Research: Effective Neural Team Formation via Negative Samples

ABSTRACT

Forming teams of experts who collectively hold a set of required skills and can successfully cooperate is challenging due to the vast pool of feasible candidates with diverse backgrounds, skills, and personalities. Neural models have been proposed to address scalability while maintaining efficacy by learning the distributions of experts and skills from successful teams in the past in order to recommend future teams. However, such models are prone to overfitting when training data suffers from a long-tailed distribution, i.e., few experts have most of the successful collaborations, and the majority has participated sparingly. In this paper, we present an optimization objective that leverages both successful and *virtually unsuccessful* teams to overcome the long-tailed distribution problem. We propose three negative sampling heuristics that can be seamlessly employed during the training of neural models. We study the synergistic effects of negative samples on the performance of neural models compared to lack thereof on two large-scale benchmark datasets of computer science publications and movies, respectively. Our experiments show that neural models that take unsuccessful teams (negative samples) into account are more efficient and effective in training and inference, respectively.

1 INTRODUCTION

Collaborative teams are the primary vehicle for coordinating experts with diverse skills for a particular project, and team formation has firsthand effects on creating an organizational performance [3, 5, 6, 11, 17, 21]. Examples include forming a research group on *'machine learning'* whose success can be measured by scientific publications, or a movie's cast and crew for the next blockbuster *'sci-fi'* movie with a touch of *'drama'*. Forming a successful team whose members can effectively collaborate and deliver the outcomes within the specified constraints, such as planned budget and timeline, is challenging due to the immense number of candidates with various backgrounds, skills, and personality traits, as well as unknown synergistic balance among them; not all teams with best experts are necessarily successful [18].

Researchers have proposed neural machine learning models that learn relationships among experts and their social attributes through neural architectures. They consider all past successful team compositions as training samples to predict optimum teams for a given set of required skills in order to bring efficiency while maintaining efficacy due to the inherently iterative and online learning procedure in neural architectures. Among the first, Sapienza et al. [16] proposed a neural autoencoder to form an optimum team. Autoencoders are, however, prone to overfitting and are not able to capture the uncertainty in sparse data [4]. Rad et al. [14] have shown that training datasets in team formation suffer from the long-tailed distribution; that is, few experts have most of the successful collaborations for a small set of skills while the majority has participated sparingly. As a result, popular experts receive higher scores for the given skills and are more frequently recommended, leading to popularity bias. Rad et al. [14], hence, employed a variational Bayesian neural model to overcome the performance drain of the long tail problem through uncertainty on weights of the neural model. Existing neural models are, however, trained solely on successful teams and overlook *un*successful ones.

In this paper, we propose an optimization objective that leverages both successful and unsuccessful teams via various negative sampling heuristics and investigate the synergistic effect of utilizing unsuccessful teams during training of the state-of-the-art variational Bayesian neural model [14] as well as non-Bayesian neural models. Literature has shown that leveraging not only positive samples (e.g., friendship in social networks) but also negative samples (e.g., distrust) convey complementary signals to neural models and improve accuracy in various tasks [8, 10, 12, 13, 15, 19, 22].

Most real-world training datasets in the team formation domain, however, do not have explicit unsuccessful teams (e.g., collections of rejected papers), or what constitutes a failure remains controversial (e.g., movie's box office vs. critical reviews). In the absence of unsuccessful teams and based on the closed-world assumption, we presume a subset of experts as an unsuccessful team if they have not already collaborated for the required skills. To this end, we develop three negative sampling heuristics: 1) *uniform*: where subsets of experts are randomly chosen with the same probability as samples of unsuccessful teams, 2) *unigram*: where subsets of experts are chosen based on their frequencies in the training set, and 3) *smoothed unigram in training minibatches*: where we employed Laplace smoothing to calculate the unigram probability of subsets of experts in each training minibatch. In unigram and smoothed unigram heuristics, experts that have collaborated more often on skills different from the given input skills will be chosen more frequently to mitigate popularity bias. We utilize negative samples during training to incite the neural models to learn vector representations (embeddings) for experts and skills in the same vector space such that vectors of experts who have already collaborated for the required skills (have been in the same teams) end up closer to each other whereas vectors of experts who have not collaborated yet (virtually unsuccessful teams) become distant. We reproduce the proposed Bayesian and non-Bayesian neural models under the negative sampling heuristics, and lack thereof, on two large-scale datasets from different domains with distinct statistical distributions of skills in teams: *i*) computer science papers (`dblp`) and *ii*) movies (`imdb`). The empirical results show that incorporating negative samples consistently exhibits a stronger predictive power for the optimum team in Bayesian neural models. However, non-Bayesian neural models are sensitive to the distribution of teams over skills and negative samples may have adverse impacts on effectiveness, as seen in the movies dataset (`imdb`).

2 NEURAL TEAM FORMATION

We aim to find an optimal team of experts who collectively hold a set of required skills and can work together to yield success. Given a set of m skills $\mathcal{S} = \{i\}_1^m$ and a set of n experts $\mathcal{E} = \{j\}_1^n$, t_{se} is a team of experts $e \subseteq \mathcal{E}; e \neq \emptyset$, that collectively hold a subset of skills $s \subseteq \mathcal{S}; s \neq \emptyset$, and $\mathcal{T} = \{(t_{se}, y); y \in \{0, 1\}\}$ indexes all previous successful and unsuccessful teams. Given a subset of skills s and all the previous collaborations $\mathcal{T}$, we aim at identifying an optimal subset of experts e such that their collaboration in the predicted team (s, e) will be successful, that is $(t_{se}, y = 1)$, and avoiding subset of experts e' that $(t_{se'}, y = 0)$. More concretely, we aim to estimate a mapping function f of parameters θ from a subset of skills and experts to a boolean set; $f_\theta : P(\mathcal{S}) \times P(\mathcal{E}) \longrightarrow \{0, 1\}$.

Given all previous collaborations $\mathcal{T} = \{(t_{se}, y); y \in \{0, 1\}\}$, we maximize the average log probability of teams' success or failure:

$$\frac{1}{|\mathcal{T}|} \sum_{(t_{se}, y) \in \mathcal{T}} \log \mathrm{P}(y|t_{se}) \tag{1}$$

where t_{se} is a team of experts e who collectively hold the set of skills s and can either work successfully together or fail otherwise. We propose to learn vector representations (embeddings) for experts and skills in the same vector space with the expectation that vectors of experts whose teams have been successful for the required skills will end up closer to each other in the vector space while vectors of experts whose teams for the required skills have been unsuccessful will end up farther from each other. We estimate the $\mathrm{P}(y|t_{se})$ through pairwise cosine similarities of vector representations for the skills $\forall i \in s$ and experts $\forall j \in e$. Specifically, for a successful team $(t_{se}, y = 1)$, we estimate $\mathrm{P}(y = 1|t_{se})$ by learning $v_s = \sum_{i \in s} v_i$ and $v_e = \sum_{j \in e} v_j$ that are close in the vector space and have high cosine similarity while for an unsuccessful team $(t_{se}, y = 0)$, we estimate $\mathrm{P}(y = 0|t_{se})$ by learning v_s and v_e that are far from each other and have low cosine similarity. Formally, $\mathrm{P}(y|t_{se})$ can be formulated using the sigmoid function σ:

$$P(y|t_{se}) = \sigma(v_e^\top \cdot v_s) \tag{2}$$

where v_s and v_e are the vector representations of the skill and expert subsets, respectively.

3 NEGATIVE SAMPLING HEURISTICS

Most available data in team formation only consists of successful teams. The dblp dataset of published research papers in computer science does not have unsuccessful submissions. In the imdb dataset of movies, it remains controversial what constitutes a failure for a movie; its reception by the people (box office) or critical reviews. In the absence of unsuccessful training instances, we follow the closed-world assumption that no currently known successful team for the required skills is considered unsuccessful. We assume that groups of experts e who have little or no collaborative experience for the required set of skills, i.e., few or no t_{se}, have a low chance for a successful collaboration.

Inspired by [8, 10, 12, 13, 19], we propose an optimization function that discriminates successful from unsuccessful teams through negative sampling from a distribution over the subsets of experts:

$$\sum_{t_{se} \in \mathcal{T}} [\log \sigma(v_e^\top \cdot v_s) + \sum_{t_{se'} \sim \mathbb{P}: t_{se'} \notin \mathcal{T}}^{k} \log \sigma(-v_{e'}^\top \cdot v_s)] \tag{3}$$

where $\mathbb{P}$ is the probability distribution from which we draw k subsets of experts e' as negative samples for a given subset of skills s where $t_{se} \in \mathcal{T}$ but $t_{se'} \notin \mathcal{T}$. We present three different negative sampling distributions, two static negative sampling distributions [12] and an adaptive noise distribution [2, 7, 13], and study their effects on neural models:

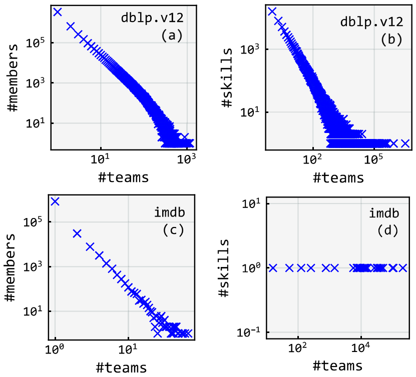

Figure 1: Distribution of teams over skills and members.

(1) **uniform distribution**, where each subset of experts e' is chosen with the same probability from the uniform distribution over all subsets of experts $\mathcal{P}(\mathcal{E})$, i.e. $\mathrm{P}(e') = \frac{1}{|\mathcal{P}(\mathcal{E})|}$
(2) **unigram distribution**, where each subset of experts e' is chosen regarding their frequency in all previous teams, i.e. $\mathrm{P}(e') = \frac{|t_{s'e'}|}{|\mathcal{T}|}$ and $t_{s'e'}$ is a team with skill subset $s' \neq s$. Intuitively, subsets of experts that have been in previous teams for other subsets of skills will be given a higher probability and chosen more frequently as negative samples to dampen the effect of popularity bias.
(3) **smoothed unigram distribution *in training minibatch***, where we employed the add-1 or Laplace smoothing when computing the unigram distribution of the experts in each training minibatch, i.e. $\mathrm{P}(e') = \frac{1+|t_{s'e'}|}{|b|+|\mathcal{E}|}$, where b is a minibatch subset of $\mathcal{T}$, and $t_{s'e'}$ is a successful team including expert e' in each training minibatch. Minibatch stochastic gradient descent is the *de facto* method for neural models where the data is split into batches of data, each of which is sent to the model for partial calculation in order to speed up training while maintaining high accuracy. Since only a few teams of experts exist in each minibatch, we employ the Laplace smoothing so that no subsets of experts have zero probability.

4 EXPERIMENTS

In this section, we lay out the details of our experiments and findings toward answering the following research questions[1]:

RQ1: Does negative sampling improve the effectiveness of neural models for the task of team formation? To this end, we benchmark the state-of-the-art Bayesian neural model [14] as well as non-Bayesian neural baselines with our proposed negative sampling heuristics compared to lack thereof.

RQ2: Are the impacts of negative sampling heuristics robust across different training datasets with diverse statistical characteristics? We benchmark baselines for the proposed negative sampling heuristics on computer science publications and movies, respectively.

RQ3: How does negative sampling help efficiency of neural models during training while improving inference effectiveness. In this regard, we benchmark the inference performance of the baselines on the test set in an increasing number of training epochs.

[1] Codebase and complete results will be available upon acceptance.

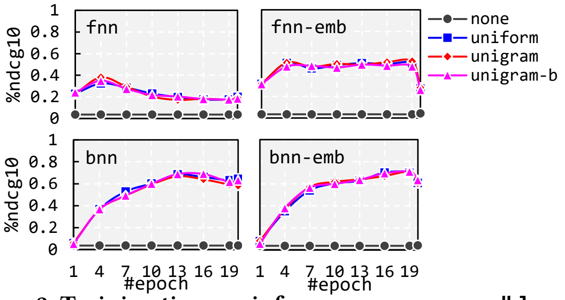

Figure 2: Training time vs. inference accuracy on dblp.v12.

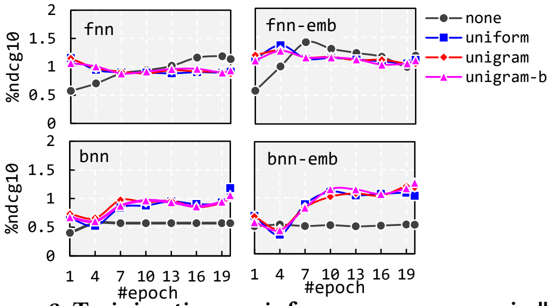

Figure 3: Training time vs. inference accuracy on imdb.

4.1 Setup

4.1.1 Dataset. Our testbed includes two datasets, namely, imdb[1] and dblp.v12[20]. In dblp.v12, each instance is a publication in computer science consisting of authors and fields of study (fos). We map each publication to a team whose authors are the experts and fields of study are the set of skills. In imdb, each instance is a movie consisting of its cast and crew such as actors and director, as well as the movie's genres. We consider each movie as a team whose members are the cast and crew, and the movie's genres are the skills.

In both datasets, we can observe long tails in the distributions of teams over experts. As shown in Figure 1a,c, many experts (researchers in dblp, and cast and crew in imdb) have participated in very few teams (papers in dblp and movies in imdb). However, imdb and dblp are following different distributions with respect to the set of skills. While dblp suffers further from the long-tailed distribution of skills in teams (Figure 1b), imdb follows a more fair distribution (Figure 1d). Specifically, imdb has a limited variety of skills (genres) which are, by and large, employed by many movies.

4.1.2 Baselines. Our testbed includes two neural architectures: *i*) feed-forward non-Bayesian (non-variational) neural network (fnn) with eq. 3 as the optimization function, and *ii*) Bayesian (variational) neural network [14] (bnn) with Kullback-Leibler optimization. Both models include a single hidden layer of size d=100, leaky relu and sigmoid are the activation functions for the hidden and the output layers, respectively, and Adam is the optimizer. The input and output layers are sparse occurrence vector representations (one-hot encoded) of skills and experts of size $|\mathcal{S}|$ and $|\mathcal{E}|$, respectively. Moreover, we also used pre-trained dense vector representations for the input skill subsets (-emb). Adapted from paragraph vectors of Le and Mikolov [9], we consider each team as a document and the skills as the document's words. We used the distributed memory model to generate the real-valued embeddings of the subset of skills with dimension of d=100. We train the models with a learning rate of 0.1 over 20 epochs including minibatches of size 4096. We evaluate baselines with and without our proposed negative sampling heuristics (-uniform, -unigram, -unigram-b). To have a minimum level of comparison, we also add a model that randomly assigns experts to a team (random). In total, we compare 16 + 1 baselines.

4.1.3 Evaluation Strategy and Metrics. To demonstrate prediction effectiveness, we randomly select 15% of teams for the test set and perform 5-fold cross-validation on the remaining teams for model training and validation that results in one trained model per each fold. Given a team t_{se} from the test set, we compare the ranked list of experts e', predicted by the model of each fold, with the observed subset of experts e and report the average performance of models on all folds in terms of normalized discounted cumulative gain (ndcg), and mean average precision (map) at top-{2,5,10} as well as precision (pr), recall (rec), and area under the receiver operating characteristic (rocauc). To evaluate training efficiency vs. inference efficacy, we train the baselines on an increasing number of epochs and evaluate them on the test set at each epoch.

4.2 Results

In response to **RQ1**, i.e., whether negative sampling improves the effectiveness of neural models, from Table 1 and 2, we can observe that **(1)** all negative sampling heuristics improve Bayesian neural baselines on dblp and imdb in terms of all metrics. In comparison, Bayesian baselines with no negative sampling (bnn and bnn-emb) are the weakest. Specifically, smoothed unigram negative sampling in minibatches (bnn-unigram-b and bnn-emb-unigram-b) consistently outperforms all other neural baselines in terms of ndcg for top-{2,5,10} and rocauc. Further, Bayesian baselines with dense vector representations of skills outperform Bayesian baselines with sparse vectors which is in line with Rad et al's experiments.

Contrary to Bayesian baselines, non-Bayesian baselines (fnn-*) do not show a consistent trend across datasets which bring us to our second research question **RQ2**, i.e., whether the impact of negative sampling heuristics is consistent across training data from diverse statistical distributions. From Table 1, we can see that negative sampling heuristics improve non-Bayesian baselines in dblp in terms of all metrics. However, in imdb, we cannot observe a consistent synergistic trend by using negative sampling heuristics. Indeed, non-Bayesian baseline *without* negative samplings (fnn) is the strongest baseline in terms of map, ndcg, precision (pr) and recall (rec) for top-{2,5}. We attribute the inefficiency of negative sampling heuristics for neural models on imdb to the small size of skill set (genres) and uniform distribution of teams (movies) over skills. Almost all the genres are fairly adopted by many movies, as seen in Figure 1d. The fact that dense vector representations for skills are not effective for non-Bayesian baselines in imdb is further cementing this view. Overall, **(2)** we conclude that the effect of considering unsuccessful teams via negative sampling in non-Bayesian neural models depends on the underlying distribution of teams over skills in the training set (dblp vs. imdb). Moreover, **(3)** in our experiments, that Bayesian neural models outperform non-Bayesian ones, reported earlier by Rad et al. on dblp, could *not* be generalized to imdb.

Table 1: Average performance of 5-fold neural models on test set in dblp.v12.

	%map2	%map5	%map10	%ndcg2	%ndcg5	%ndcg10	%pr2	%pr5	%pr10	%rec2	%rec5	%rec10	rocauc
random	0.0001	0.0001	0.0002	0.0002	0.0002	0.0004	0.0001	0.0002	0.0002	0.0001	0.0003	0.0006	0.4992
fnn	0.0045	0.0307	0.0612	0.0082	0.0227	0.0369	0.0045	0.0113	0.0155	0.0067	0.0188	0.0188	0.5000
fnn-uniform	0.0487	0.0741	0.0943	0.1074	0.1350	0.1993	0.1020	0.1030	0.0986	0.0597	0.1522	0.2913	0.6512
fnn-unigram	0.0437	0.0677	0.0880	0.0952	0.1249	0.1907	0.0932	0.0985	0.0971	0.0552	0.1447	0.2854	0.6505
fnn-unigram-b	0.0436	0.0665	0.0847	0.1005	0.1249	0.1846	0.0993	0.0979	0.0932	0.0569	0.1429	0.2702	0.6500
fnn-emb	0.0084	0.0402	0.0688	0.0134	0.0302	0.0428	0.0073	0.0174	0.0209	0.0134	0.0255	0.0215	0.4999
fnn-emb-uniform	0.0668	0.1043	0.1305	0.1537	0.1901	0.2716	0.1543	0.1505	0.1346	0.0870	0.2179	0.3925	0.6313
fnn-emb-unigram	0.0700	0.1084	0.1356	0.1564	0.1942	0.2803	0.1523	0.1500	0.1378	0.0884	0.2194	0.4038	0.6331
fnn-emb-unigram-b	0.0656	0.1015	0.1277	0.1444	0.1782	0.2607	0.1415	0.1374	0.1291	0.0830	0.2011	0.3770	0.6322
bnn	0.0061	0.0254	0.0569	0.0123	0.0204	0.0365	0.0061	0.0107	0.0155	0.0101	0.0161	0.0195	0.5000
bnn-uniform	0.0487	0.0741	0.0943	0.1074	0.1350	0.1993	**0.4005**	**0.3555**	**0.3102**	**0.2297**	**0.5124**	**0.8974**	0.7150
bnn-unigram	0.0437	0.0677	0.0880	0.0952	0.1249	0.1907	0.3388	0.2885	0.2670	0.1944	0.4175	0.7737	0.7132
bnn-unigram-b	0.1757	0.2499	0.3039	0.3983	0.4505	0.6312	0.3904	0.3402	0.3017	0.2256	0.4929	0.8774	**0.7168**
bnn-emb	0.0112	0.0326	0.0634	0.0175	0.0267	0.0413	0.0089	0.0145	0.0186	0.0168	0.0201	0.0201	0.5000
bnn-emb-uniform	0.1620	0.2296	0.2817	0.3663	0.4176	0.5895	0.3656	0.3405	0.2909	0.2121	0.4934	0.8463	0.7123
bnn-emb-unigram	**0.1792**	**0.2569**	0.3060	0.4022	0.4623	0.6205	0.3783	0.3298	0.2858	0.2213	0.4802	0.8313	0.7077
bnn-emb-unigram-b	0.1752	0.2537	**0.3094**	**0.4069**	**0.4728**	**0.6515**	0.3938	0.3518	0.3033	0.2252	0.5065	0.8775	0.7090

Table 2: Average performance of 5-fold neural models on test set in imdb.

	%map2	%map5	%map10	%ndcg2	%ndcg5	%ndcg10	%pr2	%pr5	%pr10	%rec2	%rec5	%rec10	rocauc
random	0.0006	0.0009	0.0012	0.0017	0.0018	0.0031	0.0017	0.0015	0.0018	0.0008	0.0018	0.0045	0.4988
fnn	**0.3639**	**0.5009**	0.5663	**1.0322**	**0.9760**	1.1331	**1.0252**	**0.7927**	0.5556	**0.4681**	**0.9187**	1.2931	0.5061
fnn-uniform	0.2038	0.3272	0.4191	0.5802	0.6470	0.9116	0.5760	0.5664	0.5219	0.2634	0.6593	1.2292	0.5930
fnn-unigram	0.1977	0.3085	0.4003	0.5522	0.6035	0.8773	0.5386	0.5190	0.5078	0.2492	0.6054	1.1884	0.5924
fnn-unigram-b	0.2141	0.3329	0.4329	0.5911	0.6448	0.9363	0.5864	0.5598	0.5377	0.2767	0.6551	1.2788	0.5941
fnn-emb	0.2487	0.3784	0.4977	0.7057	0.7898	1.1805	0.7028	0.6970	0.7024	0.3243	0.8082	1.6275	0.5585
fnn-emb-uniform	0.2476	0.3872	0.4961	0.6979	0.7817	1.1083	0.7049	0.6979	0.6467	0.3278	0.8029	1.4972	0.6063
fnn-emb-unigram	0.2433	0.3703	0.4740	0.6758	0.7344	1.0442	0.6654	0.6338	0.6026	0.3110	0.7381	1.3961	0.6075
fnn-emb-unigram-b	0.2678	0.3929	0.4974	0.7559	0.7807	1.1027	0.7507	0.6654	0.6267	0.3443	0.7776	1.4650	0.6064
bnn	0.0490	0.1227	0.2231	0.1301	0.3027	0.5725	0.1560	0.3327	0.4200	0.0842	0.3775	0.9536	0.5000
bnn-uniform	0.3167	0.4441	0.5397	0.8560	0.8735	1.1833	0.8193	0.7261	0.6484	0.3863	0.8431	1.5104	0.6305
bnn-unigram	0.2568	0.3749	0.4550	0.7423	0.7651	1.0246	0.7320	0.6563	0.5673	0.3347	0.7711	1.3280	0.6304
bnn-unigram-b	0.2483	0.3810	0.4670	0.6954	0.7673	1.0580	0.6883	0.6654	0.6001	0.3226	0.7795	1.3931	0.6313
bnn-emb	0.0642	0.1334	0.2024	0.1695	0.3124	0.5180	0.1643	0.3194	0.3531	0.0834	0.3696	0.8102	0.5000
bnn-emb-uniform	0.2855	0.3970	0.4730	0.7739	0.7766	1.0370	0.7424	0.6355	0.5548	0.3544	0.7575	1.3128	0.6262
bnn-emb-unigram	0.2820	0.4243	0.5256	0.7969	0.8727	1.1852	0.7819	0.7569	0.6675	0.3620	0.8767	1.5490	0.6338
bnn-emb-unigram-b	0.3267	0.4692	**0.5728**	0.9207	0.9272	**1.2662**	0.8858	0.7802	**0.7053**	0.3982	0.9002	**1.6341**	**0.6470**

In response to **RQ3**, i.e., whether negative sampling increases the efficiency of neural models during training while improving inference effectiveness, from Figure 2 and 3, we can observe that **(4)** Bayesian neural models that benefit from negative samples outperforms other models in less number of training epochs for sparse and dense vector representation across all datasets in terms of ndcg10. With respect to the non-Bayesian neural models, we can observe similar synergistic effects of negative sampling heuristics on obtaining the *best* inference effectiveness with a fewer training epochs over dblp. However, we cannot observer similar trend over imdb. In fact, **(5)** non-Bayesian neural models *without* negative sampling (fnn and fnn-emb) could gradually gain the momentum and achieve the stellar performance over imdb at epoch 7 and after. This observation further explains that when teams are *well*-distributed over a limited set of skills (e.g., movies over genres), overly usage of negative samples in many epochs of training decouples the vectors of experts and skills that should have been stayed close for their participation in successful teams, and consequently degrades the inference performance.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed three negative sampling heuristics to utilize the synergistic effect of virtually unsuccessful teams during neural model training. Our experiment, when performed on two large-scale datasets with distinct distributions of teams over skills and experts, show that (1) negative sampling improves the effectiveness of Bayesian neural models for the task of team formation; (2) depending on the distribution of teams over skills, while improving the performance of non-Bayesian neural baselines in datasets with a large variety of skills (e.g., dblp), negative sampling may discount the efficacy of neural models in datasets with limited skill set (e.g., imdb); and (3) negative sampling helps with efficiency during training while improving inference effectiveness for Bayesian neural models. For future work, we aim at reproducing neural models on other datasets like patents as teams of inventors. We also aim at identifying real unsuccessful teams, e.g., for publications based on their sleeping time in arXiv[2] or budget-box office ratio for movies.

[2] arxiv.org

REFERENCES

[1] [n.d.]. IMDb Datasets. https://www.imdb.com/interfaces/. Accessed: 2022-05-14.

[2] Kian Ahrabian, Aarash Feizi, Yasmin Salehi, William L. Hamilton, and Avishek Joey Bose. 2020. Structure Aware Negative Sampling in Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6093–6101. https://doi.org/10.18653/v1/2020.emnlp-main.492

[3] Rodrigo Borrego Bernabé, Iván Álvarez Navia, and Francisco José García-Peñalvo. 2015. Faat: Freelance as a Team. In *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality* (Porto, Portugal) *(TEEM '15)*. Association for Computing Machinery, New York, NY, USA, 687–694. https://doi.org/10.1145/2808580.2808685

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1613–1622. https://proceedings.mlr.press/v37/blundell15.html

[5] K.M. Bursic. 1992. Strategies and benefits of the successful use of teams in manufacturing organizations. *IEEE Transactions on Engineering Management* 39, 3 (1992), 277–289. https://doi.org/10.1109/17.156562

[6] Maxine Craig and Debi McKeown. 2015. How to build effective teams in healthcare. *Nursing times* 111, 14 (2015), 16—18. http://europepmc.org/abstract/MED/26182585

[7] Shabnam Daghaghi, Tharun Medini, Nicholas Meisburger, Beidi Chen, Mengnan Zhao, and Anshumali Shrivastava. 2021. A Tale of Two Efficient and Informative Negative Sampling Distributions. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 2319–2329. https://proceedings.mlr.press/v139/daghaghi21a.html

[8] Jérôme Kunegis, Julia Preusse, and Felix Schwagereit. 2013. What is the Added Value of Negative Links in Online Social Networks?. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 727–736. https://doi.org/10.1145/2488388.2488452

[9] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32* (Beijing, China) *(ICML'14)*. JMLR.org, II–1188–II–1196.

[10] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting Positive and Negative Links in Online Social Networks. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. Association for Computing Machinery, New York, NY, USA, 641–650. https://doi.org/10.1145/1772690.1772756

[11] Joyce Magill-Evans, Megan Hodge, and Johanna Darrah. 2002. Establishing a transdisciplinary research team in academia [Electronic version]. *Journal of allied health* 31 (02 2002), 222–6.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 3111–3119.

[13] Pengda Qin, Weiran Xu, and Jun Guo. 2016. A Novel Negative Sampling Based on TFIDF for Learning Word Representation. *Neurocomput.* 177, C (feb 2016), 257–265. https://doi.org/10.1016/j.neucom.2015.11.028

[14] Radin Hamidi Rad, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2020. Learning to Form Skill-based Teams of Experts. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2049–2052. https://doi.org/10.1145/3340531.3412140

[15] Steffen Rendle and Christoph Freudenthaler. 2014. Improving Pairwise Learning for Item Recommendation from Implicit Feedback. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining* (New York, New York, USA) *(WSDM '14)*. Association for Computing Machinery, New York, NY, USA, 273–282. https://doi.org/10.1145/2556195.2556248

[16] Anna Sapienza, Palash Goyal, and Emilio Ferrara. 2019. Deep Neural Networks for Optimal Team Composition. *Frontiers Big Data* 2 (2019), 14. https://doi.org/10.3389/fdata.2019.00014

[17] Peter D. Sherer. 1995. Leveraging Human Assets in Law Firms: Human Capital Structures and Organizational Capabilities. *ILR Review* 48, 4 (1995), 671–691. https://doi.org/10.1177/001979399504800405 arXiv:https://doi.org/10.1177/001979399504800405

[18] Roderick I. Swaab, Michael Schaerer, Eric M. Anicich, Richard Ronay, and Adam D. Galinsky. 2014. The Too-Much-Talent Effect: Team Interdependence Determines When More Talent Is Too Much or Not Enough. *Psychological Science* 25, 8 (2014), 1581–1591. https://doi.org/10.1177/0956797614537280 arXiv:https://doi.org/10.1177/0956797614537280 PMID: 24973135.

[19] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. 2016. A Survey of Signed Network Mining in Social Media. *ACM Comput. Surv.* 49, 3, Article 42 (aug 2016), 37 pages. https://doi.org/10.1145/2956185

[20] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. https://www.aminer.org/citation. In *KDD'08*. 990–998.

[21] Julie Younglove-Webb, Barbara Gray, Charles William Abdalla, and Amy Purvis Thurow. 1999. The Dynamics of Multidisciplinary Research Teams in Academia. *The Review of Higher Education* 22, 4 (1999), 425–440. http://muse.jhu.edu/journals/review_of_higher_education/v022/22.4younglove-webb.html

[22] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing Top-n Collaborative Filtering via Dynamic Negative Item Sampling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland) *(SIGIR '13)*. Association for Computing Machinery, New York, NY, USA, 785–788. https://doi.org/10.1145/2484028.2484126