Organization Prediction for Scientists and Scholars for OAG Data

Timothy Le
Department of Computer Science
Stanford University
Stanford, CA 94305
tle7@stanford.edu

Jacqueline Yau
Department of Computer Science
Stanford University
Stanford, CA 94305
jyau@stanford.edu

Abstract

The Open Academic Graph (OAG) dataset is a large dataset with missing entries; in particular the field we are interested in predicting is an author's current organization. In this paper, we seek to understand what types of approaches allow us to predict the current organization successfully. We consider feature-based and graph-based approaches, including a knowledge-graph approach. We find that a knowledge-graph approach based on the ConvE [2] model yields the highest accuracy for predicting the current organization.

1 Introduction/Motivation/Problem Definition

Research over various fields has grown significantly over the last few years, connecting scholars from all over the world in sharing new ideas and discoveries. With this vast network of researchers and locations at which to conduct research, scientists and scholars often migrate from their institutions to other locations better suited for their research interests. We are interested in tracking the movement of these scholars, as they go to different places to continue and further their research.

Using a large-scale academic graph, we can try to track the migration of these scientists. As scientists and scholars relocate during their careers, their university affiliations change, thus the current university affiliation is not always known and is difficult to keep track of. We would like to perform link prediction on the most likely current affiliations of scientists and scholars given by the Open Academic Graph (OAG) from [6] and [7].

A common approach to predicting the author's current affiliation would be to model this data as a knowledge graph and perform knowledge graph completion, since for each author, the known data is not consistent. In particular, not every author's current affiliation/organization is known, so knowledge graph completion can be performed. While performing knowledge graph completion is viable, we are interested to see if such an approach is too complex. Therefore, in this paper, we seek to predict an author's current organization based on the following approaches: a feature vector based prediction approach, an unsupervised random walk approach, and a ConvE [2] approach.

2 Related Work

There has been extensive prior work on methods for link prediction. From class, we have learned that TransE uses a margin-based pairwise ranking loss function to ensure that for a head, relation, tail triple (h, r, t), $h + r \approx t$. ProjE [5], on the other hand, optimizes a ranking loss function of the list of candidate entities to complete an edge. ProjE has the advantage that it a relatively small number of parameters, slightly more than the number of parameters of TransE, and it does not require prior training unlike other prior models discussed in the paper. This approach represents

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

a viable knowledge graph approach to apply. We elected to use the ConvE based approach which similarly ranks candidate entities while also applying a deeper network. ConvE's use of representing 2D embeddings of entities and relations possibly offered the potential of being able to entities and relations better. If ConvE is not a successful knowledge graph approach, ProjE would be a direction for us to go towards.

Another method for link prediction is doing personal recommendation on bipartite networks projections [8]. The paper describes using bipartite graphs that assign weights to help extract hidden information from networks that could be used for personal recommendation, which the paper claims is an improvement to collaborative filtering. In our project specifically, we would like to try using this method to output recommendations for the current affiliation for a scientist from the OAG.

The idea for [8] is that they propose a weighting method which can be directly applied in extracting hidden information of networks, a method to compress bipartite networks and offers a solution for personal recommendation. Their recommendation algorithm starts with an unweighted bipartite network, where the resource (recommendation power) for each node in one set is equally distributed to its neighbors in the other set and vice versa. The resource allocation process is split into two steps, first from the first set (let's call this set A) to the other set (set B) and then back to A. Using this weighting method, the recommendation algorithm first compresses the bipartite user-object network by object-projection, resulting in a weighted network. Then, for a given user of this weighted network, put some resource on those objects that had already been collected by the user. In other words, if an object has been collected by a given user, its initial resource is a unit(what the unit represents is defined by the problem), zero otherwise. Then, for any user, all his/her uncollected objects are sorted in descending order by the resource allocated for it, and the objects with the highest value of final resources are recommended, a network-based inference.

In the future, we are interested in trying out this approach of weighting the possible affiliations of a scientist through the resource-allocation method mentioned, and then using the weighted network to recommend which affiliations are most likely/suitable for the scholar. Specifically, we would map the scholars to be users, and affiliation (e.g. university or institution) to be the objects. However, this approach is relatively more complex when we are considering the efficacy of utilizing graph structure for OAG data link prediction.

One more implementation for link prediction that was considered is supervised link prediction in bipartite networks [1]. This paper describes using a variety of link prediction techniques and investigated how they could be applied to bipartite graphs. The authors explain that modified versions of similarity/distance metrics like Jaccard's coefficient and Adamic Adar can be effective for bipartite network link prediction if the metrics were modified to be applied over a larger set of close nodes than just neighbors.

Various random walk methods as well as matrix approximations have been tested as well, and a supervised classifier was found to be the best method at predicting overall what new edges will be in the network in the future [1]. However, the authors note that a supervised model has the disadvantage of being a very complicated model, relying on the results of other unsupervised techniques. They note that a supervised random walk performed equally as well as supervised classifier, and that supervised random walks were able to provide scores for every candidate edge in a reasonable amount of time. One weakness for a supervised random walk is that it is slower to train. We are interested in using a simpler random walk in our approaches.

The above bipartite graph approaches represent viable and potentially complex options compared to what we seek to investigate. We seek to understand the efficacy of utilizing the OAG data's graph structure to predict the current author affiliation. Therefore, we are interested in using the random walk algorithm to try and predict new edges in the graph. So, given a bipartite graph of authors and affiliations as the two sets nodes, we would like to use the random walk methods to try to predict the most likely affiliation for an author.

3 Dataset

The data we will be using will be the Open Academic Graph (OAG), a knowledge graph that combines two academic graphs: Microsoft Academic Graph (MAG) [6] and AMiner [7]. The OAG contains papers from both academic graphs and links relations between MAG and AMiner. The OAG provides

three different sets of data: academic venues, academic papers, and authors of the papers. Because of the large amount of data available in these three datasets (authors, venues, and papers), we sought to explore how accurate of results we could achieve by only looking at the authors portion of this dataset. As a future work we could incorporate more data from the venues and papers datasets to evaluate how doing so improves our author affiliation accuracy. We are accessing the data through using the BigQuery tables available at [4].

3.1 Dataset Features and Pre-processing Details

For each author, we collected the following information: id, current organization, previous organizations, which could potentially include the current organization, and research interests. We additionally filtered out entries with missing 'org' fields so that we could properly evaluate our accuracy when predicting an entry's 'org' field. Our baseline feature vector approach and ConvE approach use all of these fields while our bipartite model (the unsupervised random walk) only uses the id and current organization fields. We found that processing too many (about 1 million) entries led to the ConvE infrastructure and our feature vector baseline crashing. We therefore conservatively chose to process 100,000 author entries in total.

We randomly sorted the data and placed 80% in the training data, and the other 20% was used for test and validation data. For the validation/test data, we removed any entries in which the org field was unseen in the training data because it would not be possible to correctly predict these entries. Doing so led to removing about 600 entries (or about 3% of the training data). For the unsupervised random walk [1] approach only, no entries were removed. More details on that choice are described in the approach section.

4 Models/Algorithms/Methods

Based on this OAG dataset, it is not initially clear to us what type of model will allow us to have the best accuracy in predicting an author's current organization. Because of the many features for an author, it may be that the features alone are sufficient to predict the current organization. Because the data can be represented as a bipartite graph, primarily utilizing the graph structure through an unsupervised random walk may be sufficient for promising results. Otherwise, a knowledge graph approach which combines the features of an author and the structure of the graph may be what is optimal. We describe all three approaches in this section.

4.1 Baseline: Feature Vector

In order to test the predictive power of the data without incorporating any graphical structure, we implemented an approach in which we represented each author's data as a feature vector and trained a Stochastic Gradient Descent classifier to predict an author's current organization. We will describe how we formed the feature vector.

In this approach, using the gensim package, we train two word2vec [3] models: one for all research interest words in the data and one for all organizations in the data. For each author, we form a feature vector that is the concatenated embeddings of a subset of the author's research interests and a subset of the author's previous organizations.

We note that the number of research interests and previous organizations listed varied per author. Therefore, in order to ensure that each author's feature vector was of the same length, we randomly sampled a subset of research interests and embedded those interests. We similarly sampled a subset of previous organizations and embedded those organizations. If an author had fewer previous organizations or interests than the number sampled, we concatenated an embedding of all zeros in order to keep each feature the same length. For the previous organizations, we randomly selected 2 because based on random samples, we saw that each author had an average of fewer than 2 previous organizations. For research interests, we randomly selected 26 interests for each author, the number of average interests we found.

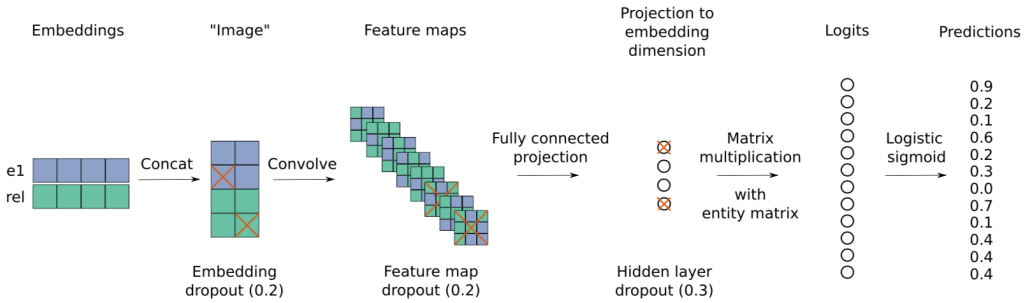

Figure 1: Overview Architecture of ConvE approach, originally from [2].

4.2 Unsupervised Random Walk

Unsupervised random walk [1] was implemented to predict links between authors and affiliations. Link prediction can be viewed as a binary classification task on whether there will be an edge between two nodes at some future time [1]. Thus, to evaluate unsupervised random walk, we removed some of the existing/known edges between authors and affiliations and kept track of them as ground truth, then attempted to predict which pairs of nodes will have an edge between them. To do so, our train and test set are split into two lists of edges, graph.txt and new_edges.txt. The graph.txt is a list of edges that we keep as known, and the new_edges.txt is a list of edges that were originally known, but we remove them from the graph to make them unknown.

We can view a random walk as a finite Markov chain that moves through a graph $G = (V, E)$ [1]. Let each edge $(u, v) \in E$ for nodes $u, v \in V$, be assigned a weight $w(u, v)$. Each edge is assigned a transition probability $M_{u,v} = \frac{w(u,v)}{d(u)}$ where $d(u) = \sum_{(u,v)\in E} w(u, v)$ is the weighted degree of u. However, for our specific problem with the OAG graph, we chose to use unweighted edges, so all $w(u, v) = 1$, since the original paper[1] chose to weight edges based on which edges were created more recently. However, there is no time information given by the OAG graph, so we were unable to use the same weights and instead chose to have unweighted edges. The transition probabilities are used in the random walk through the graph. From node u, we would move to node v with probability $M_{u,v}$. To use this random traversal of the graph to determine which nodes would be most often visited and predict an edge, we see that the probabilities $p_t = (p_t(0), p_t(1), \ldots, p_t(n))$ of ending a random walk at a particular node after t steps is given by $p_t = p_{t-1}M$. This will converge to a stationary distribution p_s as $t \to \infty$, where $p_s = p_s M$ [1]. The unsupervised random walk we use is a personalized random walk from a starting node u. Thus, in addition to traversing through edges, there is a probability β of jumping back to the starting node u during a walk. Thus, the probability distribution for personalized random walk is $p_t = (1 - \beta)p_{t-1}M + \beta e_u$ where e_u is a one-hot vector with 1 at index u and zeroes for all other entries.

Random walks were run to produce a score for each candidate edge (u, v) for nodes $u, v \in V$. We score the edge with stationary distribution probability of reaching v when starting the random walk from u, with no weights ($w(u, v) = 1$ for all edges). Then, of all edges reached by the random walk starting from node u, we we predict a link with the node v that had the highest scoring candidate edge. Since our graph is a bipartite graph, the adjacency matrix will be highly sparse, since there are no edges between nodes of the same set (i.e. no edges between authors and no edges between affiliations). Due to a sparse adjacency matrix, we could exploit that structure and run random walks fairly effectively without taking up too much time and memory. If the adjacency matrix were not sparse due to the structure of a bipartite network, then we would need to consider a much larger number of candidate edges, which would make running random walk on a very large and dense graph, infeasible.

4.3 ConvE

ConvE [2] is an approach for knowledge graph link prediction. More specifically, this knowledge graph is defined as a set of (s, r, o) triples, where s is the subject entity, r is the relation, and o is

Table 1: Results from Feature Vector, Unsupervised Random Walk, and ConvE approaches

Approach Name	Number of Train Examples	Number of Test Examples	Test Accuracy (%)
Feature Vector	$40,000$	$9,491$	69.6%
Random Walk	80,000 [1]	20,000 [2]	1.26%
ConvE	$80,000$	$9,666$	91.6%

the object entity. One of the main features of ConvE is that it embeds entities and relations as 2D embeddings and applies convolutions on these embeddings. We now describe this approach in detail.

In the forward direction, the model starts with a subject entity s and a relation entity r. It then looks up the embeddings of these entities in their respective embedding matrices to have $\mathbf{e}_s$ and $\mathbf{e}_r$. The embedding matrices are $\mathbf{E} \in \mathbb{R}^{|\epsilon| \times k}$ and $\mathbf{R} \in \mathbb{R}^{|R| \times k'}$, where ϵ is the set of all entities, R is the set of all relations, k is the entity embedding size, and k' is the relation embedding size. The model then reshapes $\mathbf{e}_s$ and $\mathbf{e}_r$ to 2D embeddings and then concatenates these embeddings. The model then applies convolution filters to the concatenated embeddings.The output of this convolution operation is a tensor $T \in \mathbb{R}^{c \times m \times n}$, where c is the number of feature maps, and m and n are the dimensions of the feature maps. T is then reshaped into a vector $v_T \in \mathbb{R}^{cmn}$. v_T is then projected into a k-dimensional space via a linear transformation by matrix $\mathbf{W} \in \mathbb{R}^{cmn \times k}$. v_T is matrix multiplied with $\mathbf{E}$. The result of this matrix multiplication is passed to a logistic sigmoid function, yielding the probabilities of possible object entities. $\mathbf{e}_o$ corresponds to the object entity o with the highest prediction probability. Figure 1 from [2] provides a visual of the architecture and outlines the dropout that occurs for the concatenated embedding, T, and v_T.

5 Results and Findings

Table 1 outlines the results from out three approaches. The main evaluation method we used for our link prediction task is accuracy. We define this accuracy as given an author and potentially other information such as his/her research interests and previous organizations, the accuracy is the number of correctly predicted current organizations divided by the total number of entries considered. In the ConvE model, this is considered the "left accuracy", which predicts the organization given the author and relation, as opposed to predicting the author given the organization and relation. Below, we describe each experiment and the results of each one.

5.1 Baseline: Feature Vector

We initially aimed to run the baseline model on all 100,000 examples. After running the model for over 20 hours, the model did not complete training. We therefore ran the baseline on $50,000$ examples. The training accuracy was 68.5%. We note that the train accuracy is slightly lower than the test accuracy in Table 1. We have seen the model increase performance with more data; for example, running with $16,000$ training examples yielded a a training accuracy of 66.7% test accuracy of 48.6% on $3,555$ test examples. Therefore, a future step would be to run the baseline model on all 100,000 examples to completion. We could also try overfitting to the training data since the current training accuracy is relatively low. Overall, we find that the baseline model has $5,498$ unique organizations, so the current test accuracy is reasonable and well above chance.

5.2 Unsupervised Random Walk

To evaluate how well the unsupervised random walk performed in predicting links between authors and affiliations, we use the new_edges.txt as ground truth to calculate accuracy. The unsupervised random walk runs started on the author nodes of the new_edges.txt and walks for $t = 10$ steps before outputting the scores of each candidate edge with the starting node u. The teleport/jumping probability was set to be $\beta = 0.2$ by default. We took the maximum of the candidate edge scores and returned the edge with the highest score as our predicted link.

[1] Out of the 80,000 train examples, graph.txt had 64733 and new_edges.txt had 15267

[2] Out of 20,000 examples, graph.txt had 16486 and new_edges.txt has 3514

The train and test sets are shuffled randomly with each run, but the proportion is around the same, about 20% of the examples of each set are set off to be new_edges.txt, with these edges removed from the graph. The accuracy calculation is accuracy $= \frac{\text{Number of correct link predictions compared against ground truth}}{\text{Total number of predicted examples}}$, where the ground truth are the edges listed in new_edges.txt and the numbetend examples is the number of author nodes in new_edges.txt.

From ten runs of unsupervised random walk, the accuracy of link predictions fluctuated around 1.25%, with the average of accuracy being 1.16%. Analyzing these results, one observation to note is that when creating the examples, the negative examples outnumber the positive examples by a magnitude of 10: there were approximately 15,144 positive examples and 152,037 negative examples. This is due to the sparsity of the graph leading to a high class imbalance, because while the affiliation nodes can have any degree, the degrees of the author nodes are at most 1, since one author will only have one current affiliation, but an affiliation could have multiple authors linked to it. One way to deal with this issue would be to construct smaller and more balanced sets of examples of evaluate the unsupervised random walk on.

Another issue to note is that our data with the OAG has a bit of a mismatch with the unsupervised random walk algorithm. As noted in the approach, the unsupervised random walk method performs link prediction as a binary classification task on whether or not there will be an edge between two nodes at some future time [1]. However, the OAG data does not inherently have any notion of time: past, present, or future. In the original paper [1] that implemented the unsupervised random walk, they were able to weight the edges of their network by giving a higher weight to edges that were created more recently to encode information about how the graph changes over time. Since OAG does not provide any sort of data to be able to track how the authors and affiliations evolve over time (there is a field in OAG that lists all affiliations for an author, but does not order them chronologically so we are unable to track the migration through time), we could not use any heuristics to provide weights for the edges for OAG and as a result, the random walk truly was random, traversing unweighted edges to try to predict links for authors.

5.3 ConvE

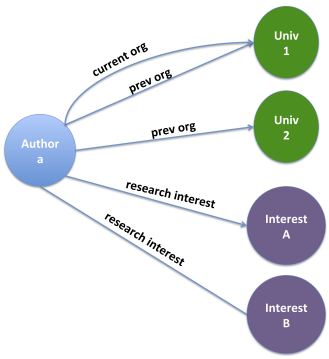

Figure 2: Example of an author node with current and prev org edges to example universities and research interest edges to example interests.

Figure 2 depicts how we modeled the OAG data as a knowledge graph to apply the ConvE approach. We see that an author node has three type of outgoing edges: “current org” (corresponding to the ‘org’ field), which is directed to a university node; “prev org" (corresponding to entries in the array of previous orgs), which is directed to a University node, abbreviated “Univ”; and “research interest”, which is directed to an interest node. We note that each author only has outgoing edges, and each interest and university node only has incoming edges. Because a university node can have an incoming “current org” and “prev org” edge, we note that this knowledge graph is a multi-graph. In total, our knowledge graph contains about $100,000$ author nodes, about $9,000$ organization nodes, and about $30,000$ research interest entities.

Table 2: OAG Knowledge Graph In-Degrees

Node Type	Relation Type	Average In-Degree	Maximum In-Degree
Organization	Current Organization	12.6	1,144
Organization	Previous Organization	14.8	1,773
Interest	Research Interest	87.7	13,382

Because the our task is to predict a link from an author entity to an organization entity, we gathered all (author, "current organization", organization) triples, and we split these triples into 80% train, 10% validation, and 10% test splits. In addition to these current organization triples, the train set also included all of the (author, "previous organization", organization) and (author, "research interest", interest) triples. Therefore, the validation and test sets only contained current organization triples.

We ran the model for about 300 epochs rather than running the model longer because we saw that increasing the number of epochs did not affect the loss or accuracy significantly on earlier runs of the model. After 296 epochs, our test accuracy, the ConvE "left" accuracy, is 91.6% with a validation accuracy of 91.4%. Running the model until 300 epochs yields a training loss of about 0.0001. As a reminder, ConvE gives an object prediction probability for all entities given a subject entity and a relation. If we consider the accuracy of either of the top-2 highest predictions being the correct object entity, also knows as Hits @2, then ConvE yields an Hits @2 of 98.7% for the test set.

We find the accuracy of applying this model to the OAG data to be a promising approach. We note that one characteristic of our dataset is that we noticed that some entries had only one previous organization, which was also the current organization since previous organizations may also include current organizations. This means that only predicting the single previous organizations would yield a decent accuracy. Through randomly sampling $20,000$ entries, we find that 78.4% of the entries have a single previous organization that is also the current organization. We note that our ConvE test accuracy is higher than this percentage while the baseline model is below this percentage. This indicates that ConvE is going beyond simply identifying these entries where the single previous organization is the same as the current organization.

Compared with the feature vector baseline, the ConvE approach is capable of achieving a much higher accuracy by incorporating all previous organizations and all research interests along with considering the graphical structure of the data, possibly contributing to its higher test accuracy. It is also possible that ConvE's accuracy benefits mainly from being able to process more training data than the baseline. However, we note that the ConvE approach has about $9,000$ unique organizations while the baseline has $5,498$ unique organizations. Therefore, it is not the case that ConvE processing more training data only contributed to having more data that supported representing the organizations of the baseline model better. It is therefore viable to consider that ConvE's higher accuracy is not only from processing more training data.

The authors in [2] found that ConvE performs well on knowledge graphs where there are nodes with high in-degrees of the same relation. From Table 2, we find that our knowledge graph contains such nodes for all three relation types, which may contribute significantly to ConvE's test accuracy. This ability to model nodes of high same-relation in-degree is important in modeling each author node because each author node is complex. On average each author node has 29.2 outgoing edges to organization and interest nodes. Having a deeper model that can represent a complex author node and the nodes it is directed to may be a key factor in ConvE's high accuracy with OAG data.

An interesting future study for understanding ConvE's efficacy with OAG data would be to to determine if having high in-degree nodes for the "Current Organization" relationship was sufficient to achieving strong results, or if having high in-degree nodes for "Previous Organization" and/or "Research Interest" relation(s) was also needed. If having high in-degree nodes for the "Current Organization" was sufficient for high accuracy, it may be that ConvE only needs to represent high same-relation in-degree nodes for relations it is predicting since it is not predicting the "Previous Organization" and "Research Interest" relations for our task. Otherwise, having high in-degree nodes for all three relations may be an important factor for ConvE's success with ConvE data.

6 Conclusion and Future Work

Our work has analyzed three approaches in making progress towards having a more robust OAG dataset. The feature vector baseline predicting the current organization based solely on the previous organization and interest information. The unsupervised random walk allowed considering the graph structure. The ConvE approach represents being able to combine both the feature data and the graph structure and yielded the best result. With more time, would aim to modify the architecture to make it a deeper network. While the ConvE approach Hits @2 score is very high, we would ideally want the accuracy to be above 98%. Being able to develop a deeper network may allow for better representation of these complex graph structures. As the authors in [2], while ConvE is a relatively deep model when it comes to other knowledge graph approaches, there are many successful deeper convolutional network models in computer vision. Knowledge graphs may benefit from borrowing deep architectures from computer vision.

Other than modifying the architecture, as mentioned in the dataset section, we are only using the authors dataset from the entire OAG data. Incorporating the data in the papers dataset and the venues of these papers would allow us to build a more complex knowledge graph and potentially increase the accuracy of predicting an author's current organization. Understanding the effects of adding the papers and venues datasets would allow us to examine whether or not the authors dataset is sufficient to have strong current organization prediction results for the OAG data. Overall, we are pleased to demonstrate that representing the OAG data as a knowledge graph with a convolutional architecture is a promising outlook for this dataset.

7 Contributions

7.1 Jacqueline Yau

Jacqueline worked on the baseline, especially with using BigQuery to gather the data. She also worked on the unsupervised random walk approach, including structuring the open repository to fit the nature of the OAG data.

7.2 Timothy Le

Timothy worked on implementing the feature vector baseline and the ConvE model. He also developed a multi-graph representation of the knowledge graph in order to analyze its properties.

References

[1] Kameshewar Chinta, Kevin Clark, and Arathi Mani. Supervised link prediction in bipartite networks. CS224W Final Project, 2014.

[2] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[4] Daniel Perez Rada. bigquery-oag. `https://github.com/ESHackathon/bigquery-oag`.

[5] Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[6] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246. ACM, 2015.

[7] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, KDD '08, pages 990–998, New York, NY, USA, 2008. ACM.

[8] Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical review E*, 76(4):046115, 2007.