.87
FcBarca																				1	0.68	0.85	0.8512	0.83	0.871
RealMad																					0.704	0.7	0.703	0.705	0.72
Virat																						1	0.86	0.859	0.89
Obama																							0.91	0.852	0.895
Trump																								0.862	0.822
Nasa																									0.927

User Doc based result

	Drake	Akshay	SrkKhn	SrBachn	Youtube	BBC	CNN	ESPN	NY Times	Sports Centre	Bill gates	Obama	Trump	Nasa
Drake	0.7589	0.5313	0.501	0.4665	0.4692	0.4993	0.4729	0.5265	0.4496	0.5107	0.5136	0.5268	0.466	0.45
Akshay		0.7797	0.602	0.52	0.54	0.5952	0.6029	0.612	0.53	0.58	0.5439	0.5949	0.5283	0.544
SrkKhn			0.8099	0.521	0.5511	0.5743	0.5557	0.602	0.4888	0.5566	0.506	0.57	0.5	0.546
SrBachn				0.753	0.422	0.5454	0.5291	0.5297	0.45	0.5003	0.4436	0.4923	0.45	0.472
Youtube					0.6907	0.5233	0.5532	0.5572	0.54	0.5223	0.5445	0.566	0.5287	0.559
BBC						0.8113	0.6741	0.6326	0.5876	0.6122	0.518	0.5874	0.5556	0.593
CNN							0.8498	0.6576	0.63	0.6376	0.5629	0.6183	0.599	0.596
ESPN								0.6588	0.562	0.6325	0.553	0.6245	0.565	0.579
NYTimes									0.8142	0.635	0.55	0.624	0.5636	0.573
SportsCentre										0.6447	0.522	0.5962	0.5602	0.554
Billgates											0.7483	0.623	0.5432	0.536
Obama												0.7929	0.574	0.553
Trump													0.6185	0.541
Nasa														0.646

Post Doc based result

Although it was observed that user doc method performed better than the Post Doc method, both had following disadvantage. They suffered from Out of Vocabulary word which pre-trained model from genism doc2vec or glove could not provide.

Modified Algorithm

In order to overcome our existing algorithm, draw back we used recently achieved State of the Art BERT model. How-ever we are experimenting with following BERT model considering our business requirement.

- **BIO-BERT**: which is trained on PubMed journals.
- **SCI-BERT**: which is trained on 1.14 Million papers from semanticscholoars.org
- **CLINICAL-BERT**: which is trained on Clinical corpus.

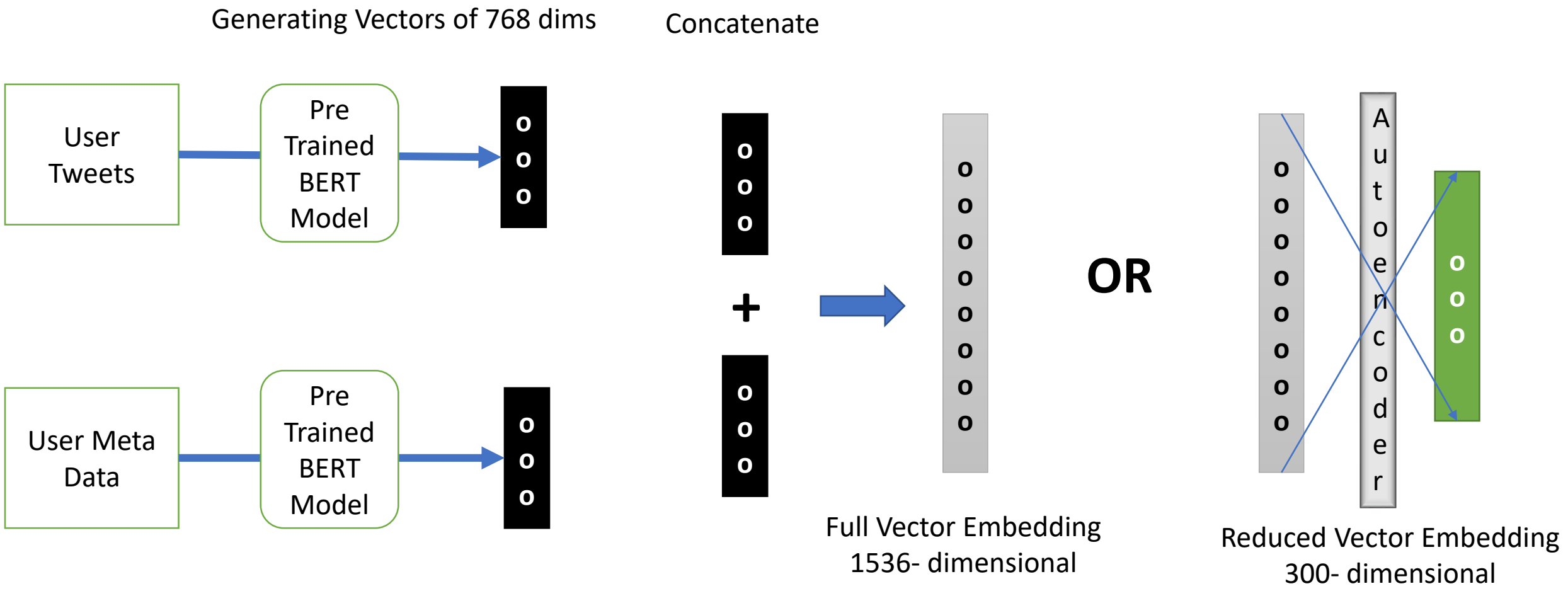
For both Post Doc method as well as User Doc method, our pre trained BERT model generated a vector of 768 dimensions.

We also experimented with a reduced version of these embedding obtained from passing through an Auto Encoder, in hope that it may learn certain hidden representation between the vectors and improved the quality of the embeddings.

We will discuss detailed findings of the experiments later in the presentation.

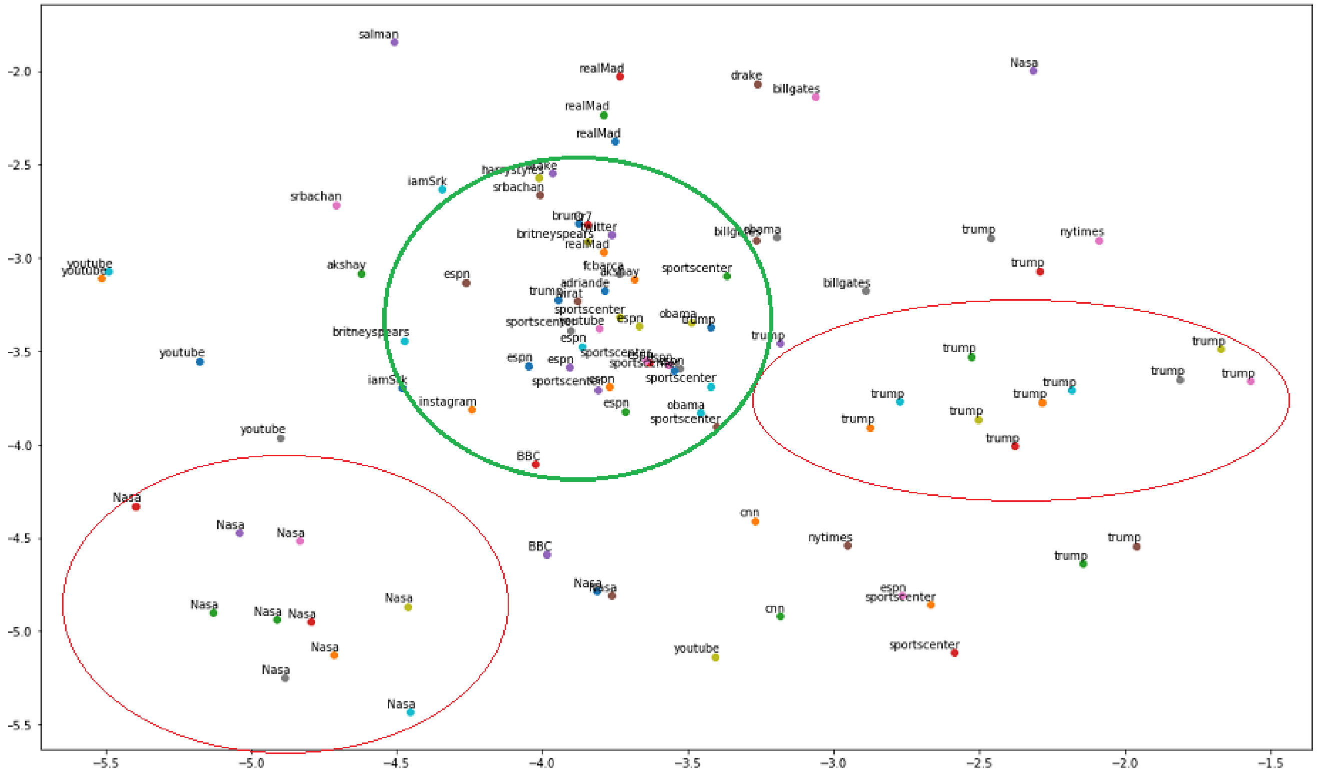
Generation of Embedding using pre trained Model

For Evaluation we are considering two types of Vectors of same user embedding. One is the actual version and other is the reduced version, in order to account any hidden information that can be obtained.



Evaluation Algorithm

Draw back of Existing methods

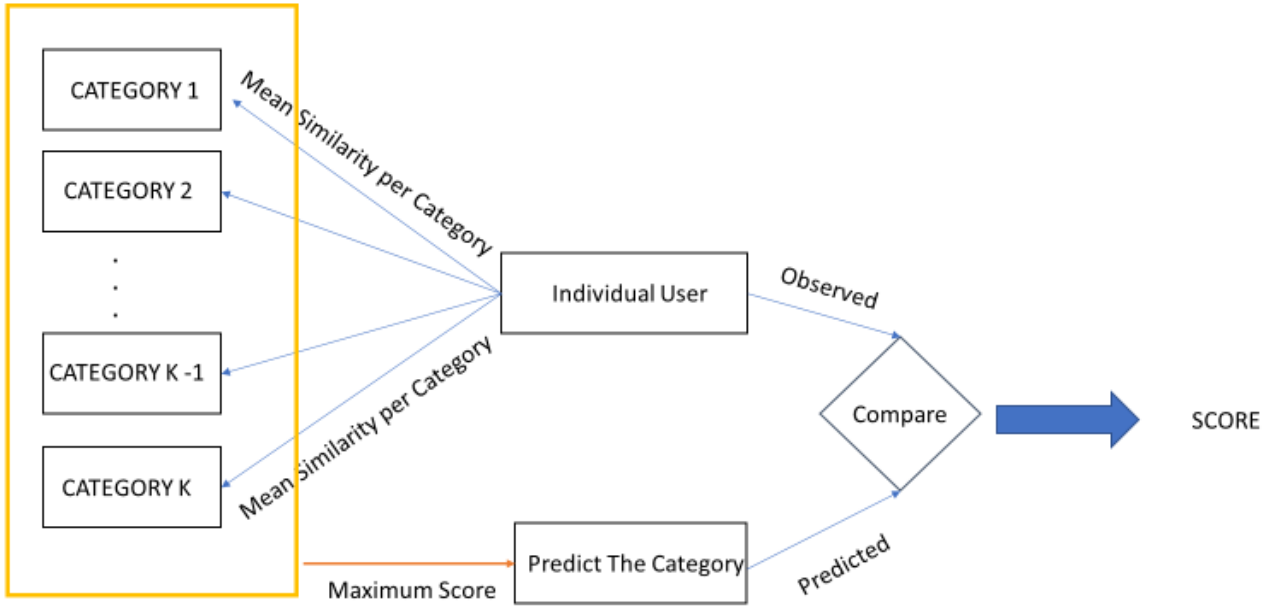


Our existing methods to evaluate the embedding depend highly on visually analyzing the scatter plots of reduced embeddings and then looking for cluster formation in the same.

A major drawback of this method is that these dimension reduction is performed using PCA techniques. However, there is information loss although is smaller amount when we linearly reduce the dimension, such as done in PCA. Also it depends highly on human evaluations.

Another method is to calculate similarity score for like category users or users which fall into similar category, while calculating the dis-similarity score from different users. A user embedding for whom similarity score as well as dis-similarity score are both high, can be assumed as a good user embedding.

Proposed Algorithm



In order to overcome existing drawbacks of evaluation method, I developed an algorithm that gives an accuracy score of these embedding based on a pre defined task.

We segregate some prelabelled user's embedding under defined category. Following which we test an individual user embedding on basis of its classification power to correct category. A diagram of the method is given below.

Evaluation Experiment

The Evaluations are done for each method for three types of task.

Evaluation for 8 categories:
The different categories are :
Nurse, Professor/Director, Physicians, Doctors/Surgeons/Specialist,
Veterans/Pharmacist, Organisations, Student, Others

Evaluation for 4 categories:
The users are next classified for a lower number of bins, mainly 4 categories. These were on basis of number of users under consideration
Nurse , Doctors and Professionals , Organisations , Others

Evaluation for 2 categories:
The users are next classified for a lower number of bins, mainly 4 categories. These were on basis of number of users under consideration
Health Care Practitioner, Others

Validation users per category was taken 10 each.

Results and Findings

Post Doc and User Doc using Doc2Vec

Below is the comparison of Post Doc and User Doc accuracy on classification task, using genism doc2vec model

	POST DOC			USER DOC		
Embedding Type	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**
User Embedding Full	0.1141	0.3011	0.3369	0.1232	0.3320	0.3620
User Embedding Reduced	0.1320	0.2963	0.3870	**0.1917**	**0.3552**	**0.4802**

Observation: As observed the User Doc method performed better than the Post Doc method on all 3 classification task.

Post Doc and User Doc using Clinical BERT

Below is the comparison of Post Doc and User Doc accuracy on classification task, using per-trained Clinical BERT model

	POST DOC			USER DOC		
Embedding Type	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**
User Embedding Full	**0.2968**	**0.4208**	**0.7670**	0.1004	0.4903	0.5734
User Embedding Reduced	0.2420	0.3978	0.7670	0.0821	0.4787	0.5734

Compared to existing embedding, when we used Clinical Bert we observed an improvement in performance.

Improvement Summary:

- Accuracy on 8 categories: **+10.51 %**
- Accuracy on 4 categories: **+6.56%**
- Accuracy on 2 categories: **+28.68%**

Post Doc and User Doc using Sci BERT

Below is the comparison of Post Doc and User Doc accuracy on classification task, using per-trained Sci BERT model

	POST DOC			USER DOC		
Embedding Type	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**
User Embedding Full	**0.3196**	0.2702	**0.7728**	0.2237	0.5019	0.5770
User Embedding Reduced	0.2831	0.4440	0.7778	0.2602	**0.5057**	0.5771

In comparison to Clinical BERT result Sci BERT was able to achieve a better accuracy, however since we are restricting our self to choosing only single type of embedding generation method, the improved summary are follows:

Improvement Summary:

- Accuracy on 8 categories: **+2.28 %**
- Accuracy on 4 categories: **Clinical BERT performed better**
- Accuracy on 2 categories: **+1.58%**

Post Doc and User Doc using Bio BERT

Below is the comparison of Post Doc and User Doc accuracy on classification task, using per-trained Bio BERT model

	POST DOC			USER DOC		
Embedding Type	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**	**Accuracy on 8 categories**	**Accuracy on 4 categories**	**Accuracy on 2 categories**
User Embedding Full	0.2831	0.5444	0.7670	0.1004	0.4942	0.5734
User Embedding Reduced	**0.3561**	**0.5637**	**0.7786**	0.0958	0.4942	0.5806

Significant improvement observed in comparison to previous results, when using Bio BERT as pre trained model.

Improvement Summary:

- Accuracy on 8 categories: **+3.65 %**
- Accuracy on 4 categories: **+14.29%**
- Accuracy on 2 categories: **Almost same as Sci BERT**

NETWORK BASED USER EMBEDDING

Generating Embedding using linear method

For a given user we consider its network comprising of users they have mentioned/ tweeted about or replied to.
I have applied the methodology from which word based embedding were generated.

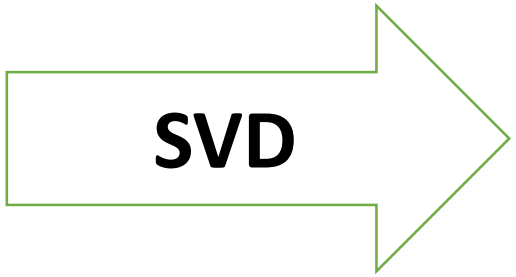
- Develop a co-occurrence matrix which is sparse in nature for **N** number users, co-occurrence matrix is **N*N dimension**.
- Calculate PPI score for each index, that is based on the formula :

PPI-score: log [count (user1, user2) * N / (count (user1) * count (user2))]

- We will need to reduce the dimension of the matrix using **Singular Value Decomposition** method, this will not only reduce sparsity but also develop any hidden or indirect linkage between users if any.

	User 1	User 2	User 3	User 4	Total
User 1	0	2	0	1	3
User 2	2	0	1	0	3
User 3	0	1	0	3	4
User 4	1	0	3	0	4

Co-occurrence Matrix



	User 1	User 2	User 3	User 4
User 1	0	0.66	0.05	0.33
User 2	0.66	0	0.24	0.12
User 3	0.05	0.24	0	0.78
User 4	0.33	0.12	0.78	0

Each row indicates representation for users

Accuracy for Network based user embedding

We evaluated the user representation generated by network based structure using linear method.

Embedding Type	Accuracy on 8 categories	Accuracy on 4 categories	Accuracy on 2 categories
Network based User representation	0.0821	0.0694	0.7778

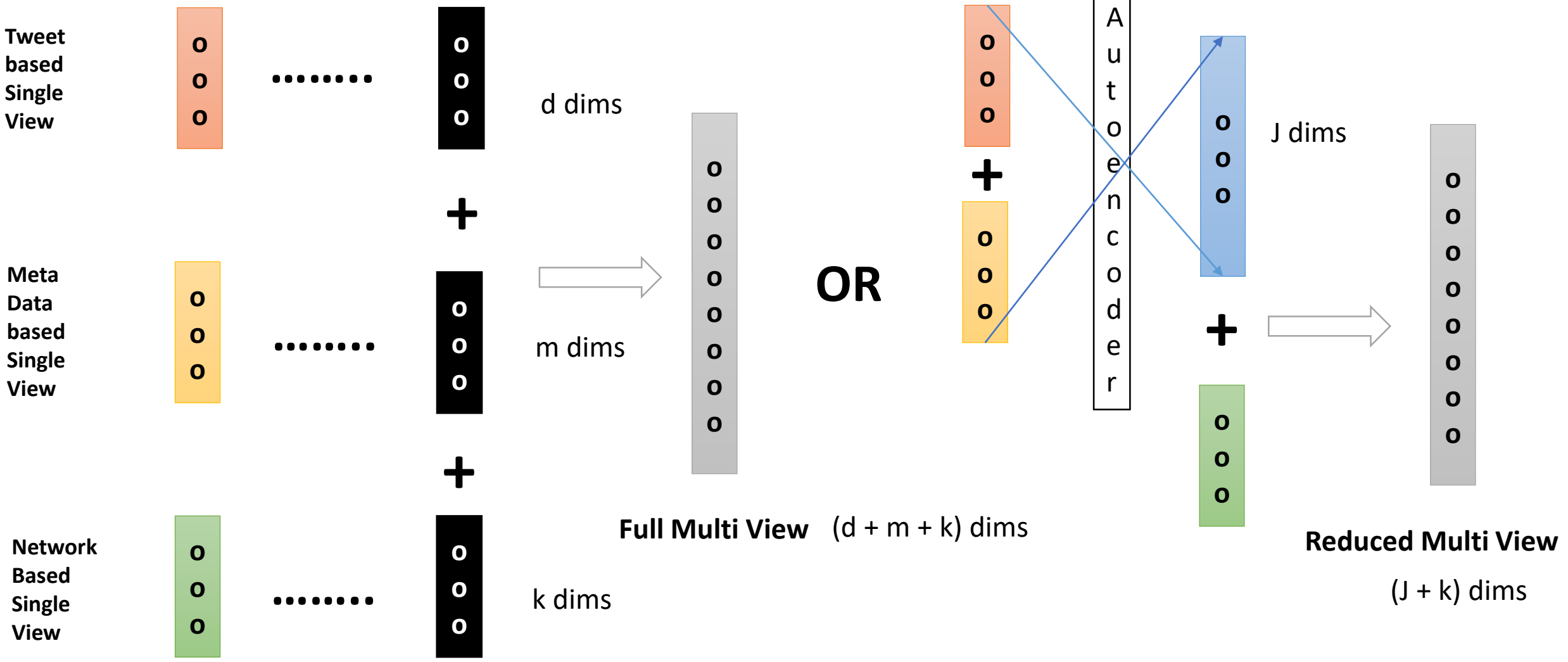
As can be observed the embedding performed poorly in 2 of the 3 task as compared to other single view approach.
Hence, a Multi View based algorithm was followed to generate a high quality result.

Multi View Approach

Generation of Multi View Approach

Concatenating Single Views

Auto Encoder merges text based and meta data based embedding to generate a combined embedding of J dimension



Multi View using Linear network representation

We evaluated the multi view representation using linear network only for Bio BERT and Sci BERT generated output, because of the significant better performance achieved earlier

Embedding Type	Accuracy on 8 categories	Accuracy on 4 categories	Accuracy on 2 categories
Multi View Embedding Full using Bio BERT	0.0639	0.0965	0.4802
Multi View Embedding Reduced using Bio BERT	0.0867	0.1081	0.7132
Multi View Embedding Full using Sci BERT	0.1278	0.1505	0.5304
Multi View Embedding Reduced using Sci BERT	0.1050	0.1042	0.5125

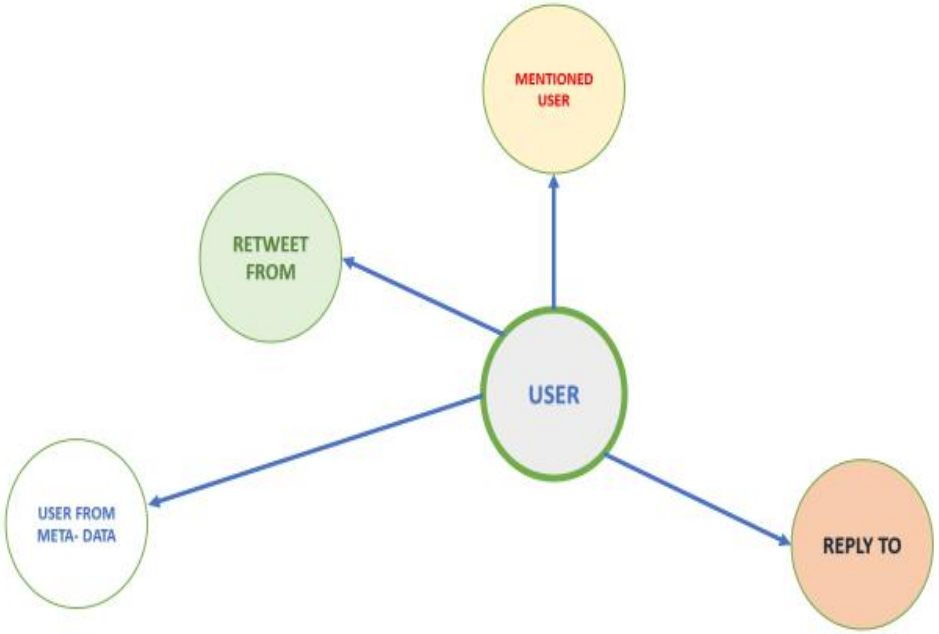
As observed the performance did not improved rather degraded in multi view approach. On analysis it was observed this was due to already poor performing network representation using linear method

Note: results are shown for vector dimension 1656 (full) and 300 (reduced) only as they were the best observed among many tried.

Generating Embedding using Non-Linear Method

In order to create embedding from network of user, I used **Graphical Neural Network** and converted the network into a graphical structure, there by applied **Deep Walk** algorithm explained below:

- Represent Network in form of Graph where each node represent a user and an edge between two user indicate existence of a connection.
- Perform random walk on these nodes and develop a corpora like structure where each walk can be considered as sentence like structure.
- Perform skip gram method on the corpus and develop embedding for each node.
- **This performed better result and was able to train with far less data.**



Findings and results from Multi View using Deep Walk

The Multi View was applied exactly in the same order as was performed for linear method, network representation. Below are the findings:

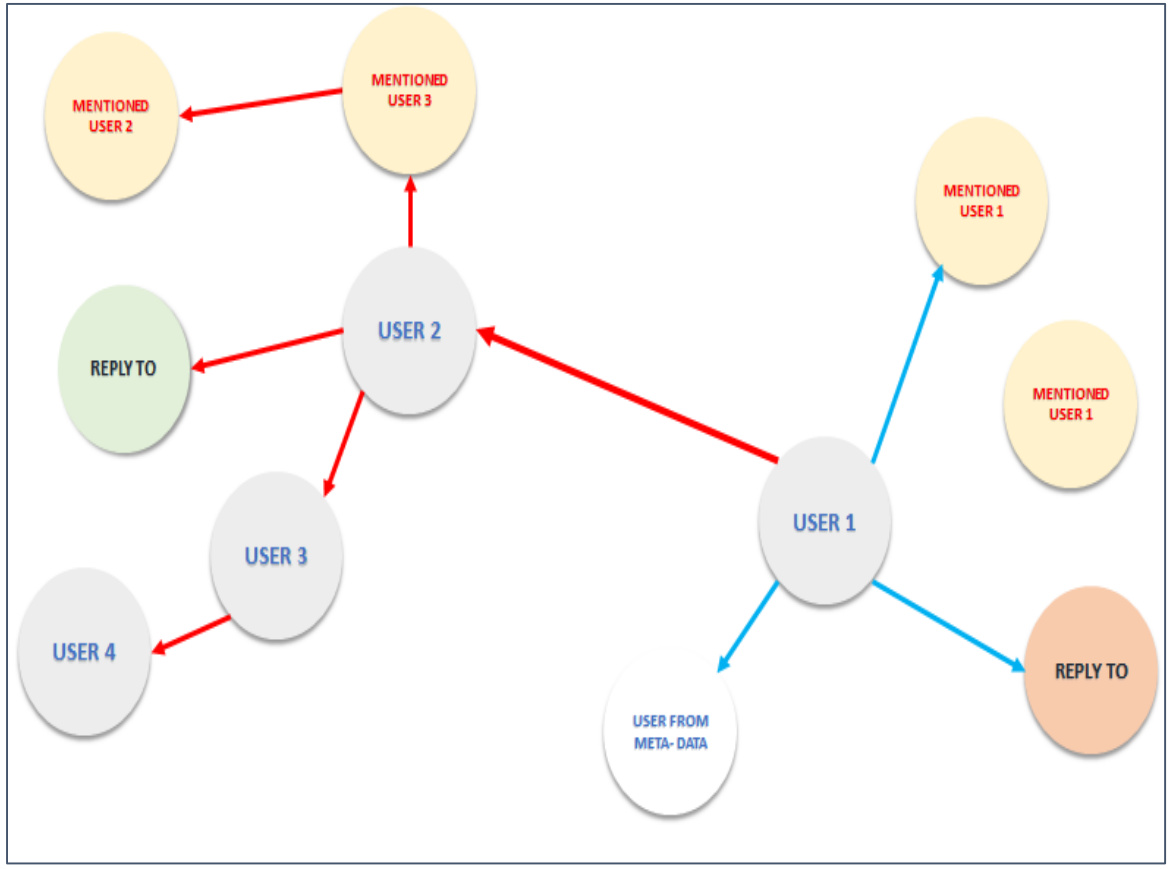
Embedding Type	Accuracy on 8 categories	Accuracy on 4 categories	Accuracy on 2 categories
Multi View Embedding Full using Bio BERT	0.3424	0.5752	0.7634
Multi View Embedding Reduced using Bio BERT	**0.3561**	**0.6525**	**0.7786**
Multi View Embedding Full using Sci BERT	0.3013	0.2741	0.7756
Multi View Embedding Reduced using Sci BERT	0.2739	0.4942	0.7778
Multi View Embedding Full using Clinical BERT	0.2100	0.4942	0.7670
Multi View Embedding Reduced using Clinical BERT	0.2602	0.5559	0.7526

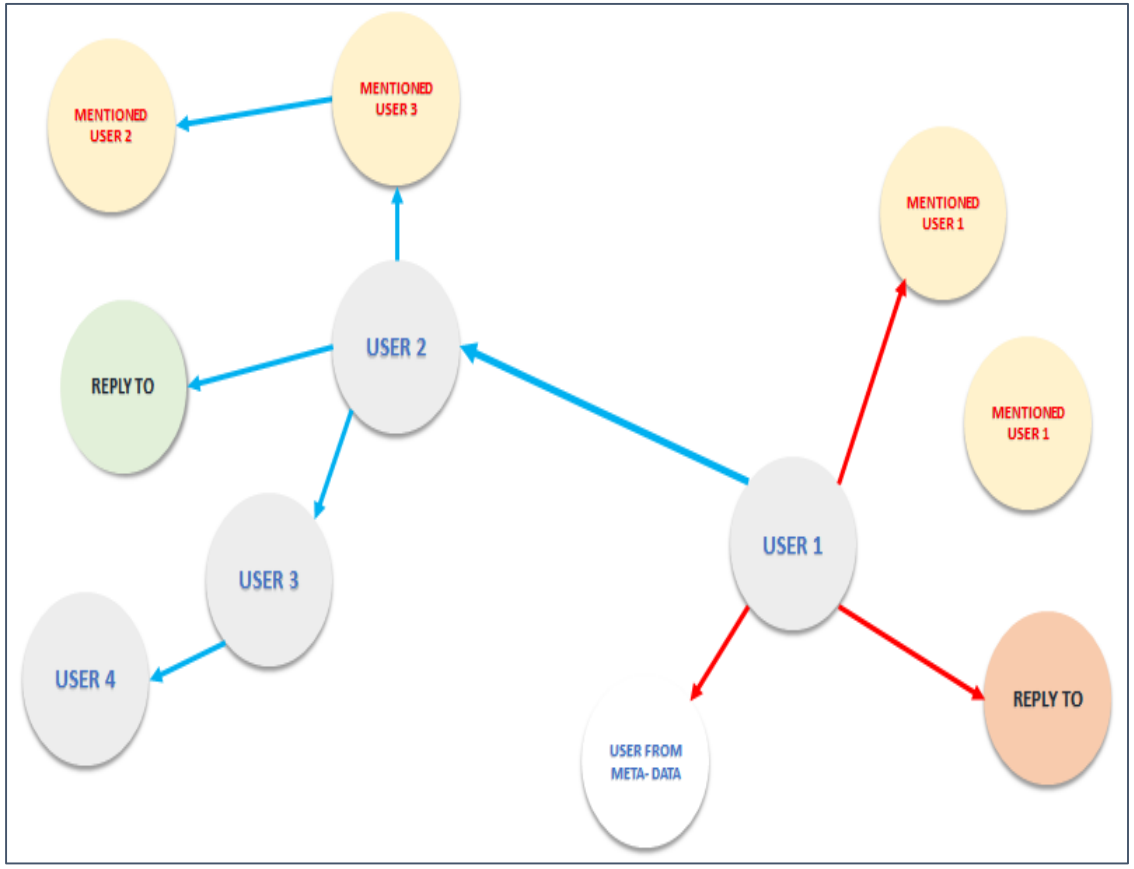
As can be observed there has been an improvement in the performance of the result.

Improvement Summary:

- Accuracy on 8 categories: **Almost same as best performing Bio BERT**
- Accuracy on 4 categories: **+8.88%**
- Accuracy on 2 categories: **Almost same as best performing Bio BERT**

Draw back of Deep Walk

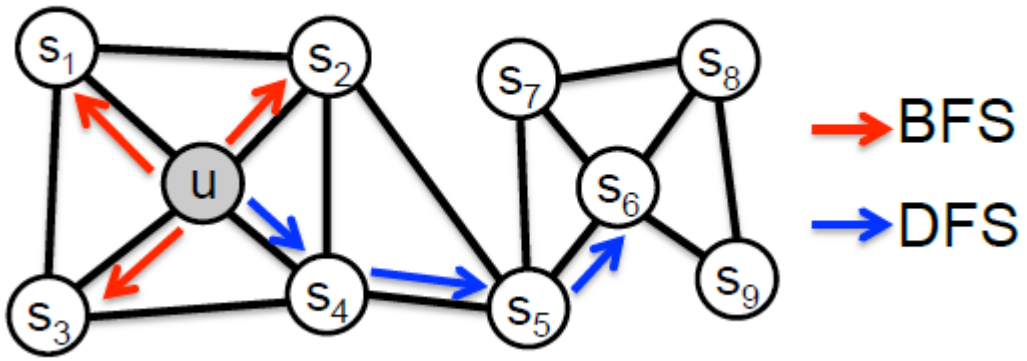




Blue path indicates the paths taken, while Red path indicates the paths neglected.

Node2Vec approach

To overcome the draw back of deep walk a state of the art approach called Node2Vec was introduced.
It uses BFS and DFS based algorithm to predict whether the path taken is too deep, that is based on DFS or too shallow, which is in case for neighbours only detected by BFS.
Node2Vec approach try to balance the path between going too deep or being at local node only with the help of probabilities associated with each edges.



Findings and results from Multi View using Node2Vec

The Multi View was applied exactly in the same order as was performed for deep walk, network representation. Below are the findings:

Embedding Type	Accuracy on 8 categories	Accuracy on 4 categories	Accuracy on 2 categories
Multi View Embedding Full using Bio BERT	0.3424	0.6370	0.7670
Multi View Embedding Reduced using Bio BERT	**0.3424**	**0.6949**	0.7734
Multi View Embedding Full using Sci BERT	0.3013	0.3243	**0.7992**
Multi View Embedding Reduced using Sci BERT	0.3150	0.5830	0.7778
Multi View Embedding Full using Clinical BERT	0.2785	0.5559	0.7670
Multi View Embedding Reduced using Clinical BERT	0.1917	0.6602	0.7455

As can be observed there has been an improvement in the performance of the result. Since we are considering only single method as best for evaluation, hence we are considering Bio Bert based approach as its performance is comparable in all 3 task.

Improvement Summary:

- Accuracy on 8 categories: **Almost same as best performing DEEP WALK**
- Accuracy on 4 categories: **+4.24%**
- Accuracy on 2 categories: **Almost same as best performing DEEP WALK**

Evaluation on Basis of Link Prediction

Traditional Link Prediction

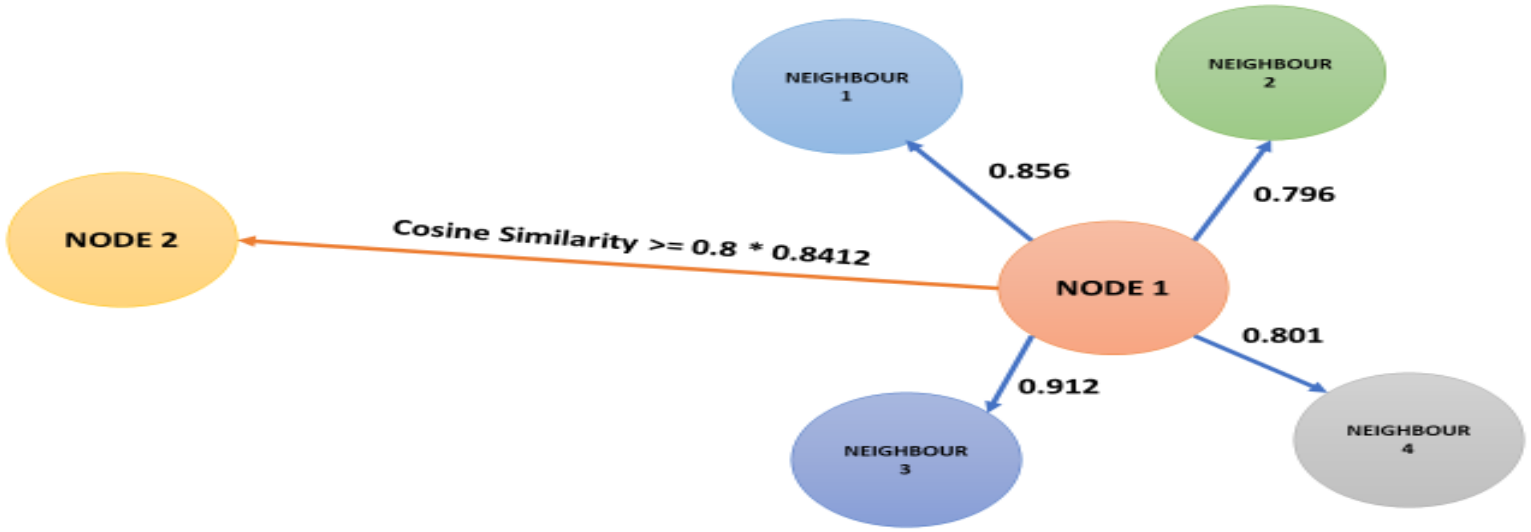
The traditional Link prediction method uses algorithms such as :
Jaccard Coefficient: It is calculated by number of common neighbours normalized by total number of neighbours.
Resource Allocation Index: It is defined as a fraction of a resource that a node can send to another through their common neighbours.
Adamic Adar: Predicts links on basis of links shared between two nodes.

However, these algorithms do not test the quality of nodes embedding in any way. Hence in order to use link prediction to evaluate the user embedding with add an additional constraint for link prediction task.

Proposed Modification

We infused a more strict constraint over the existing link prediction task by traditional method. A constraint that in order for a link or edge to be established between 2 nodes, the cosine similarity between the two nodes should be greater than a particular threshold score.

In order to decide the threshold we calculate the mean similarity score between nodes concerned and their neighbours MeanNodeA , MeanNodeB and for a link to be established the cosine similarity between these two nodes must be above a multiplicative factor of maximum score between MeanNodeA and MeanNodeB. This multiplicative factor is decided using standard precision and recall collected over training and validation data.



Experiments and Findings

We experimented with nearly 4280 users and the connection between them. We purposely masked or dropped almost 7016 edges from the network in a randomised fashion without making the dataset un-balanced. Between unconnected nodes there was a possibility of 2,61,345 edges that could have been formed. Next we predicted the edges on basis of our proposed algorithm. It is important to note that the correct predictions means the edges that were predicted matched the ones present in our test data. While other edges from test data were not successfully recognised.

Method	Total Number of edge prediction made	Correct Prediction	Accuracy
Post Doc Basic	51099	994	0.1416
User Doc Basic	51099	996	0.1419

The basic User Doc and Post Doc method have a very low accuracy score. However they have the most accurate edges filtered.

Results using Pre-trained Models

Method	Total Number of edge prediction made	Correct Prediction	Accuracy
Post Doc using Clinical Bert (Full)	95579	1784	0.2542
Post Doc using Clinical Bert (Reduced)	96173	1802	0.2568
Post Doc using Sci Bert (Full)	96319	1808	0.2576
Post Doc using Sci Bert (Reduced)	**96450**	**1820**	**0.2594**
Post Doc using Bio Bert (Full)	95898	1816	0.2588
Post Doc using Bio Bert (Reduced)	**96236**	**1820**	**0.2594**

It is important to note that embedding generated using both Sci-Bert and Bio-Bert had similar accuracy the only difference is the number of edges filtered for matching.

Multiview Embedding using Pre-trained Bert Model and Network representation using Deep Walk

Method	Total Number of edge prediction made	Correct Prediction	Accuracy
Post Doc using Clinical Bert and Deep Walk (Full)	95458	1784	0.2542
Post Doc using Clinical Bert and Deep Walk (Reduced)	95993	1806	0.2574
Post Doc using Sci Bert and Deep Walk (Full)	96232	1810	0.2579
Post Doc using Sci Bert and Deep Walk (Reduced)	**96448**	**1820**	**0.2594**
Post Doc using Bio Bert and Deep Walk (Full)	95787	1816	0.2588
Post Doc using Bio Bert and Deep Walk(Reduced)	**95408**	**1820**	**0.2594**

Multiview Embedding using Pre-trained Bert Model and Network representation using Node2Vec

Method	Total Number of edge prediction made	Correct Prediction	Accuracy
Post Doc using Clinical Bert and Node2Vec (Full)	95433	1784	0.2542
Post Doc using Clinical Bert and Node2Vec (Reduced)	96039	1806	0.2574
Post Doc using Sci Bert and Node2Vec (Full)	95780	1816	0.2588
Post Doc using Sci Bert and Node2Vec (Reduced)	**95462**	**1820**	**0.2594**
Post Doc using Bio Bert and Node2Vec (Full)	96237	1810	0.2579
Post Doc using Bio Bert and Node2Vec (Reduced)	**96450**	**1820**	**0.2594**

It is important to observe that there has not been any accuracy improvement for the task using the different embeddings but only lower number of total edges filtered. Hence we need another task to evaluate our embeddings.

Edge Classification

Proposed Experiment

We are taking into account all the edges that exist between users in our data set along with the concerned nodes and label them as positive edges or 1. We add some negative or 0 labelled data to this data set and then perform a binary classification task using the Jaccard coefficient, resource allocation and cosine similarity score between the nodes as the predictor variables.

The evaluation is performed on the basis of the weightage or the importance similarity scores plays in classifying the edges. The more weightage the similarity scores play the better the embedding. We will maintain a constant accuracy score of 97 percent for correctly classifying the edges using the basic Logistic regression model. We will then evaluate the weightage on the basis of variable coefficient predicted by the classifier model.

We are considering 20000 data set as a Training data, 2000 data set as a validation data to estimate our decision boundary and 7000 data as our Test data which results in nearly 97 percent accuracy score.

Findings and results

Training Data = 20000 edges
Validation Data = 1982 edges
Test Data = 7000 edges

Algorithm Used for Generating Embedding	Decision Boundary at cutoff	Jaccard Coefficient	Resource Allocation coefficient	Similarity Score Coefficient	Weightage of Similarity Score	Precision	Recall	Accuracy
Post Doc Basic	0.1	8.6187	6.4125	0.4574	2.95313973	0.9621	0.9792	0.971
User Doc Basic	0.1	7.6854	8.6254	0.9547	5.5295242	0.9574	0.9674	0.969

As observed the traditional User doc and Post doc generated user embedding has very little weightage or importance in making the classification decision.

Pre-trained Bert Model

Algorithm Used for Generating Embedding	Decision Boundary at cutoff	Jaccard Coefficient	Resource Allocation coefficient	Similarity Score Coefficient	Weightage of Similarity Score	Precision	Recall	Accuracy
Post Doc using Clinical Bert (Full)	0.25	8.3231	6.3722	3.903	20.98578902	0.9824	0.9716	0.977
Post Doc using Clinical Bert (Reduced)	0.35	8.5924	6.39	1.7316	10.3601771	0.98621	0.9692	0.9775
Post Doc using Sci Bert (Full)	0.1	7.8142	6.222	7.868	35.92005186	0.9532	0.988	0.971
Post Doc using Sci Bert (Reduced)	0.1	8.15182	6.2341	6.2571	30.31097194	0.9592	0.9855	0.972
Post Doc using Bio Bert (Full)	0.1	8.373	6.34	3.989	21.3292696	0.9643	0.9804	0.9727
Post Doc using Bio Bert (Reduced)	0.15	8.539	6.349	2.821	15.92975323	0.9761	0.9744	0.9754

As we move to a better embedding we can observe a jump in the weightage of the similarity score and the best performing Sci-BERT model based embedding almost shares equal weightage in edge classification as the other 2 scores.

Multi View Embedding using Deep Walk

Algorithm Used for Generating Embedding	Decision Boundary at cutoff	Jaccard Coefficient	Resource Allocation coefficient	Similarity Score Coefficient	Weightage of Similarity Score	Precision	Recall	Accuracy
Post Doc using Clinical Bert and Deep Walk (Full)	0.1	7.8943	6.2035	7.9844	36.15762922	0.9764	0.9753	0.976
Post Doc using Clinical Bert and Deep Walk (Reduced)	0.1	7.6449	6.0767	9.9442	42.01928521	0.9698	0.9791	0.9747
Post Doc using Sci Bert and Deep Walk (Full)	0.1	7.5785	6.1701	9.1246	39.8921008	0.9767	0.9753	0.9761
Post Doc using Sci Bert and Deep Walk (Reduced)	**0.1**	**6.6559**	**5.9058**	**13.1977**	**51.2345008**	**0.9764**	**0.9753**	**0.9761**
Post Doc using Bio Bert and Deep Walk (Full)	0.1	7.9545	6.261	7.4888	34.50376193	0.9704	0.9777	0.9742
Post Doc using Bio Bert and Deep Walk(Reduced)	**0.1**	**7.4696**	**6.0453**	**11.416**	**45.79056512**	**0.9698**	**0.9779**	**0.9741**

Significant improvement in the weightage with almost 50 percent decision making attribute.

Multi View Embedding using Node2Vec

Algorithm Used for Generating Embedding	Decision Boundary at cutoff	Jaccard Coefficient	Resource Allocation coefficient	Similarity Score Coefficient	Weightage of Similarity Score	Precision	Recall	Accuracy
Post Doc using Clinical Bert and Node2Vec (Full)	0.1	7.76154	6.0936	8.6296	38.3798078	0.9724	0.9777	0.9761
Post Doc using Clinical Bert and Node2Vec (Reduced)	0.1	7.4209	5.82224	10.8922	45.1296729	0.9695	0.9782	0.9741
Post Doc using Sci Bert and Node2Vec (Full)	0.1	7.5312	6.1411	9.3037	40.4931233	0.9767	0.9753	0.9761
Post Doc using Sci Bert and Node2Vec (Reduced)	**0.1**	**6.4119**	**5.6905**	**13.7279**	**53.1464985**	**0.9767**	**0.9753**	**0.9761**
Post Doc using Bio Bert and Node2Vec (Full)	0.1	7.8492	6.1712	8.1123	36.6530066	0.9704	0.9777	0.9742
Post Doc using Bio Bert and Node2Vec (Reduced)	**0.1**	**7.1729**	**5.732**	**12.1918**	**48.5792953**	**0.9689**	**0.9779**	**0.9737**

A jump of nearly 3 percent in the weightage for same accuracy score, when graph representation using Node2Vec was used to create the embedding.

Result Summary

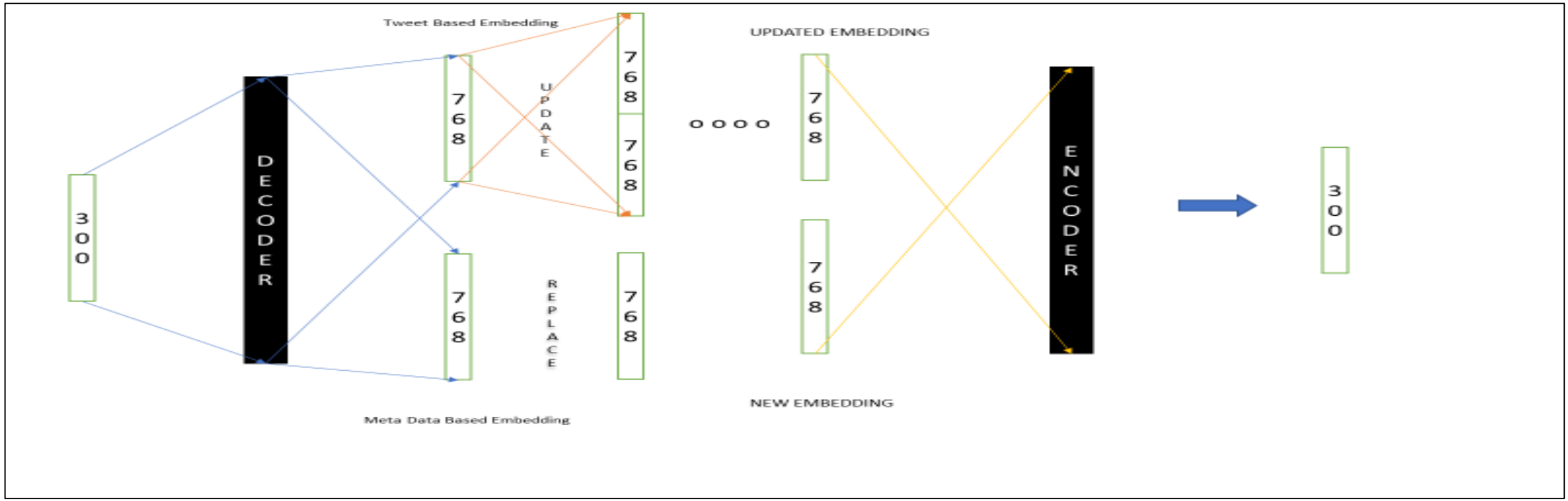
Algorithm to generate Embedding	Ranking based on performance for MultiClassification Task	Ranking based on performance for Link Prediction Task	Ranking based on performance for Edge Classification Task	Average Ranking	Final Standing
Post Doc Basic	11	11	11	11	**11**
User Doc Basic	10	10	10	10	**10**
Single View using Clinical Bert	9	9	9	9	**9**
Single View using Sci Bert	8	6	7	7	**8**
Single View using BioBert Bert	7	3	8	6	**5**
Multi View using Clinical Bert + Deep Walk	5	7	6	6	**7**
Multi View using SCI Bert + Deep Walk	6	4	2	4	**4**
Multi View using BIO Bert + Deep Walk	2	2	4	2.66	**2**
Multi View using Clinical Bert + Node2Vec	4	8	5	5.66	**6**
Multi View using SciBert + Node2Vec	3	1	1	1.66	**1**
Multi View using BIO Bert + Node2Vec	1	5	3	3	**3**

Updating Embedding

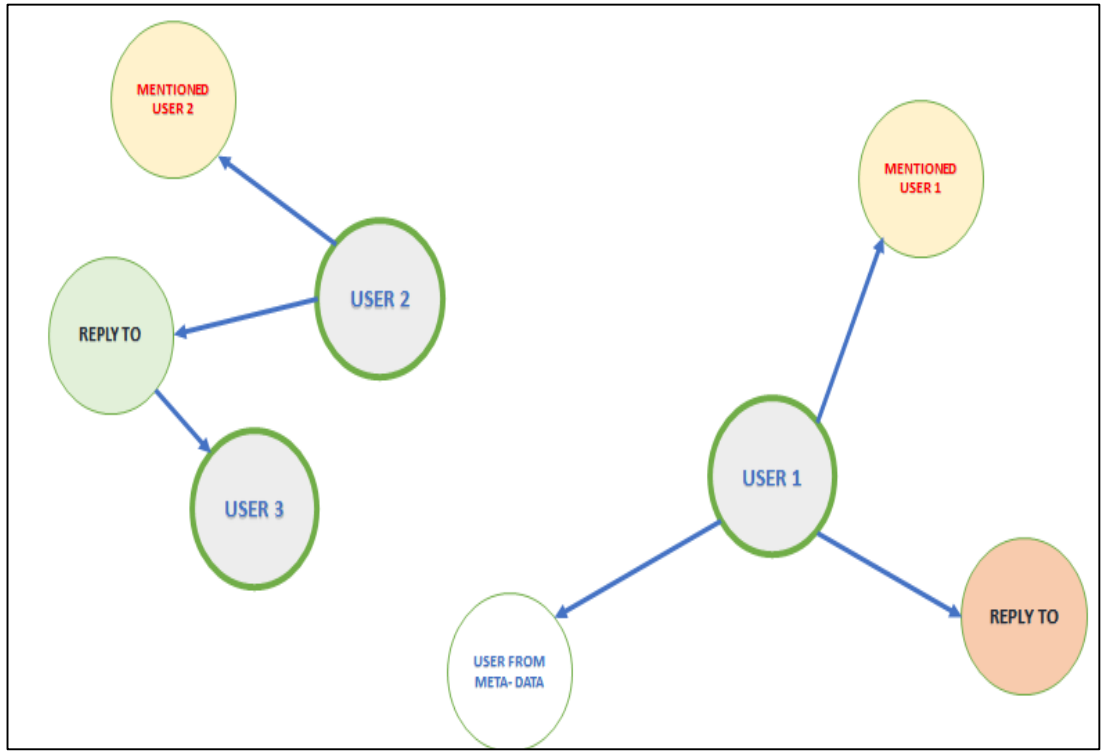
Proposed Method

We can apply the updating algorithm on Text based embedding obtained from tweets and Network based user embedding.

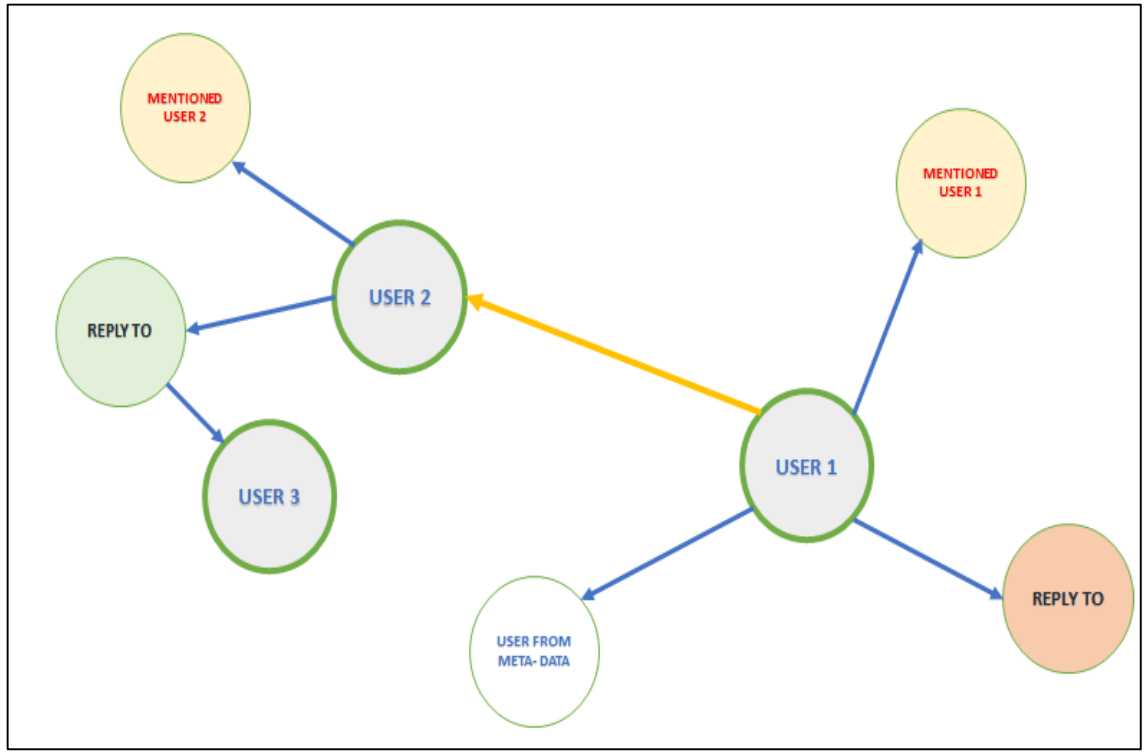
Embeddings generated from tweets can be updated with help of an Autoencoder where tweets from multiple time frame could be passed through pre trained Bert model and then merged together using encoder, there by avoiding any loss of information that existed previously and without increasing the dimension of the embedding.



When new tweets are encountered it is possible that the network of user changes. Hence the network will need an updating followed by creation of new embedding. Since we are using deep walk approach model need not be retrained for this purpose, however including and excluding edges between nodes must be taken care of.
Consider pre connection image below user1 will have random walk among nodes it is connected to, while user 2 will have random walk which will also include user 3 since they are indirectly connected.
Post connection the random walk space for all users are now increased and so are their network.



Pre Connection



Post Connection

Experiment and Findings

Following users were first categorised and then their user embedding was updated in order to check if their embedding changes.

We are considering two types of users: test user whose embedding will be updated and target users whose data (tweets) will be used to update the embeddings.

AUTHOR_ID	TRANS_AUTHOR_BIO	Snippet
CDFrazier87	['B.A., @DickinsonCol and M.A./Ph.D., @Georgetown. I like my hot dogs like I like my footnotes: Chicago-style. @WeAreGAGE for life. Views my own. He/his/him.']	['Diabetes, asthma, and heart disease, can result in complications if you catch #COVID19. They can also be very expensive for people to manage in the US today. But, politicians like @JoeBiden would have you think that #M4A has nothing to do with #COVID19. https://t.co/MiJatXH9SM']

Test User 1
Correctly Classified as
Student (category 3)

AUTHOR_ID	TRANS_AUTHOR_BIO	Snippet
BrukeGetachew	['Data scientist and support guru. Would-be software engineer. Currently @silamoney formerly @simple #BlackTechTwitter']	['RT @IcebergAk Funny how everything folks say they couldnâ€™t do theyâ€™ve showed they can now https://t.co/AZ6K7k5ORn']

Test User 2
Correctly Classified as
Others (category 4)

AUTHOR_ID	TRANS_AUTHOR_BIO	Snippet
Beth17459893	['Registered Nurse at a NYC Hospital,member of NYSNA, Pro-Animal, Pro-Environment, Compassionate about the world around me.']	['RT @CNN Pharmaceutical company Novo Nordisk said it will offer insulin free of charge for 90 days to diabetes patients who lost health insurance coverage because they lost their jobs https://t.co/Qdm30DJCBC', 'RT @crampell Coronavirus becomes number one cause of death per day in U.S., surpassing heart disease and cancer https://t.co/czaQANM9Wf', 'RT @NYGovCuomo Now more than ever, government needs to deliver results for the people. 2021 budget highlights: â€¢Enacts Paid Sick Leave â€¢Legalizes gestational surrogacy â€¢Enacts first-in-nation domestic terrorism law â€¢Caps insulin co-pays

Test User 3
Correctly Classified as
Nurse (category 1)

AUTHOR_ID	TRANS_AUTHOR_BIO	Snippet
AuntLowlen	["Aunt, psychologist/teacher's aid, book worm, music lover (classically trained soprano), red wine aficionado, survivor. Speaks fuck you cancer fluently."]	['RT @AyannaPressley Do you know why African-Americans align with so many COVID-19 co-morbidities? Structural racism. Food deserts=unequal access to healthy & fresh foods= diabetes, heart disease. Enviro injustices mean dirty air, water. Greed & neglect, subpar housing=asthma']

Target User
Correctly Classified as
Professor (category 3)

Observation

It was observed that the users test user 1 and test user 2 was correctly updated to category 3 of the target user. However test user 3 which was classified as nurse did not changed the category which might be possible due to lack of data that is in form of tweets from target user.

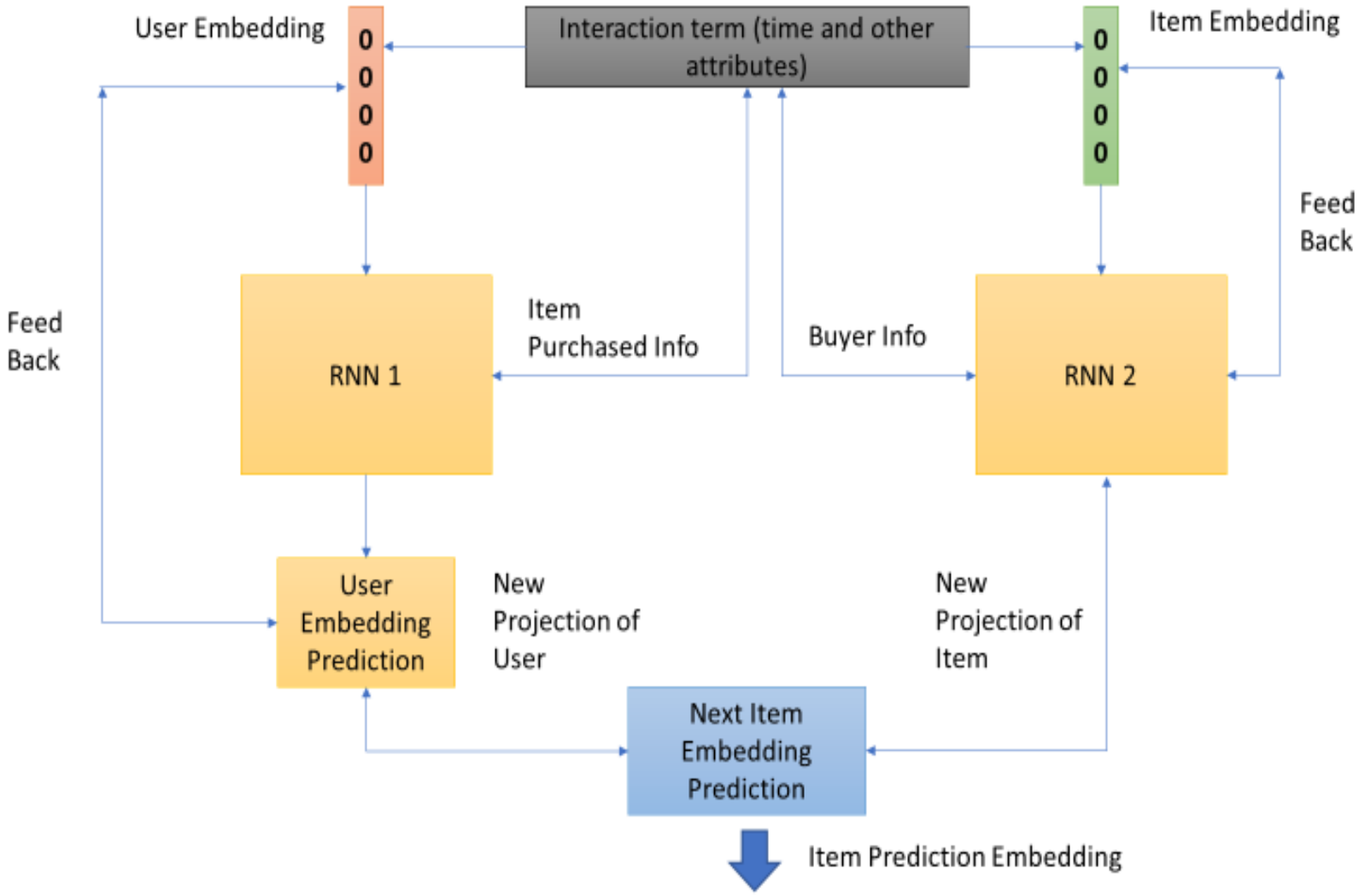
JODIE

Joint Dynamic User-Item Embedding

Proposed Algorithm

In an event when we have interaction activity of our users with respect to products like e-commerce or insurance policy etc. We can utilised the generated embeddings as a starting point to update the user embedding along with product embedding to produce prediction for future purchase.

The Architecture of JODIE model uses two RNN model to update user as well as product embedding in order to represent them in a combined feature space for better prediction.



Key Highlights

The Key highlight of Jodie algorithms are :

- **Loss function:** The loss function is calculated using the difference (mse) between the embedding of the product predicted and the embedding of the product actually purchased. This way the embedding is updated for every purchase.
- **RNN advantage:** Since RNN is used, the information related to history of the purchase is inscribed in the user embedding. Also the item embedding are updated accordingly and hence better recommendation could be achieved.
- **Generates Embedding:** The model generates embedding instead of any probability of an Item being purchased hence the recommendation could be made on basis of products with highest similarity to the predicted embedding.
- **Time could be incorporated in user Projection:** When predicting a new item, a time since last purchase can be incorporated as a latent cross matrix factorisation to project new user embedding to achieve more realistic prediction of items.

Thank You

Enjoyed working on it.