Matching Graduate Applicants with Faculty Members

Shibamouli Lahiri, Carmen Banea, and Rada Mihalcea

University of Michigan, Ann Arbor, MI 48109, USA
{lahiri, carmennb, mihalcea}@umich.edu

Abstract. Every year, millions of students apply to universities for admission to graduate programs (Master's and Ph.D.). The applications are individually evaluated and forwarded to appropriate faculty members. Considering human subjectivity and processing latency, this is a highly tedious and time-consuming job that has to be performed every year. In this paper, we propose several information retrieval models aimed at partially or fully automating the task. Applicants are represented by their statements of purpose (SOP), and faculty members are represented by the papers they authored. We extract keywords from papers and SOPs using a state-of-the-art keyword extractor. A detailed exploratory analysis of keywords yields several insights into the contents of SOPs and papers. We report results on several information retrieval models employing keywords and bag-of-words content modeling, with the former offering significantly better results. While we are able to correctly retrieve research areas for a given statement of purpose (F-score of 57.7% at rank 2 and 61.8% at rank 3), the task of matching applicants and faculty members is more difficult, and we are able to achieve an F-measure of 21% at rank 2 and 24% at rank 3, when making a selection among 73 faculty members.

Keywords: graduate application, statement of purpose, keyword extraction, information retrieval

1 Introduction

Every year, millions of students worldwide apply for graduate education in the United States. In Fall 2012 alone, US universities received 1.98 million graduate applications, and more than 461,000 students enrolled in graduate studies for the first time between Fall 2011 and Fall 2012.[1] With such a high number of students applying to US universities for graduate studies, and that number increasing over the years,[2] the problem of processing this voluminous amount of applicant data into a more manageable and more automated pipeline assumes paramount importance.

Ph.D. applicants in particular pose a greater challenge because they need to be screened for funding offers and matched with potential advisors. While some applicants do specify the group or the professor with whom they would like to work with,

[1] https://www.cgsnet.org/us-graduate-schools-report-slight-growth-new-students-fall-2012

[2] http://www.cgsnet.org/ckfinder/userfiles/files/R_IntlApps12_I.pdf

many do not provide a selection. The problem is somewhat alleviated by having a separate survey in online application forms that allows applicants to mention which faculty members they would like to work with, and rank those faculty members in order of preference. Still, it largely remains the university's and ultimately the departments' responsibility to ensure Ph.D. applicants are matched with appropriate faculty members. Departments typically employ a faculty subgroup or separate staff members to read through graduate applications, forward them to appropriate faculty members, and create online "profiles" of applicants so that they could be matched more easily with faculty members. The problem, however, is that not all faculty members toward whom an applicant shows interest can offer financial support or have a matching interest in the applicant.

Our goal in this project is to *automate the process of matching applicants with faculty members*. In particular, we want to leverage the free text available as part of the applications to aid us in the decision process. To showcase our approach, we use the applicant data from the Computer Science and Engineering department at a large Midwestern university that had over 1,100 graduate applications in Fall 2014. Manual matching of Ph.D. applicants with appropriate faculty members was also available. We designed several information retrieval systems that would:

1. Match applicants and research areas:
   (a) given an applicant, retrieve the most likely research areas the applicant would match;
   (b) given a research area, retrieve from the pool of available applicants those most likely to be a good match;
2. Match applicants and faculty:
   (a) given an applicant, retrieve those faculty members with similar research interests;
   (b) given a faculty member, retrieve the most likely applicants to possess similar research interests;
   (c) given an applicant, retrieve the most likely research areas the applicant would match, and then from those, select faculty members with similar research interests.

The rest of the article is organized as follows. We outline related studies in Section 2, followed by a description of our dataset in Section 3. Section 4 presents exploratory analysis of the keywords extracted from faculty published work and applicants' statement of purpose, setting the stage for Section 5, where we describe information retrieval systems and the importance of keywords in constructing them. Section 6 concludes the paper, outlining future research directions.

2 Related Work

The problem of matching graduate students with faculty members has three close analogs in natural language processing: authorship attribution, author profiling, and author-topic modeling.

In authorship attribution, the goal is to predict who authored a particular document. The problem is usually cast as a classification task, where we have a large set of training documents with known authors, and a smaller set of test documents with unknown authors. Machine learning models are trained on the training documents, and then deployed on test documents to predict the unknown authors. For details on authorship attribution, please see the surveys by Juola [1], Stamatatos [10], and Koppel et al [3]. In some flavors of authorship attribution, test documents are used as search queries against training documents, and the author of the top-ranked (training) document is considered predicted label [10]. In our study, we consider papers written by faculty members as "training documents", and statements of purpose written by students as "test documents". Performance on the test set is judged based on the ground truth faculty-applicant pairing we have. A potential limitation of this approach comes from the fact that in authorship attribution, we would like to uncover the *writing style* of an author, whereas in this case, we are interested in the *content match* between a paper written by a faculty member, and a statement of purpose authored by an applicant. We resolve this issue by using keywords (cf. Sections 4 and 5).

Author profiling is very similar to authorship attribution, except that the goal here is to build a "stylistic profile" of an author instead of predicting a class label. The profile is usually a vector of words and/or phrases frequently used by the author, and may also include grammatical constructs and parse tree fragments. An author is represented by several vectors that are built on documents written by him/her. These vectors can be used to identify the author's unique writing style (*fingerprint*) and to extract other useful properties such as gender, age, education, and personality traits. Author profiling has been discussed in depth in the survey by Stamatatos [10]. In our case, author profiling could serve as a fundamental building block where papers written by faculty members are used to create their authorship profiles, and then a statement of purpose that is most similar to a faculty member's profile, is assigned the corresponding faculty member. This approach, albeit sound in principle, has the same important drawback as authorship attribution; it focuses on *stylistic* rather than content information, and is therefore not very useful.

Content information of authors can be explicitly incorporated in a probabilistic setting, where documents are modeled as a collection of topics, and topics are modeled as a collection of words. Topic generation depends on authors represented as (observed) random variables in the model [8]. An unseen document can be assigned a probability distribution over authors and topics, thereby helping find out which authors are the most likely to have written that document. In our case, we could use the set of papers written by faculty members to train an author-topic model, and then the statements of purpose could be "folded in" the model to extract their most representative author and topic probability distributions.

While all the above ideas are good, we did not find an approach that closely matches our purpose. The only similar study we found comes from IBM India Research Lab [9]. They designed a system called "PROSPECT" to screen candidates for recruitment. Their system combines elements from recommender systems, information retrieval, and author profiling to come up with a software and graphical user interface that improves candidate ranking by 30% and provides faceted search functionality to conduct finegrained analyses such as highest degree of the candidate, relevant and total work experience, skills, and his/her city of residence. Since companies like IBM receive thousands of job applications for many job postings, it becomes crucial to augment the slow and cumbersome manual candidate-screening process with an automated decision-making tool such as PROSPECT. Our use case is also very similar, in that we want to screen hundreds of graduate applicants and match them with potential advisors. In our case, faculty members serve the same purpose as human resources staff screening job postings, and graduate applicants are similar to job candidates. Inspired by PROSPECT, we pursued five keyword-based approaches to tackle this problem. All approaches use information retrieval techniques, and stand to benefit from *learning to rank*, given enough data [5].

3 Data Description

Since our problem formulation involves the ranking of *faculty members* against *applicants* (and vice versa), we need a convenient textual representation for both. We opt to represent applicants by their *statements of purpose* (SOP), and faculty members by the papers they have (co-)authored in the prior 12 years (between 2004 and 2015). Anonymized statements of purpose are available for all applicants in the Fall 2014 cohort at the Computer Science and Engineering department at the university in question. Note that SOPs usually talk about what the applicant has achieved in the *past*, what (s)he is doing at *present*, what (s)he would like to do/be in the *future*, and how all these *connect* with the particular department and its faculty.

Papers were collected for 73 faculty members from their Google Scholar Citations[3] and DBLP[4] profiles. We collected 4,534 papers authored between 2004 and 2015, and converted their PDFs into text using UNIX *pdftotext* utility. Sometimes multiple faculty members collaborate on a single paper; we counted those papers multiple times, once for each participating faculty. Authorship statistics of the 5 most prolific authors are shown in Table 1. Note that a few of the most prolific authors wrote over 200 papers between 2004-2015, or almost 17 papers a year. This data follows a power-law distribution with exponent $\alpha = 3.45$ (statistically significant with p-value = 0.999).

Faculty Member	**Number of Papers**
Tommy M. Rosenbalm	331
Thomas M. Burns	300
Ali H. Salgado	212
Richard G. Meza	146
Nicole L. Thompson	140

Table 1: Number of papers (co-)written by several faculty members (anonymized) between 2004 and 2015.

[3] http://scholar.google.com/

[4] http://www.informatik.uni-trier.de/ ley/db/

Faculty Member	**Number of Applicants**
Richard C. Hardy	45
George E. Ford	45
Robert S. Peters	42
Jeff L. Jurgens	41
Dennis R. Salisbury	38

Table 2: Number of applicants assigned to several faculty members (anonymized).

Research Area	**Number of Faculty**	**Applicant to Faculty Ratio**	**% of Applicants in Area**
Artificial Intelligence	**29**	7.03	**67.11**
Chip Design, Architecture, and Emerging Devices	22	3.41	24.67
Databases and Data Mining	6	**16.00**	31.58
Embedded and Mobile Systems	12	6.67	26.32
Human-Computer Interaction	8	6.63	17.43
Languages, Compilers, and Runtime Systems	13	3.54	15.13
Networking, Operating Systems, and Distributed Systems	16	4.56	24.01
Robotics in CSE	7	11.71	26.97
Secure, Trustworthy, and Reliable Systems	22	4.32	31.25
Theory of Computation	10	5.4	17.76
Warehouse-Scale and Parallel Systems	19	4.16	25.99

Table 3: Research areas at the Computer Science and Engineering department at a large Midwestern university. Highest value in each column is boldfaced. Applicants are from Fall 2014 pool.

We also obtained a pairing of Ph.D. applicants (Fall 2014 cohort) with faculty members, constructed manually by a small group of faculty. Note that each applicant is identified by a numeric ID and may be matched with multiple faculty members. On the other hand, a faculty member is represented by his/her username, and may be matched with (or express interest in) several different applicants. There were 1107 applicants in total, of which 304 were matched with a faculty member. Different faculty members received a different number of applications. Faculty members receiving the highest number of applications in Fall 2014 cohort are shown in Table 2.

The faculty conducts research in 11 different areas, as shown in Table 3. The areas vary in terms of number of faculty, percentage of applicants, and applicant-to-faculty ratio. Artificial Intelligence (AI), for example, has the highest number of faculty members and the highest percentage of applicants. Databases and Data Mining, on the other hand, comprises the lowest number of faculty and the second highest percentage of applicants, which leads to the highest applicant-to-faculty ratio across all research areas. These observations could be helpful in identifying areas where additional faculty members need to be recruited.

4 Exploratory Analysis of Keywords

To represent the SOPs and papers by their *content* rather than *style*, we use an automatic system to extract keywords. We employ a state-of-the-art system previously used in the email domain [4]. Keyword statistics are provided in Table 4; note that we also include the counts for filtered keywords using Wikipedia article titles to obtain a more salient listing of keywords.

Keyword Type	**SOPs Keyword Count**	**Papers Keyword Count**
All keywords	53,166	123,171
Multi-word keywords	44,473	98,470
All keywords after filtering	13,472	27,563
Multi-word keywords after filtering	6,022	10,170

Table 4: Keyword statistics.

machine learning	1166.78
computer vision	713.66
computer science and engineering	706.06
artificial intelligence	679.31
computer architecture	651.95
data mining	562.43
electrical engineering	516.14
natural language	493.75

Table 5: Top multi-word keywords from SOPs, ranked by *tf.idf*.

We first want to see *what students talk about most* in their SOPs in terms of keywords. Table 5 shows that the most salient keywords in SOPs are general and trendy terms such as “machine learning,” “artificial intelligence,” “data mining,” and “computer vision.” Other terms are even more generic, such as “computer science and engineering,” and “electrical engineering.”. These keywords indicate that students are indeed familiar with the trendy terms and buzzwords in Computer Science and Engineering, and most students want to go to those areas. In comparison, when we look at *what the faculty talk about most* in their papers (cf. Table 6), we observe highly technical terms and domain-specific keywords such as “nash equilibrium” and “episodic memory.” This observation leads support to the fact that students are usually not sufficiently aware of the publication records of different faculty members (*information gap*), and students usually apply to "hot" areas rather than established areas (where there are more papers), perhaps because of increased media attention to those areas. This information gap further shows that our problem is complex, as we need to match texts from students and texts from faculty containing disparate sets of keywords.

2004	**2005**	**2006**	**2007**
test set	file system	file system	natural language
ubiquitous computing	sensor network	natural language	file system
power management	error rate	data set	data race
computer science	virtual machine	error rate	ad hoc
lower bound	power management	ad hoc	energy consumption
2008	**2009**	**2010**	**2011**
file system	**network virtualization**	power consumption	data race
control logic	control logic	file system	shared memory
power consumption	power consumption	**reward function**	**episodic memory**
energy consumption	data set	computer science	error rate
computer science	virtual machine	signal processing	**medical device**
2012	**2013**	**2014**	**2015**
energy consumption	electrical engineering	**social media**	**homomorphic encryption**
energy efficiency	**data center**	anomaly detection	data mining
nash equilibrium	computer science	power consumption	anomaly detection
computer science	natural language	data mining	**data science**
electrical engineering	energy efficiency	**computational linguistics**	**reinforcement learning**

Table 6: Most important keywords from papers published in different years. Importance was measured by *tf.idf*. Top keywords that are unique to each year are shown in boldface.

An intriguing question at this point is to explore *how the keywords change over the years*. We analyzed the publications of all faculty members by year and ranked the keywords used by *tf.idf*. Table 6 shows that there is a distinct *trend* in the top-ranked keywords, in the sense that each year seems to focus on some particular problems (perhaps at the expense of others), and each year has some *new problems* that were not salient before. Year 2014, for example, introduces "social media" as a salient keyword, whereas year 2015 introduces "data science." It is important to note that graduate applicants are often not aware of such subtle variations and trends going on in the research community and thus cannot prepare accordingly.

We next explore *how the faculty members rank according to their diversity and focus* of research topics, as related to applicants. While diversity is usually defined as the opposite of similarity in Information Retrieval [7], we measured *diversity* in the context of keywords by Jaccard Similarity[5] between all keywords of a faculty and keywords from all applicants, whereas *focus* was measured by Jaccard Similarity between all keywords of a faculty and keywords from applicants assigned to him/her. Table 7

[5] https://en.wikipedia.org/wiki/Jaccard_index

Diversity	**Focus**	**Content Density**
Nicole L. Thompson	**Stephen M. Evans**	Jack J. Santoro
Francis G. Okelley	**Richard C. Hardy**	Rodolfo C. Hayes
John L. Wheatley	**Kevin D. Llanes**	Nicole L. Thompson
Tommy M. Rosenbalm	**George E. Ford**	James C. Rhinehart
Ali H. Salgado	Michael M. Lewis	Francis G. Okelley

Table 7: Ranking of faculty members (anonymized). Top faculty members that are unique to a particular ranking are shown in boldface.

shows that these two rankings are substantially different. Furthermore, looking at *content density* (total number of keywords as a fraction of total number of words – averaged over papers), we see that the ranking changes again. It is important to note such subtle differences, because they help applicants make an informed decision.

Intriguingly, we find *focus* to be highly positively correlated with *popularity* (Spearman's ρ = 0.8), where the latter is measured by *how many students are assigned to a faculty* (cf. Table 2). *Diversity* and *popularity* are only moderately correlated (Spearman's ρ = 0.28), whereas the correlation between *diversity* and *focus* is even lower (ρ = 0.12). Very low correlation is observed between *content density* and *focus* (ρ = 0.04). Similarly low values are obtained for correlations between *content density* and *popularity*.

5 Information Retrieval Models

The objective of our study is to help academic departments *match applicants with faculty members*. We cast this problem as an *information retrieval*-like task, where given an applicant as query, our system retrieves research areas and faculty members. The system is also able to retrieve applicants with respect to faculty members as queries. We consider the following use cases:

1. match applicants and research areas
   (a) consider an applicant's statement of purpose as a query, while all publications in a given research area form a single document, and retrieve the most similar of these documents; retrieval is done among 11 research areas. We will call this variation *SOP as query, research areas as documents*.
   (b) consider all publications in a research area as a query for which we seek to retrieve the strongest matching statement of purpose pertaining to the applicants; retrieval is done among 304 applicants. We will refer to this variation as *research area as query, SOP as documents*.
2. match applicants and faculty
   (a) consider an applicant's statement of purpose as a query, while all publications pertaining to a given faculty as a single document; the retrieval is done for the most similar documents. This variation is represented as *applicant as query, faculty members as documents*; retrieval is done among 73 faculty members.

(b) consider the cumulative publications of a faculty member as query, while each applicant is represented through his / her statement of purpose. This variation is referred to as *faculty as query, applicants as documents*. Retrieval is performed among 304 applicants.

(c) consider an applicant's statement of purpose as a query. Retrieval of the most relevant faculty members is performed hierarchically, first with respect to the best matching research groups (represented through the totality of articles published by faculty in that group), and then with respect to the best matching faculty members from within the top groups. We will refer to this variation as *applicant as query, faculty members as documents – hierarchical*; retrieval is first performed against the 11 research areas, and then against the faculty members in the top research areas.

While applicant publications and/or data gathered from application forms could potentially be used to match applicants with faculty, we considered such an approach to be problematic because of the difficulty in gathering data, lack of prior publications (esp. for Master's applicants), and penalizing applicants that mostly have industry experience.

5.1 Vector Generation

For each one of the approaches mentioned above, vectors are generated for different feature types, filtering, and weighting options.

Feature types. Two types of vectors are derived to represent a query or a document: using the vocabulary of single words encountered in the text (*unigrams*), or using the keywords encountered in the same text (*mwe*[6]). While the first technique is straightforward, for the second technique we extract keywords from applicant statements of purpose (SOPs) using a state-of-the-art supervised keyword extractor [4] trained on two keyphrase extraction corpora. The first corpus consists of a set of 211 academic papers with keyword annotations [6], while the second corpus was released as part of the SEMEVAL 2010 Keyphrase Extraction Task [2] and also encompasses a set of 184 academic papers annotated for keywords. The extractor uses noun phrases and named entities as candidates, as well as surface, frequency, phraseness, and graph-based features; it performs shallow post-processing after extraction to remove punctuation.

Filtering. The unigrams and the keywords mentioned above are referred to in the ensuing experiments as *all*, since they do not undergo filtering. A second instance of these features is derived, based on whether they are associated with a Wikipedia article[7]; this list is referred to as *filtered*, and retains fewer, higher quality and more salient entries.

We should emphasize that all the vectors are constructed on keywords/unigrams extracted from SOPs rather than those appearing in the published articles. The SOP-derived keyword list / vocabulary tends to be more generic and concise, as applicants do not yet have an in-depth grasp of various research areas and their SOP is shorter than

[6] "mwe" stands for multi-word expressions.

[7] Listing of article titles retrieved from https://dumps.wikimedia.org/

an article, thus allowing the vectorial space to model applicants more closely while also being more efficient.

Weighting options. The above feature types are weighted using three common weighting schemes: *binary*, term frequency (*tf*), and term frequency inverse document frequency (*tf.idf*).

Information retrieval framework. Using a query vector, document vectors are ranked with respect to their cosine similarity computed against the query vector, and the top k are retrieved by the system. The system predictions are evaluated against ground truth faculty-applicant pairings that were manually derived by a small group of faculty members. Performance was measured using standard precision, recall, and F-score at different ranks (k).

Overall, we construct 12 vector space models encompassing all the combination of parameters detailed above. The most robust results are obtained using: keywords and unigrams (for vocabulary), tf.idf (for feature weighting), and all and filtered (for filtering). As such, in the subsequent discussions we will focus on these variations. The baseline is represented through the combination *unigram all tf.idf*, namely using all the vocabulary encountered in the SOPs as unigrams with tf.idf weighting.

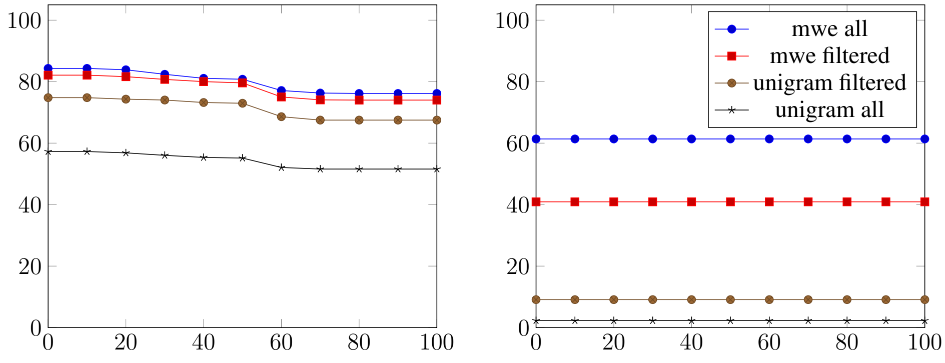
5.2 Matching Applicants and Research Areas

Our first use case scenario matches applicants and research areas. This scenario allows departmental faculty or staff to be provided with the best research areas for a given candidate, and then manually assign candidates to faculty in those areas, thereby simplifying the matching process. We explore two venues:

1. Applicant as query, research areas as documents.
2. Research area as query, applicants as documents.

Figure 1a shows the interpolated precision-recall curve for the first approach (SOP query, area documents), while Figure 1b shows the same metrics for the second approach (area query, SOP documents) all of these derived for rank $k = 5$. We note that the first approach performs significantly better, achieving an interpolated precision level of over 80%, compared to the best performing variation falling under the second approach, which achieves an interpolated precision level of approximately 60%. Focusing on the first approach, the best performing variation is *mwe all tf.idf*, but is closely followed by *mwe filtered tf.idf*. Given that the former uses approximately 44 thousand dimensions, while the latter uses only 6,022 dimensions, we can conclude that (1) modeling via multi-word keywords is significantly better than accounting for the entire vocabulary, and (2) filtering these keywords for saliency achieves a more compact and efficient model, without a meaningful drop in performance.

Figure 2 shows the corresponding F-score curve for the two approaches, this time for different ranks. We notice that the best F-score of 61.8% occurs at rank 3 (21.2% higher than the corresponding baseline), while the second best F-score of 59.9% is encountered at rank 4 (where the baseline F-score is the highest, yet the prediction still surpasses it by 17.5%). A higher F-score is to be expected in this scenario compared to results achievable for matching students and faculty, since here we are limiting our



(a) SOP query, area documents.

(b) Area query, SOP documents.

Fig. 1: Interpolated precision-recall curves (at $k = 5$) showing two approaches for matching applicants and research areas, each with four variations. X-axis shows the recall level (%), while Y-axis shows the interpolated precision level (%).

match to 11 research areas.[8] We should stress that the optimal usability outcome for this task is represented through high performance at low ranks, i.e. the system should *correctly* retrieve a few matching research groups for a given SOP; as shown, the system achieves very high performance for ranks 2 through 4. This should accurately guide the process of assigning professors from those top retrieved groups and reduce the amount of manual work involved.

5.3 Matching Applicants and Faculty

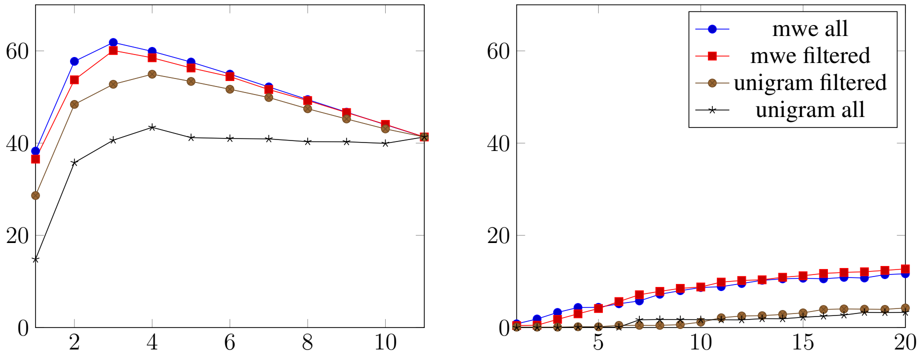
The second and more desirable scenario consists of matching applicants and faculty. This allows the entire task to be automated, and therefore provides most savings in terms of financial and human resources for a department. We identify three venues:

1. applicant as query, faculty members as documents
2. faculty member as query, applicants as documents
3. applicant as query, faculty members as documents – hierarchical

The first two are similar to those proposed in the previous section, but this time the match is done directly with the faculty member, while the third consists of a hierarchical approach, where the match is first performed in regards to the best matching research group, and then the faculty is retrieved from within that group.

Probing further into the behavior of our system and baseline, we plotted the *interpolated precision-recall curve*, averaged over all search queries at a rank k=5. The resulting graphs are shown in Figure 3. We observe that similarly to the equivalent variations in Section 5.2, the SOP (applicant) query-based retrieval outperforms faculty

[8] The random baseline in this scenario is 9.1%.



(a) SOP query, area documents.

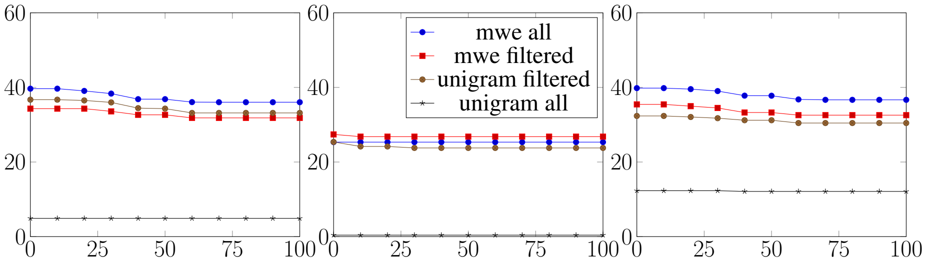
(b) Area query, SOP documents.

Fig. 2: F-score curves for two approaches for matching applicants and research areas, each with four variations. X-axis shows the Rank, while Y-axis shows the F-score (%).

query-based retrieval under all variations (see Figures 3a and 3b). This is to be expected, since in the first scenario the retrieval is made among 73 faculty members, while in the second scenario, it is made among 304 applicants. Enacting a hierarchical based approach which generates an intermediary mapping to research areas and then retrieves the strongest matching faculty candidates from within the returned areas, achieves a similar performance to the first approach directly mapping to faculty. (see Figure 3c). As in the previous subsection, the best variation remains *mwe all tf.idf*, with a performance of approximately 40% interpolated precision level, 35% higher than the *unigram all tf.idf* baseline achieving slightly below 5% interpolated precision level. This shows that keywords rather than vocabulary offer a lot more plasticity and bring more value for this task.

Figure 4 showcases the performance of the three approaches matching applicants to faculty at different ranks. Here as well, the *SOP query, faculty documents* represents the highest performing use case, retaining its high performance at low k values by achieving a F-score higher than 19% for ranks 2 through 10 using the best performing *mwe all tf.idf* variation, and reaching a maximum of 24.4% for rank 3. The hierarchical system displays a similar performance, as it is able to attain an F-measure above 19% starting at rank 2 as well, but since it is a two step system, it is too inefficient compared to a one step system to motivate its usage. The random baseline accuracy for matching an applicant to a faculty member is 1.4%.

Considering all the use cases, however, we can say that we are able to successfully retrieve research areas and faculty members against SOP queries. Our system always surpasses the baseline by a wide margin, and using the *mwe all tf.idf* variation consistently achieves the best results. This is a great boon for the faculty members and staff members, because instead of manually sifting through hundreds of applications, they can now use our system to screen applicants before starting the laborious manual

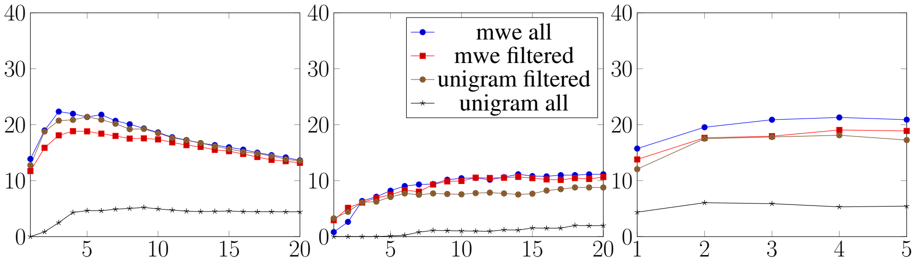


(a) SOP query, faculty documents.

(b) Faculty query, SOP documents.

(c) SOP query, faculty documents – hierarchical.

Fig. 3: Interpolated precision-recall curves (at $k = 5$) showing three approaches for matching applicants and faculty, each with four variations. X-axis shows the recall level (%), while Y-axis shows the interpolated precision level (%).



(a) SOP query, faculty documents.

(b) Faculty query, SOP documents.

(c) SOP query, faculty documents – hierarchical.

Fig. 4: F-score curves showing three approaches for matching applicants and faculty, each with four variations. X-axis shows the Rank, while Y-axis shows the F-score (%).

checking process. Anecdotal evidence from faculty members in our department showed that this was indeed the case, and they were happy with the search results produced by our system.

6 Conclusion

In this paper, we introduced a new task – matching graduate applicants with faculty members using text-based features. The problem is complex, given that there are no standard annotated datasets, not much relevant related work, and *content disparity* between the textual materials authored by applicants and those authored by faculty members. We created our own dataset comprising 4,534 papers authored by 73 different faculty members at the Computer Science and Engineering department of a large Midwestern university. We further considered an in-house set of 1,107 statements of purpose, and a set of 788 faculty-applicant pairings constructed manually. Keywords were extracted from papers and SOPs, and a detailed exploratory analysis was performed leading to insights regarding the content depth and subtlety of documents. We obtained encouraging results using standard information retrieval techniques using five different use cases, concluding that keywords offer a significantly better representation (more efficient and better results) compared to bag-of-words variations. Overall, we are able to match students to research groups with an F-score of 62%, while for the more difficult task of matching students to faculty, we are able to achieve a 24% F-score. Our future work includes obtaining more data (especially applicant data), more reliable faculty-applicant annotations, and more sophisticated models that take into account the sparsity of the task.

Acknowledgements

We are grateful to Lauren Molley and Joseph Zimmer for their help with the implementation of some system components. We also want to thank the reviewers for their valuable feedback. This material is based in part upon work supported by the Michigan Institute for Data Science, by the National Science Foundation (grant #1344257), and by the John Templeton Foundation (grant #48503). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Michigan Institute for Data Science, the National Science Foundation, or the John Templeton Foundation.

References

[1] Patrick Juola. Authorship Attribution. *Found. Trends Inf. Retr.*, 1(3):233–334 (2006)

[2] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, Stroudsburg, PA, USA. Association for Computational Linguistics (2010)

[3] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.*, 60(1):9–26 (2009)

[4] Shibamouli Lahiri, Rada Mihalcea, and Po-Hsiang Lai. Keyword Extraction from Emails. *Natural Language Engineering*, 23(2):295–317 (2017)

[5] Hang Li. A Short Introduction to Learning to Rank. *IEICE Transactions*, 94-D(10):1854–1862 (2011)

[6] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase Extraction in Scientific Publications. In *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, ICADL'07, pages 317–326, Berlin, Heidelberg. Springer-Verlag (2007)

[7] Barry Smyth and Paul McClave. Similarity vs. Diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '01, pages 347–361, London, UK. Springer-Verlag (2001)

[8] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The Author-topic Model for Authors and Documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States. AUAI Press (2004)

[9] Amit Singh, Catherine Rose, Karthik Visweswariah, Vijil Chenthamarakshan, and Nandakishore Kambhatla. PROSPECT: A System for Screening Candidates for Recruitment. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 659–668, New York, NY, USA. ACM (2010)

[10] Efstathios Stamatatos. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556 (2009)