Effective Mentor Suggestion System for Collaborative Learning

Advait Raut[1] *Upasana G*[2] *Ramakrishna Bairi*[3] *Ganesh Ramakrishnan*[2]

(1) IBM, Bangalore, India, 560045

(2) IITB, Mumbai, India, 400076

(3) IITB-Monash Research Academy, Mumbai, India, 400076

advait.m.raut@gmail.com, upasana@cse.iitb.ac.in, bairi@cse.iitb.ac.in, ganesh@cse.iitb.ac.in

Abstract

Accelerated growth of the World Wide Web has resulted in the evolution of many online collaboration platforms in various domains, including education domain. These platforms, apart from bringing interested stakeholders together, also provide innovative and value added services such as smart search, notifications, suggestions, etc. Techpedia is one of such a platform which facilitates students to submit any original project description and aims to nurture the project idea by mentoring, collaborating and recognizing significant contributions by the system of awards and entrepreneurship. An important aspect of this platform is its ability to suggest a suitable mentor to a student's project. We propose an elegant approach to find an appropriate mentor for a given project by analyzing the project abstract. By analyzing past projects guided by various mentors and by engaging Wikipedia knowledge structure during the analysis, we show that our method suggests mentor(s) to a new project with good accuracy.

KEYWORDS: mentor suggestion, collaboration, information extraction, wikipedia.

Proceedings of the Workshop on Speech and Language Processing Tools in Education, pages 1–8,
COLING 2012, Mumbai, December 2012.

1 Introduction

The internet has become more and more popular medium for online collaboration from past several years. Research suggests that collaborative learning has the potential to foster interaction and social support lacking in traditional learning environments [3]. Now a days online collaboration has pervaded into almost every field. There are varieties of platforms where people can collaborate and share their ideas, resources, techniques and work towards a common goal. The success of online collaboration has attracted more and more people/organizations to participate and grow the network. As the size of collaboration network increases, it poses many challenges in terms of resource management, organization, timely access to right piece of information, to name a few. This calls for sophisticated techniques and algorithms for better management and utilization of resources in a collaboration. In this paper we propose a solution to one of such problem (more precisely mentor identification problem, discussed later) in an academic domain. More specifically, we conducted our experiments on a well known collaboration platform called Techpedia [1]. However, we strongly believe that our techniques can be well adopted for any other academic collaboration platforms which support basic characteristics of an academic project life cycle management.

Techpedia is one of well known online collaborative platform for technology projects by students to link the needs of industry and grassroots innovators with young minds and to promote collaborative research. It facilitates students and innovators to share their ideas/project and collaborate with mentors/guides, industries and sponsors. Every project will have one or more mentors who are competent in the project field and guide the project. As the number of projects and mentor network increases in size, it becomes very challenging for a person to identify and approach a mentor for his/her project manually. Hence it is important to have robust algorithms to automatically detect suitable mentors for a given project abstract.

Apart from providing collaboration platform, Techpedia also acts as a huge repository of past projects details such as abstracts, techniques, applications of projects, development details, mentor details, project related communication details, and many more. It also provides details about mentors such as their area of expertise, skill set, experience etc. By utilizing all these information, we propose an information extraction technique for mentor identification for a new project by analyzing its abstract.

Since project abstracts are generally very concise, we observed that augmenting the abstracts with related Wikipedia entities before running the extraction process produces a better result. In section 3 we empirically show that, this technique significantly improves our mentor suggestion accuracy.

Rest of the paper is organized as follows. In section 2 we state our problem more formally and provide solutions for mentor suggestion, in section 3 we compare the techniques that we proposed, in section 4 we provide the details of prior work and we end the paper with conclusion in section 5.

2 Mentor suggestion

2.1 Problem Definition

Given a set of past projects $\mathbb{P} = \left\{p_1, p_2, ...p_{|\mathbb{P}|}\right\}$, a pool of mentors $\mathbb{M} = \left\{m_1, m_2, ...m_{|\mathbb{M}|}\right\}$, and a set of catalogs (like Wikipedia, Wikipedia Miner, etc) $\mathbb{W} = \left\{W_1, W_2, ...W_{|\mathbb{W}|}\right\}$, we would like to determine a subset $M \subseteq \mathbb{M}$ of mentors who can best guide a new project p.

Each project p_i has a set of attributes like owner, title, abstract, description, mentors who guided the project, inception date, duration, students who worked on them, sponsors, etc. We refer these attributes using dot notation as shown below: $p_i.title =< project\ title >$; $p_i.abstract =< project\ abstract >$; $p_i.mentors = \{set\ of\ mentors\ guided\ project\ p_i\}$

Similarly, each mentor has attributes like name, id, organization, work experience, a brief technical profile explaining his/her skill area and technical competence, etc. Again, as in case of projects, we represent these attributes using dot notation.

Formally, we define the problem as learning a function θ that suggests the mentors for the project p as follows:

$$\theta = \{M \mid p, \mathbb{P}, \mathbb{M}, \mathbb{W}\}$$

2.2 The Algorithm

We addressed our problem using two different approaches and compared the results. In sections 2.2.1, 2.2.2 we describe our approaches, which we compare and evaluate in section 3.

2.2.1 VSM over Past Project

In this approach, we tokenized the abstracts of each project in $\mathbb{P} = \{p_1, p_2, ...p_{|\mathbb{P}|}\}$, and represented each project in a Vector Space Model (VSM), after removing stop words and stemming. Using jaccard and cosine similarity measures we found out similarity between the target project p and the past projects $\mathbb{P}$. The mentors of the best matched past project are suggested as the mentors for the new project p. The Algorithm 1 outlines this approach.

Algorithm 1 VSM over Past Project

input: New project p, Past projects $\mathbb{P}$, Mentor pool $\mathbb{M}$
output: Suggested mentors $M \subseteq \mathbb{M}$
Initialize $M = \{\emptyset\}$, $p_{best} = \emptyset$, $V_{best} = \emptyset$
V_p =Vector Space Model of $p.abstract$
▷ Find best matching project
for $i = 1$ to $|\mathbb{P}|$ **do**
    V_i =Vector Space Model of p_i
    if $CosineSimilarity\left(V_p, V_i\right) > CosineSimilarity\left(V_p, V_{best}\right)$ **then**
        $p_{best} = p_i$
        $V_{best} = V_i$
    end if
end for
$M = p_{best}.mentors$ /Comment Mentors of p_{best}
return M

2.2.2 VSM over Combination of Wikipedia Entities and Project Abstracts

In general, the project abstracts are short and concise. Due to this, the vector representation of these abstracts in VSM becomes highly sparse. This leads to poor jaccard and cosine similarities in approaches presented in section 2.2.1. To alleviate this problem, we propose a technique called Wikification. The following are the steps of Wikification:

1. Entity Spotting: We spot Wikipedia entities in every project abstract using Wikipedia Miner.
2. Entity Disambiguation: In some cases, multiple Wikipedia entities can match for a spotted word or phrase. In such cases, we disambiguate the entities based on the context available in the project abstract.
3. Semantic Expansion: For each of the identified entities in previous two steps, we collect semantically related entities by exploiting Wikipedia structure. Wikipedia maintains various kinds relations between the entities through hyperlinks, categories, see also links, redirect pages, disambiguation pages, etc. We found three types of semantic relations suitable for our work. Table 1 explains these relations. As part of offline processing, these relations were extracted for every Wikipedia entity, indexed and stored in a repository. We made use of this repository for extracting required semantic relations.

Semantic relations	**Semantic values from Wikipedia page excerpts**
Synonym : Is instrumental in identifying entities which are known with different names.	All redirected names of the Wikipedia page and the values of the Info box attributes like 'Nick Name', 'Other Names' are stored under this class.Ex: For *Sony: Sony Corp, Sony Entertainment*
Association : An association signifies the connection between two query terms. It can be unidirectional where one entity includes other within its description. It can also be bidirectional i.e entities use each other in their description. Ex:For *Sony:Sony Ericsson, Sony Products*	All valid hyperlinks of a Wikipedia page.
Sibling : The entities which have one or more common parents	Siblings are the sub categories/ pages which do not follow hyponym pattern. Ex: For *Sony: list of sony trademarks*

Table 1: Semantic Relations extracted from Wikipedia

At the end of Wikification process, we get a set of Wikipedia entities that are related to the project abstracts. We used entity descriptions along with project abstract text to build a VSM. Using jaccard and cosine similarity measures we found out the similarity between the target project p and the expanded abstracts of past projects $\mathbb{P}$. The mentors of the best matched past project are suggested as the mentors for the new project p. The Algorithm 2 outlines this approach.

3 Experiments and Evaluation

3.1 Experimental Setup

Experiments and Evaluation was done on Techpedia corpus. We identified 600 projects and 64 associated mentors as our training and testing data. We manually inspected and corrected some of the projects where mentor assignment was not proper and some mentor profiles where data was missing.

Algorithm 2 VSM over Combination of Wikipedia Entities and Project Abstracts

input: New project p, Past projects $\mathbb{P}$, Mentor pool $\mathbb{M}$, Catalogs $\mathbb{W}$
output: Suggested mentors $M \subseteq \mathbb{M}$
Initialize $M = \{\varnothing\}$, $p_{best} = \varnothing$, $V_{best} = \varnothing$
▷ Find best matching project
for $i = 1$ to $|\mathbb{P}|$ **do**
    V_i =Vector Space Model of p_i
    ▷ Spot Wikipedia entities using Wikipedia Miner
    $S = \{Entities\ spotted\ in\ p_i.abstract\ by\ Wikipedia\ Miner\}$
    $E = \{\varnothing\}$
    for each entity $e \in S$ **do**
        $E = E \cup \{Semantically\ related\ entities\ to\ e\}$
    end for
    $p_i.expandedAbstract = p_i.abstract$
    for each entity $e \in E$ **do**
        $p_i.expandedAbstract = p_i.expandedAbstract \cup e.text$
    end for
    V_p =Vector Space Model of $p.expandedAbstract$
    if $CosineSimilarity\left(V_p, V_i\right) > CosineSimilarity\left(V_p, V_{best}\right)$ **then**
        $p_{best} = p_i$
        $V_{best} = V_i$
    end if
end for
$M = p_{best}.mentors$ ▷ Mentors of p_{best}
return M

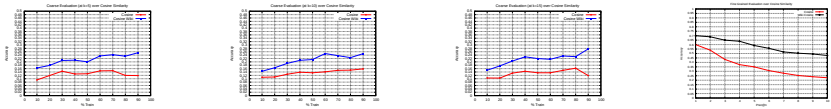
We used Wikipedia Miner APIs to query and extract spotting details from Wikipedia Miner portal services. We also used offline Wikipedia dump (2011) and extracted semantic relations (detailed in Table 1) of all Wikipedia entities offline, indexed and stored them them using Lucene for quicker access and processing.

3.2 Evaluation Methodology

We adopted 2 fold evaluation methodology where we split our data (600 projects) into training and testing sets. We loosely use the word training here to refer the set of past projects from which we learn the mentor-project relationship. We treated the projects in testing set as new projects by masking the mentors assigned to those projects. We then evaluate the accuracy of our system by comparing the suggested mentors to the masked mentors. We propose two schemes of evaluation: 1. Coarse Grained Evaluation, 2. Fine Grained Evaluation.

3.2.1 Coarse Grained Evaluation

For each project in Test set, ranked mentor list was found based on the similarity score of matching test project abstract with the past project abstracts from the training set. For evaluation, we considered mentor is correctly identified if the actual mentor (which was masked before running the test) is present in top-k mentors of the ranked mentor list. We evaluated our system for k=5, 10 and 15. Figures 1a, 1b, 1c show the result of coarse grained evaluation.



(a) Precision @ 5 (b) Precision @10 (c) Precision @15 (d) Fine grained

Figure 1: Evaluation Results

3.2.2 Fine Grained Evaluation

Note that, there can be multiple mentors for the same field of work. It is acceptable to have any one of those mentors as the suggested mentor by the system. But our previous scheme of evaluation (section 3.2.1) does not consider this, resulting in poor accuracy. Hence we devised a new evaluation scheme (we call it Fine Grained Evaluation) wherein, we manually inspect if the system suggested mentors for the projects in test set fall under the same area of expertise as demanded by the projects. Since it was tedious to manually evaluate 600 projects, we randomly selected 100 projects which we further split into 60:40 as train and test sets. As in the previous scheme, here again we obtained a ranked list of mentors and evaluated precision at top-k. Figure 1d shows the result of fine grained evaluation.

3.3 Observations

3.3.1 Coarse Grained Evaluation

We observe that using Wikipedia Miner along with semantic expansion (Algorithm 2) has yielded better accuracies than rest of the methods(Algorithms 1).

3.3.2 Fine grained evaluation

Even with this scheme, we observe that using Wikipedia Miner along with semantic expansion (Algorithm 2) has yielded better accuracies than rest of the methods (Algorithms 1).

4 Prior Work

The work [4]describes two stage model for finding experts relevant to a user query: relevance and co-occurrence. Co-occurrence model learns co-occurrence between terms and the author of that document. [5] social search model aims to rank expert answerer to a user query by assigning topics to query using Latent Dirichlet Allocation [2]. Wikipedia Miner [6, 7] aims to find and link Wikipedia entities within a document with entity disambiguation.

5 Conclusion

We presented a body of techniques to suggest suitable mentors for a new submitted project in an online collaborative platform like Techpedia. Our work was hinged around the need of selecting a mentor for a project in a large corpus of projects with ease and accuracy. We demonstrated with multiple different techniques how this can be achieved. We also showed how an external catalog like Wikipedia can be engaged to enhance the accuracy of suggestion and empirically proved that semantic expansion yields better results. As part of our future work, we aim to improve the accuracy of suggestion by considering mentor's technical profile more rigorously and by adopting semantic analysis of project details using NLP techniques.

References

[1] http://www.techpedia.in.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[3] C. J. Bonk and K. S. King. Electronic collaborators: Learner-centered technologies for literacy, apprenticeship, and discourse. March 1998.

[4] [Y Cao et al. A two-stage model for expert search. MSR-TR-2008., 2008.

[5] Damon Horowitz and Sepandar D. Kamvar. The anatomy of a large-scale social search engine. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 431–440, New York, NY, USA, 2010. ACM.

[6] Olena Medelyan, Ian H. Witten, and David Milne. Topic indexing with wikipedia.

[7] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM.