Automated Generation of Interview Panels

Sujit Kumar Chakrabarti, IIITB

Abstract

We present a method for automatically creating interview panels for research interviews at IIITB.

1 Introduction

In IIIT-B there are seven research domains:

1. Computer Science (CS)
2. Data Science (DS)
3. Networking, Communication and Signal Processing (NCS)
4. Software Engineering (SE)
5. Electronics System Design (ESD)
6. Information Technology and Society (ITS)
7. Mathematics and Basic Sciences (MBS)

The selection process of students in the research programmes involves inviting applications from candidates, shortlisting through application review, written test and interview and final selection. Most of the process is online and is carried out using an indigenously developed web application called Pravesh developed by Prof. Srinivasaraghavan and students.

The applicants have to choose the primary research domain under which they wish to apply. For each (research) domain, there is a *domain representative* (a faculty member listed under the domain) who does an initial screening of all the applications in his/her domain. One of the objectives is to identify desk rejects: applications which are so weak as not to merit further scrutiny. However, the main reason for this step is to refer the applications in his/her domain to appropriate faculty members in that domain based on matching research interest. Each application must be reviewed by at least 3 reviewers. A candidate can get short listed for final interview only if there is at least one

faculty member who has reviewed his/her application and has shown his/her interest in supervising the candidate should he/she be finally selected.

We see the following disadvantages of this process:

1. **Labour Intensive.** Domain representative's initial task is very labour intensive. He/she has to go through all the applications in his/her domain – which may run into several hundred – to assign reviewers. This must be done really quickly. Therefore, there are chances of error (applications going to irrelevant reviewers, and/or missing relevant reviewers).
2. **Exclusive Domains.** The applicant must choose a single primary domain. Though there is a facility to choose a secondary domain, this is mostly a choice made to the exclusion of all other choices. This is very restrictive. Quite often, the alignment of a candidate's research interests is not well aligned within the boundaries of a domain. There are many candidates whose interests cut across the boundaries of predesignated research domains. In such cases, an application may miss being noticed by a faculty member with matching research interest because of domains not matching. For example, an applicant may be interested in working on applying formal methods for verifying machine learning algorithms. Which domain should he/she apply to – DS, CS or SE?

We believe that we should not force applicants to commit exclusively to one domain. Instead, they should be asked to mention their interests as a set of keywords. The review panel for the application and the panel of interviewers in case the application is short listed should then be composed based on keyword matching between the interests of the candidate and those of pertinent faculty members.

In this article, we present the details of a preliminary experiment we did to test this idea.

2 Basic Idea

The basic idea of our approach is simply to match applications with faculty members based on the keywords. Simply put, suppose that the candidate C's application has 3 matching topics with professor P_1 and 2 with professor P_2. All other things remaining the same, the application should be reviewed by P_1 and not P_2 if only one of them has to be chosen as the reviewer. This is the basic idea of our approach.

When applying this idea on a real data set, we encounter a number of other challenges:

1. **Common keywords.** There are a number of keywords which appear in a large number of applications, e.g. Machine Learning, Artificial Intelligence, etc. Applicants mention these not so much because they are interested in these topics, but because these are trending topics of the day.
2. **Overloaded faculty members.** For a similar reasons, there are faculty members who would appear in an inordinately large number of review panels, simply because they too are interested in such trending topics as machine learning.

Therefore, the automated reviewer assignment process must associate less importance to common keywords as compared to rarer keywords. Also, it must have a mechanism to avoid adding faculty members to review panels where they have a match through common keywords.

To formalise the discussion, we define the following:

1. Faculty set F.
2. Topic set T.
3. Application set A.
4. Faculty topic set $FT \ : \ F \mapsto 2^{\{T\}}$.
5. Application topic set $AT \ : \ A \mapsto 2^{\{T\}}$.
6. **Match triples.** For faculty interest and applicant interest, we construct match triples ($F \times T \times A$) which are generated by merging the information available in FT and AT.
7. **Faculty score S_F.** The number of match triples a faculty member (a.k.a professor) appears in.
8. **Topic score** S_T. The number of match triples a topic appears in.

Topic score is indicative of how popular a particular topic is among the applicants. Faculty score is indicative of a faculty's involvement in popular topics. Topics will high S_T are checked by a large number of applicants. If review panels are computed using the code in algorithm 1. For every application, a weight is assigned to each matching faculty based on the topics which are in the match. The following formulae are used:

2.1 By Faculty Scores

$$F(a) = faculty(M(a, \_)) \tag{1}$$

$$\forall f \in F(a)\ w_T(f, a) = \sum_{t \in M(f,a)} \frac{1}{S_T(t)} \tag{2}$$

$$L_f(a) = sort([(f_1, w(f_1, a)), (f_2, w(f_2, a)), ..., (f_2, w(f_2, a))], key = \lambda(a, b)(3)$$

The idea is to sort the faculty members with matching interest in such a way that faculty members matching a given application a through rarer topics would be considered first to be there in the panel of the corresponding applicant compared to someone who matches through a more popular keyword. For example, if for two topics, t_1 and t_2, if $S_T(t_1) > S_T(t_2)$, and two faculty members f_1 and f_2 are matched with a through t_1 and t_2, then, with everything else remaining the same, f_2 will be given more preference to be in the review panel of a as compared to f_1.

Algorithm 1 Algorithm to compute review panels

```
def calculate_review_panels(fac_apps):
  def second(t):
    _, b = t
    return b

  faculty_score, application_score, topic_score =
    calculate_scores(all_fac_apps)
  review_panels = {}
  for app in application_score:
    app_fac_list = []
    for fac in faculty_score:
      for f, a, ts in fac_apps:
        if f == fac and a == app:
          topic_weight = sum([1.0 / topic_score[t] for t in ts])
          match_score = topic_weight / faculty_score[fac]
          app_fac_list.append((fac, match_score))

    app_fac_list.sort(key=second, reverse=True)
    review_panels[app] = app_fac_list
  return review_panels
```

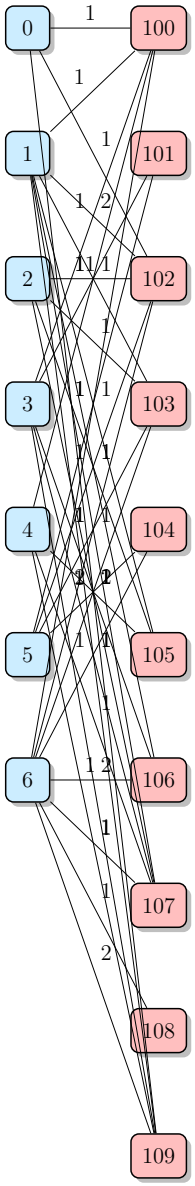

Figure 1: Faculty Candidate Mapping Graph

3 By Edge Ordering

We can compute interview panels by using a slight variant of the approach explained in the last subsection. Here, we compute the matching bipartite graph where the left group of nodes represents faculty and the right group represents candidates, just like before. The edge

annotations, this time, are $(f : F, ts : 2^T, c : C)$. Interview panels are computed using the following steps:

1. For all edges e, edge weight $W(e)$ is computed as follows:

$$W(e) = \frac{\sum_{t \in e.ts} \frac{1}{W(t)}}{W(e.f) \times W(e.c)} \tag{4}$$

2. The edges then are sorted in a descending order of their weight.
3. Finally, the interview panel for each candidate is computed by traversing the sorted edge list from top to bottom, and for each edge e encountered along the way, $e.f$ is added to $panels[e.c]$ if currently,
$|P(e.c)| < MAXSIZE \ \wedge$
$|panels(e.f)| < MAXPANELPERFACULTY$. Here, $P : C \mapsto 2^F$ maps each candidate to his/her interview panel. $panels : F \mapsto 2^{F \cup C}$ is the set of interview panels of which a given professor is a member.

4 Experiment