1 Section 3.1.3 : Similarity by Contrast

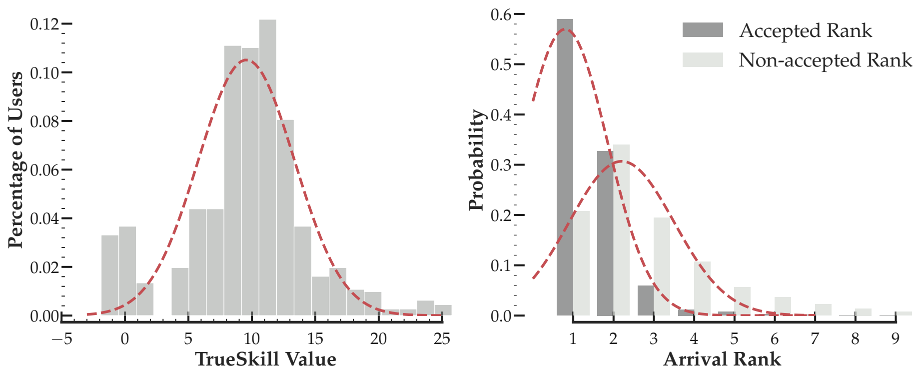
The exact view construction algorithm for similarity by contrast views are:

TrueSkill Similarity connects answers authored by a specific user, where the difference in his skill over peers is greater than margin δ. Specifically, if the user authors answers a, a' to questions q, q', we create a link between a and a' if

$$|S_{u,a} - S_{u,b}| > \delta; \forall b \in \mathcal{A}_(q)$$
$$|S_{u,a'} - S_{u,c}| > \delta; \forall c \in \mathcal{A}_(q')$$

where $S_{u,a}$ is the skill value for the user who authored answer a. Similarly, a link is created for the opposite case when difference is less than $-\delta$.

We estimate the user skill values with the TrueSkill rating system (https://pypi.org/project/trueskill/) computed with their historic performance in the community. TrueSkill values are normally distributed among users (fig. 1).



(a) TrueSkill Distribution

(b) ArrivalRank Distribution

Fig. 1: Distribution of the TrueSkill values of users and ArrivalRank of accepted answers and non-accepted answers for the movie StackExchange. Early answers are more likely to be accepted and variance of TrueSkill similarity across users is high.

Arrival Similarity The temporal arrival patterns of answers are correlated to their acceptance probabilities (fig. 1). For a specific user authoring answers a, a' to questions q, q', we establish a link between these answers if

$$|T_a - T_b| > \gamma \times \max(T_b); \forall b \in \mathcal{A}_(q)$$
$$|T_{a'} - T_c| > \gamma \times \max(T_c); \forall c \in \mathcal{A}_(q')$$

where T_a represents the relative time-gap between answer a and the question q. Conversely, we create links when difference is less than $-\gamma \times \max(T_b)$. Our hypotheis is that a similar answering schedule indicates similar user confidence or skill across questions.

2 Section 4.2 : Contrastive Convolution

In this section, we provide a theoretical analysis of our contrastive convolution. The ability of neural networks to perform classification in sparse high-dimensional manifolds has been studied in past work, especially in the context of adversarial learning [4]. We employ the ReLU activation function in our convolution layers and study the outputs of the kth layer, i.e. embeddings with k-order locality. This transformation breaks the input space into cells with smooth gradients within each cell, at whose boundaries the piecewise linear function changes (i.e. the likelihood of the two classes of answers).

We ask a specific question in the context of our Contrastive IR-GCN. *What is the impact of the layerwise discriminative magnification induced by our formulation?* Discriminative magnifications results in improved separability of the two classes in the later convolving layers, an effect we earlier demonstrated with a sample network in **??**. This positively impacts the ability of the model to explain the observed data points (i.e create p-domains that are well aligned with the contrastive samples provided) and improve the generalizability of the learned model to unseen data points. However, it is important to maintain sufficient regularization with weight decay to prevent sparse regions exhibiting sharp gradients which could affect model performance.

The capacity of our model can also be quantified in terms of the VC dimension of the aggregated classifier against the individual learners. Gradient boosting with multiple relation learners (each of which captures a specific aspect of node locality via graph convolution on the induced relations) could boost the capacity of the joint model, enabling better generalization and a more accurate fit in the data manifold (i.e. higher capacity to fit regions to fine distinctions).

Let us denote the upper bound of the VC dimension or capacity of each individual learner as D (If the individual learners do not have identical capacity, the minimum can be used to compute a lower bound on the aggregated learner capacity). Then the gradient boosted learner with T classifiers has a bound on it's capacity [6] given by,

$$\mathcal{VC}_{Agg} = T \times (D+1) \times (3\log(T.(D+1)) + 2)$$

Thus we identify two potential reasons for our performance gains, first the discriminative magnification effect which also supports the strong individual performance of the contrast view, and second the gain in capacity from boosting, which could explain it's advantage over competing aggregation methods.

3 Section 5 : Aggregating Induced Views

The prior aggregation methods used in the literature were adapted for our GCN model as follows:

Neighborhood Aggregation: is a multi relational approach that aggregates neighbors of the nodes from all views. Thus, the final adjacency matrix is the sum of all the individual adjacency matrices of each view, i.e., $A = \sum_{S_i \in \mathbf{S}} A_i$. We, then, apply Graph Convolution Network to this updated Adjacency matrix.

Stacking: stacks all GCNs belonging to a view such that output of a lower GCN is fed as an input to the GCN directly above it. Thus, output from the last layer of GCN for view i, Z_i^K s.t. $S_i \in \mathbf{S}$ will act as input features, Z_j^0 for some other view j s.t. $S_j \in \{\mathbf{S} - S_i\}$ if view j is directly above the view i. In our experiments, we obtain the best performance by using the following order: Contrastive, Similarity by Contrast followed by Reflexive.

Fusion: treats each GCN as a separate model and appends the output from the final layer of each GCN i.e. $Z_i^K; \forall S_i \in \mathbf{S}$ to the input of all the other GCN's, i.e. $Z_j^0 \forall S_j \in \mathbf{S} - S_i$ along with the original features. Thus, the input of each GCN is linear in $|\mathbf{S}|$.

4 Section 6.1 : Dataset

The list of 50 StackExchange communities per category are;

- Technology: AskUbuntu, Server Fault, Unix, TEX, Electronics, Gis, Apple, Wordpress, Drupal, DBA
- Culture/Recreation: English, Travel, RPG, Judaism, Puzzling, Bicycles, German, Christianity, BoardGames, History
- Life/Arts: Scifi, DIY, Academia, Graphic Design, Money, Photo, WorldBuilding, Movies, Music, Law
- Science: Stat, Physics, MathOverflow, CS, Chemistry, Biology, Philosophy, CS Theory, Economics, Astronomy
- Professional/Business: Workplace, Aviation, Writers, Open source, Freelancing, CS Educators, Quant, PM, Parenting

5 Section 6.2.1 : Baselines

Dual GCN (DGCN) The mean squared error (MSE) between vertex representations of two views is defined as, for instance,

$$\mathcal{L}_{reg}(Z_c, Z_{ts}) = \frac{1}{n} \sum_{j \in 1,n} \|Z_c^j - Z_{ts}^j\|$$

computes the MSE loss between Contrastive and TrueSkill Similarity GCN. [8] proposed the model for two GCN representations. We extend the original two GCN model [8] to four GCN, each representing our relational views between the nodes. We minimize the supervised loss with respect to the best performing GCN model (Contrastive), and its node representation's alignment with respect to all other GCN's node representations. The Contrastive view is seen to exhibit the best performance in isolation. Thus, the DualGCN loss can be given by:

$$\mathcal{L} = \mathcal{L}_0 + \lambda(t) \left(\sum_{S_i \in \mathbf{S}, S_i \neq c} \|\mathbf{Z}_c^K - \mathbf{Z}_i^K\| \right)$$

where $\mathcal{L}_0$ represents the supervised loss and $\mathbf{Z}_c^K$ is the node representations of the Contrastive GCN.

Relational GCN (RGCN) [5] combines the output representations of previous layer of each view $\mathbf{Z}_i^{k-1}$ of layer $k-1$ of each view to compute an aggregated input to layer k. Formally,

$$\mathbf{Z}_{rgcn}^k = \sigma \left(\sum_{S_i \in \mathbf{S}} \mathbf{Z}_i^{k-1} \right)$$

where Z_{rgcn} is final output of this model at layer k and σ is the activation function.

6 Section 6.3 : Performance Analysis

The breakdown of the results by the StackExchange community are given in the tables below.(Technology: table 1, Culture: table 2, Life: table 3, Science: table 4, Business: table 5).

Method	**AskUbuntu**		**Serverfault**		**Unix**		**TEX**		**Electronics**	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	64.63	0.668	68.38	0.648	70.90	0.597	64.93	0.717	67.69	0.699
FF [3]	68.49	0.789	68.29	0.769	73.14	0.730	65.43	0.795	68.26	0.797
DGCN [8]	70.25	0.794	72.01	0.768	75.16	0.749	69.02	0.788	70.62	0.783
RGCN [5]	55.24	0.636	58.04	0.689	64.14	0.610	56.22	0.750	53.18	0.640
AS-GCN	70.15	0.774	68.09	0.751	74.90	0.749	65.91	0.782	67.91	0.775
TS-GCN	69.49	0.782	67.15	0.752	73.95	0.748	64.67	0.777	66.58	0.777
C-GCN	72.27	0.792	72.42	0.772	76.44	0.761	69.10	0.794	72.45	0.797
IR-GCN	**75.25**	**0.798**	**73.87**	**0.776**	**78.75**	**0.765**	**70.96**	**0.796**	**74.63**	**0.799**

Method	**GIS**		**Apple**		**Wordpress**		**Drupal**		**DBA**	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	64.93	0.698	70.13	0.644	65.11	0.718	65.65	0.727	65.44	0.718
FF [3]	64.69	0.786	68.86	0.785	64.62	0.805	64.88	0.799	66.39	0.803
DGCN [8]	68.51	0.775	73.52	0.770	70.11	0.808	68.22	0.791	69.58	0.793
RGCN [5]	50.30	0.671	49.48	0.627	55.89	0.729	51.26	0.688	50.30	0.686
AS-GCN	64.82	0.771	69.76	0.776	65.00	0.791	65.06	0.785	65.95	0.793
TS-GCN	64.13	0.799	68.94	0.773	63.80	0.795	64.41	0.786	65.55	0.797
C-GCN	69.12	0.787	73.08	0.785	70.80	0.809	70.69	0.802	69.99	0.818
IR-GCN	**71.20**	**0.791**	**75.85**	**0.791**	**73.39**	**0.814**	**73.43**	**0.802**	**72.25**	**0.806**

Table 1: Accuracy and MRR values for StackExchange communities belonging to **Technology** category with state-of-the-art baselines.

7 Section 6.6 : Textual Features

Text Preprocessing: We first removed all code snippets, HTML tags, stopwords and URLs from the text of all questions and answers. We then tokenized the text using NLTK tokenizer followed by lemmatization using WordNetLemmatizer and finally converted it into lowercase.

We use torchtext to create vocabulary and limit the text of each question and answer to be 250 words long. We initialized the words in the vocabulary using 300-dimensional pretrained embeddings from Word2vec (https://code.google.com/archive/p/word2vec/). We randomly intiailized words present in the vocabulary but not in word2vec.

QA-LSTM/CNN: In this baseline, we use a biLSTM model with hidden dimension = 300 followed by 50 1D convolutional filters with a kernel size of 3. We then compute the final embeddings by applying 1D maxpooling on the output of the convolution layer. We also used Tanh nonlinearity and dropout of 0.3 on the final embeddings. We finally use these embeddings to compute a cosine simlarity score between a question and its answers. This score is used to rank the candidate answers for evaluation. We implemented the baseline in Pytorch.

Method	English		Travel		RPG		Judaism		Puzzling	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	73.98	0.564	70.94	0.655	73.29	0.590	70.16	0.665	73.94	0.548
FF [3]	73.36	0.755	73.70	0.802	72.92	0.765	67.96	0.780	73.90	0.738
DGCN [8]	77.03	0.750	76.05	0.796	74.47	0.738	72.68	0.752	75.74	0.738
RGCN [5]	60.70	0.621	61.95	0.546	60.45	0.639	57.59	0.670	60.35	0.621
AS-GCN	74.52	0.736	72.66	0.791	74.60	0.752	70.24	0.754	75.09	0.722
TS-GCN	73.97	0.736	71.44	0.794	73.20	0.752	69.57	0.763	74.15	0.727
C-GCN	76.67	0.752	76.49	0.802	77.07	0.767	73.31	0.783	77.47	0.745
IR-GCN	**78.74**	**0.761**	**78.96**	**0.810**	**79.13**	**0.773**	**75.58**	**0.798**	**79.75**	**0.747**

Method	Bicycles		German		Christianity		BoardGames		History	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	74.09	0.604	69.85	0.675	74.67	0.607	71.49	0.677	72.83	0.683
FF [3]	73.86	0.789	69.56	0.781	72.26	0.791	72.01	0.814	72.69	0.806
DGCN [8]	76.38	0.769	72.28	0.773	74.23	0.775	76.71	0.814	76.73	0.810
RGCN [5]	59.74	0.651	58.47	0.675	60.78	0.685	60.74	0.686	63.21	0.661
AS-GCN	74.74	0.757	69.25	0.757	74.01	0.772	71.23	0.785	74.14	0.805
TS-GCN	73.89	0.755	68.24	0.761	73.88	0.774	69.22	0.790	74.08	0.791
C-GCN	77.79	0.773	73.02	0.779	76.63	0.786	76.43	0.816	76.90	0.815
IR-GCN	**80.18**	**0.785**	**75.11**	**0.786**	**79.59**	**0.806**	**78.84**	**0.819**	**80.22**	**0.821**

Table 2: Accuracy and MRR values for StackExchange communities belonging to **Culture** category with state-of-the-art baselines.

Textual Similarity (T-GCN) We extract the updated embeddings of the question and answer text from the learnt QA-LSTM model. We then compute cosine similarity between the embeddings of each question and its answers. Specifically, we connect answers authored by a specific user, where the difference in cosine similarity of the answer with the other competing answers is greater than margin λ. Specifically, if the user authors answers a, a' to questions q, q', we create a link between a and a' if

$$|C_{q,a} - C_{q,b}| > \lambda; \forall b \in \mathcal{A}(q)$$
$$|C_{q,a'} - C_{q,c}| > \lambda; \forall c \in \mathcal{A}(q')$$

where $C_{q,a}$ is the cosine similarity of the answer a with respect to question q. Similarly, a link is created for the opposite case when difference is less than $-\lambda$. In our experiments, we assign $\lambda = 0.4$. The hypothesis is that irrelevant(dissimilar) answers will more likely be rejected and vice versa.

8 Reddit Experiments

Reddit[1] is another popular CQA platform with subreddits similar to StackExchange communities. In particular, we focus on Ask* subreddits as they are primarily used to seek help from a community of experts and non-experts. In particular, we crawled data from /r/askscience (science forum), /r/AskHistorians (history forum), and /r/AskDocs (medical forum) until October 2017. We performed basic preprocessing and removed posts or comments with single word/URLs or missing author/title information. We also removed infrequent users who posted less than two comments. Table 6 shows the final data statistics.

[1] https://www.reddit.com/

Method	SciFi		DIY		Academia		Graphic Design		Money	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	77.15	0.675	71.52	0.651	74.37	0.560	69.82	0.638	72.19	0.608
FF [3]	76.57	0.826	70.76	0.795	73.62	0.767	69.79	0.795	71.06	0.777
DGCN [8]	79.27	0.818	74.67	0.794	76.92	0.757	74.43	0.792	74.80	0.770
RGCN [5]	59.56	0.651	58.08	0.690	64.36	0.672	59.11	0.712	60.58	0.652
AS-GCN	74.67	0.816	70.18	0.778	75.26	0.732	70.01	0.784	72.19	0.755
TS-GCN	61.13	0.793	69.40	0.788	74.20	0.740	69.17	0.782	70.98	0.751
C-GCN	79.77	0.824	74.68	0.798	77.75	0.759	74.51	0.800	75.18	0.771
IR-GCN	**81.60**	**0.834**	**76.83**	**0.805**	**79.62**	**0.772**	**76.13**	**0.804**	**77.11**	**0.784**

Method	Photo		WorldBuilding		Movies		Music		Law	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	69.27	0.580	84.34	0.455	70.91	0.704	70.33	0.628	67.27	0.787
FF [3]	73.66	0.747	83.77	0.720	73.75	0.824	76.07	0.773	66.89	0.821
DGCN [8]	75.78	0.752	84.72	0.710	77.62	0.839	76.81	0.785	72.31	0.823
RGCN [5]	56.12	0.624	68.73	0.546	61.97	0.649	53.95	0.614	57.34	0.735
AS-GCN	74.60	0.751	84.18	0.701	74.02	0.836	76.62	0.790	66.14	0.822
TS-GCN	74.19	0.727	83.90	0.660	73.40	0.831	76.96	0.784	66.88	0.799
C-GCN	76.88	0.757	85.35	0.710	78.25	0.843	77.23	0.794	74.09	0.828
IR-GCN	**78.26**	**0.770**	**86.71**	**0.731**	**80.22**	**0.850**	**79.39**	**0.807**	**76.20**	**0.841**

Table 3: Accuracy and MRR values for StackExchange communities belonging to **Life** category with state-of-the-art baselines.

Reddit has a hierarchical comment structure. For this paper, we treat first-level comments as potential answers to the question. Users in these subreddits can get verified by providing anonymized verification documents including certification numbers, contact information, etc. to the moderators. We denote these verified users as experts. We treat an expert's comment as equivalent to an accepted answer and only consider posts which have an expert answer for our experiments. We discard posts with multiple experts' comment as it is hard to objectively choose a winner.

We employ 12 basic features for the Reddit dataset:
Activity features : ArrivalRank of the answer, Number of subsequent comments on the answer, Number of other answers to the question, Upvotes and downvotes for both, question and answer.
Text features : Word count of the question and answer

We employ post-vote features here as [2] showed that there is widespread under-provision of voting on Reddit, partially due to long comment threads. It can act as a weak signal for answer quality. Unlike the StackExchange, Reddit voting is not biased by publicly visible acceptance of answers to a question. Thus, votes ideally represent the independent judgment of the crowd.

Table 7 shows performance gains over the state-of-art baselines for the Reddit dataset. All results are reported after 5-fold cross validation. Our model improves by 16% on average in accuracy over the baselines for Reddit. The improvement in MRR is again higher for Reddit at an average increase of 7% than the baseline.

Among individual views, for Reddit, there is a huge difference in performance for each GCN. TrueSkill Similarity performs much better followed by Arrival Similarity and Contrastive. Reflexive GCN performs the worst for Reddit as it predicts each node's label independent of answers to the same question.

Method	Stat		Physics		MathOverflow		CS		Chemistry	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	67.97	0.706	67.75	0.679	67.22	0.649	66.23	0.700	68.04	0.759
FF [3]	67.97	0.817	67.41	0.803	66.76	0.767	65.56	0.794	68.31	0.836
DGCN [8]	70.53	0.799	70.27	0.774	67.38	0.713	69.49	0.806	73.52	0.826
RGCN [5]	53.16	0.661	62.05	0.650	71.88	0.644	57.15	0.701	57.46	0.729
AS-GCN	67.84	0.801	67.85	0.788	56.80	0.743	65.71	0.779	67.82	0.823
TS-GCN	66.48	0.805	67.20	0.790	57.32	0.744	64.45	0.781	64.99	0.832
C-GCN	72.41	0.815	65.08	0.803	62.23	0.762	69.76	0.796	73.52	0.845
IR-GCN	**74.78**	**0.818**	**74.68**	**0.802**	**71.88**	**0.768**	**73.12**	**0.809**	**76.58**	**0.848**

Method	Biology		Philosophy		CS Theory		Economics		Astronomy	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	65.81	0.730	73.32	0.601	70.15	0.643	65.14	0.713	69.27	0.743
FF [3]	65.74	0.831	73.69	0.752	68.88	0.775	65.53	0.807	68.83	0.816
DGCN [8]	70.77	0.823	75.84	0.761	71.61	0.782	72.42	0.810	72.67	0.820
RGCN [5]	56.87	0.731	61.36	0.609	58.46	0.688	55.88	0.730	61.05	0.682
AS-GCN	65.87	0.802	75.14	0.754	69.39	0.763	65.06	0.794	67.79	0.827
TS-GCN	64.24	0.824	74.53	0.748	67.64	0.760	65.64	0.804	66.56	0.808
C-GCN	71.65	0.833	76.57	0.748	72.70	0.774	71.16	0.798	73.05	0.829
IR-GCN	**74.45**	**0.837**	**78.82**	**0.766**	**75.50**	**0.789**	**73.15**	**0.812**	**76.84**	**0.837**

Table 4: Accuracy and MRR values for StackExchange communities belonging to **Science** category with state-of-the-art baselines.

Out of the baseline graph ensemble approaches, DualGCN and RelationalGCN, similar to StackExchange, DualGCN consistently performs better than RelationalGCN by an average of around 3% for Reddit.

References

1. Burel, G., Mulholland, P., Alani, H.: Structural normalisation methods for improving best answer identification in question answering communities. In: International Conference on World Wide Web, WWW (2016)
2. Gilbert, E.: Widespread underprovision on reddit. In: Conference on Computer Supported Cooperative Work. CSCW '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2441776.2441866
3. Jenders, M., Krestel, R., Naumann, F.: Which answer is best?: Predicting accepted answers in MOOC forums. In: International Conference on World Wide Web (2016)
4. Lu, J., Issaranon, T., Forsyth, D.A.: Safetynet: Detecting and rejecting adversarial examples robustly. In: ICCV. pp. 446–454 (2017)
5. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. Springer (2018)
6. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)
7. Tian, Q., Zhang, P., Li, B.: Towards predicting the best answers in community-based question-answering services. In: International Conference on Weblogs and Social Media, ICWSM (2013)
8. Zhuang, C., Ma, Q.: Dual graph convolutional networks for graph-based semi-supervised classification. In: World Wide Web Conference (2018)

Method	Workplace		Aviation		Writers		Open Souce		Freelancing	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	76.27	0.546	72.79	0.691	75.68	0.502	67.97	0.706	75.08	0.634
FF [3]	75.77	0.726	74.65	0.819	75.91	0.726	69.43	0.837	72.19	0.760
DGCN [8]	77.21	0.725	78.57	0.818	78.22	0.732	72.77	0.775	74.31	0.756
RGCN [5]	62.98	0.590	60.57	0.722	65.76	0.629	59.42	0.714	62.23	0.722
AS-GCN	76.91	0.699	72.44	0.815	77.26	0.707	69.42	0.783	71.08	0.762
TS-GCN	76.02	0.709	72.99	0.808	76.72	0.717	68.45	0.806	71.13	0.744
C-GCN	78.94	0.726	78.62	0.822	78.73	0.746	71.88	0.794	74.44	0.781

Method	CS Educators		Quant		PM		Parenting	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF [1,7]	80.88	0.528	68.61	0.671	80.18	0.572	75.04	0.507
FF [3]	81.95	0.702	68.87	0.805	75.95	0.745	76.97	0.717
DGCN [8]	82.63	0.661	73.34	0.804	77.94	0.740	76.79	0.748
RGCN [5]	70.23	0.599	57.36	0.710	65.69	0.647	62.97	0.578
AS-GCN	82.75	0.677	70.15	0.788	76.35	0.732	78.56	0.720
TS-GCN	81.72	0.706	67.41	0.796	75.16	0.732	77.96	0.704
C-GCN	84.65	0.759	73.43	0.806	78.52	0.728	78.91	0.751
IR-GCN	**85.20**	**0.770**	**76.92**	**0.817**	**81.14**	**0.757**	**81.61**	**0.771**

Table 5: Accuracy and MRR values for StackExchange communities belonging to **Business** category with state-of-the-art baselines.

Dataset	$\lvert\mathcal{Q}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{U}\rvert$	$\mu(\lvert\mathcal{A}_q\rvert)$
AskDocs	11189	29207	4530	2.61
AskHistorians	15425	45586	11761	2.96
AskScience	37990	121278	32117	3.19

Table 6: Dataset statistics for the Ask* Reddit communities. $|Q|$: number of questions; $|\mathcal{A}|$: number of answers; $|U|$: number of users; $\mu(|\mathcal{A}_q|)$: mean number of answers per question.

Method	AskDocs		AskHistorians		AskScience	
	Acc (%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF	59.35	0.698	65.62	0.709	65.87	0.706
FF	62.30	0.715	67.89	0.730	68.99	0.713
DGCN	77.54	0.790	80.49	0.805	75.57	0.821
RGCN	57.98	0.667	64.56	0.684	62.42	0.642
AS-GCN	76.53	0.794	80.7	0.781	78.14	0.797
TS-GCN	84.44	0.861	90.95	0.8289	87.61	0.822
C-GCN	67.39	0.753	70.57	0.7441	71.11	0.769
IR-GCN	**87.60**	**0.896**	**93.81**	**0.851**	**89.11**	**0.837**

Table 7: Accuracy and MRR values for Ask Reddits. Our model significantly outperforms by 16% in Accuracy and 7% in MRR. TrueSkill Similarity performs best among individual IR-GCNs.