Extracting Skill Endorsements from Personal Communication Data

Darshan M. ShankaraLingappa*
Helsinki Institute for Information Technology and Dept. of Computer Science
Aalto University, Finland

Gianmarco De Francisci Morales*
Qatar Computing Research Institute
Doha, Qatar

Aristides Gionis*
Helsinki Institute for Information Technology and Dept. of Computer Science
Aalto University, Finland

ABSTRACT

People are increasingly communicating and collaborating via digital platforms, such as email and messaging applications. Data exchanged on these digital communication platforms can be a treasure trove of information on people who participate in the discussions: who they are collaborating with, what they are working on, what their expertise is, and so on. Yet, personal communication data is very rarely analyzed due to the sensitivity of the information it contains.

In this paper, we mine personal communication data with the goal of generating *skill endorsements* of the type *"person A endorses person B on skill X."* To address privacy concerns, we consider that each person has access only to their own data (i.e., conversations with their peers). By using our method, they can generate endorsements for their peers, which they can inspect and opt to publish.

To identify meaningful skills we use a knowledge base created from the StackExchange Q&A forum. We study two different approaches, one based on building a *skill graph*, and one based on *information retrieval* techniques. We find that the latter approach outperforms the graph-based algorithms when tested on a dataset of user profiles from StackOverflow. We also conduct a user study on email data and find that the information retrieval-based approach achieves a MAP@10 score of 0.617.

Keywords

personal data; skill endorsements; e-mail mining

1. INTRODUCTION

The ability to identify individuals who are experts on certain skills is an essential element for organizational effectiveness. Finding experts an important ingredient in a modern digital economy, where employers use online platforms to outsource tasks to experts.

Traditionally, finding experts is facilitated by professional social networks or knowledge bases. In these systems, individuals self-declare their areas of expertise. For example, users in LinkedIn[1] build their profile by reporting their skills and expertise. However, such systems suffer from noisy and incomplete data, as well as user biases.

In this paper we introduce a new approach to build a *skill endorsement graph*, which can be used for finding experts. Our goal is to generate *skill endorsements* of the type *"person A endorses person B on skill X."* An important feature of our approach is that it leverages personal communication data, such as email, messaging applications, or discussion fora. Our assumption is that data collected from such platforms can used to generate accurate skill profiles based on what people actually do during their working days: which projects they work on, which topics they discuss, which tools they use, and so on.

On the other hand, communication data may contain private information, so it is important to develop methods that extract skills in a privacy-preserving manner. To address these privacy concerns, we consider that each person has access only to their own data (i.e., conversations with their peers). Thus, users can generate endorsements for their peers, which they can inspect and opt to publish. The skill-endorsement graph for a set of individuals can then be constructed by aggregating all individual-level skill endorsements.

A major challenge to any skill-extraction method is to identify a set of tokens that can be considered as skills. To address this challenge, we build a knowledge base of skills by processing the StackExchange Q&A forum. Our underlying assumption is that tags that appear in StackExchange posts are considered to be skills. Given the knowledge base extracted from StackExchange and individual-level communication data (such as email conversation or messaging), we study two different approaches for extracting skill endorsements between a person and their peers. The first approach is based on matching a set of keywords extracted from the communication data of a person to an underlying *skill graph* and finding the most representative skills via *personalized PageRank* (PPR). The second approach is based on *information retrieval* techniques, in particular, on similarity in a vector space model. A vector built from a person's commu-

*Emails: darshan.mallenahallishankaralingappa@aalto.fi; gdfm@acm.org; aristides.gionis@aalto.fi

[1] http://www.linkedin.com

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983884

nication text is used as a query against the StackExchange discussions to retrieve the most relevant associated skills.

We evaluate the proposed methods by using StackOverflow users for which ground-truth skills are known, as well as by conducting a user study on real email data. Our results show that the information retrieval approach outperforms significantly the graph-based approach.

2. RELATED WORK

Literature on email mining is quite sparse, mostly due to the sensitive nature of the data, and the privacy concerns of the users. Here we review the few studies that have tackled the problem of extracting skills from email data.

?] are the first to consider the problem of finding expertise via email communications. Their proposed system runs on the server of an organization having access to the emails of all the employees. First, their system collects all the emails associated with a given query. Then, it builds a graph by considering all pairs of users who have an email exchange within the email collection. Finally, it uses the HITS algorithm **?**] to extract authoritative sources from the graph. In a similar fashion, **?**] use a probabilistic approach to model associations between individuals for a given query topic.

Both previous methods find experts with respect to a given query. The scenario envisioned in this paper is different, as we extract skill endorsements for each person participating in email or online forum communication without requiring any input query. Furthermore, these methods require access to all the emails within the organization, while our method assumes access only to the communication data of a single person, in order to alleviate privacy concerns. Thus, our method is not directly comparable with these previous works.

On the other hand, **?**] consider the problem of extracting skills from a text document. First they build a skill graph from LinkedIn and Wikipedia. Then, for a given input document, they identify the most similar Wikipedia pages. These pages are used as seed nodes in a random walk that provides skill summarization. Our method is similar in spirit, but we use a different knowledge base.

3. BUILDING A KNOWLEDGE BASE

A central concept to our approach is the *skill knowledge base*, which is used to model the relationship among skills, and relate skills with other keywords found in the personal communication data. To build a high-quality skill knowledge base we formulate the following desiderata:

Coverage: represent a wide range of skills as comprehensively as possible;

Quality: accurately reflect the relationship among skills, and the relationship among skills and other keywords;

Automation: it should be built and updated automatically.

As a starting point for our skill knowledge base we use StackExchange,[2] a Q&A platform that encompasses 155 different topics, such as math, computer science, finance, law, and history. The platform hosts about 13 million questions, 21 million answers, and has more than 8 million users across all its sites. These websites cover a wide range of skills, and the data is publicly available.[3]

The users of StackExchange *tag* their questions to categorize their posts. Our main idea is to use these tags as skills, and our skill knowledge base relies heavily on these tags. In this paper we explore two alternatives: a graph-based approach, where we only model relationships between tags/skills, and a vector space-based approach, where we index the whole Q&A repository annotated by tags/skills.

In the graph-based approach, we form a *skill graph* where vertices represent tags and edges represent co-occurring tags. Edge weights are co-occurrence counts. More formally, let $\mathcal{D}$ be the set of all posts in StackExchange and $\mathcal{T}$ the set of unique tags. Let $D(B, T) \in \mathcal{D}$ be a single post, where B is the body of the post and $T \subseteq \mathcal{T}$ is the set of tags. The skill graph is a weighted graph $G(V, E, w)$ with $V = \mathcal{T}$. An edge between two vertices i and j exists if and only if there exists a post $D(B, T) \in \mathcal{D}$ such that $i \in T \wedge j \in T$. The weight of the edge is $w_{ij} = |\{D \in \mathcal{D} \mid i \in T \wedge j \in T\}|$.

In our preliminary experiments, we found that the quality of results improves when ignoring tags that appear fewer than 100 times. These low-frequency tags represent niche and uncommon skills. The final graph G has approximately $|V| = 38\,\mathrm{k}$ vertices and $|E| = 3.4\,\mathrm{M}$ edges.

Our second approach uses information retrieval techniques and indexes the whole Q&A repository annotated by the associated tags. In particular, we build an inverted index $\mathcal{I}$ on the body B of the posts $D \in \mathcal{D}$, and the tags associated with each post are retained to create a *tagged vector space*. After pre-processing the whole StackExchange dataset, the resulting index contains about 22 M posts.

4. EMAIL PROCESSING

In our email-processing pipeline, we consider that users are executing our algorithms on their own emails, and they produce skill endorsements for their peers. The emails are extracted and then, the text is cleaned and finally we identify and extract entities (noun phrases) via standard NLP tools.

Let us call *user* the individual whose emails are being analyzed (i.e., the source of the endorsements), and *peers* the other individuals that user has communicated with (i.e., the targets of the endorsements). We identify the email threads (set of emails having the same subject) associated with a peer. By using the process described above, we extract a list of entities for each email thread, which summarizes the thread, and which the algorithms described next can use to infer the endorsements.

Note that our method can also be used to other types of personal communication data, such as online fora. In such cases an appropriate data-processing pipeline should be established.

5. SKILL EXTRACTION ALGORITHMS

In this section we describe the two proposed approaches for extracting skill endorsements from email data, a graph-based algorithm and an index-based algorithm.

5.1 Graph-based algorithm

This algorithm uses the skill graph described in Section 4. Given the entities extracted from the emails between a *user* and a *peer*, the goal is to match the entities to the vertices of the skill graph and summarize all matched entities with the most important skills. This is achieved in three steps,

[2] https://stackexchange.com

[3] https://archive.org/details/stackexchange

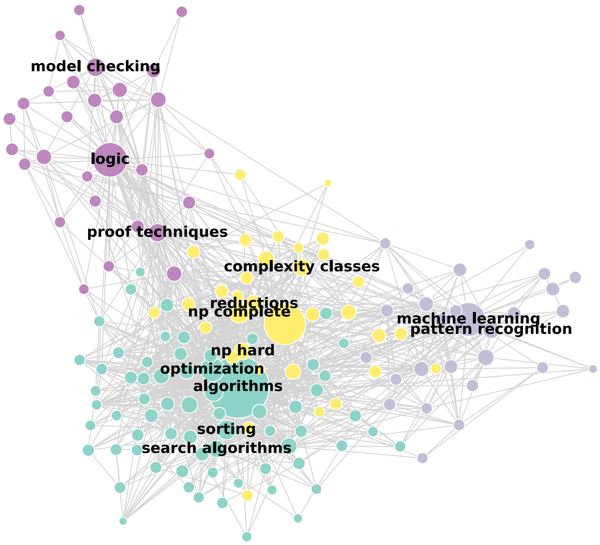

Figure 1: Graph clusters in the computer science part of the skill graph.

described next. We refer to this algorithm as GSE (graph-based skill endorsements).

Graph clustering. As a preprocessing step, we partition the skill graph into topical clusters. We use the Louvain algorithm [**?**], a hierarchical agglomerative method that optimizes modularity. Figure 1 illustrates the output of the clustering algorithm when considering the skill graph created on the *computer science* part of StackExchange. The clusters are very homogeneous and well defined, for instance, it is possible to recognize the theoretical computer science cluster in the middle in yellow, and the data mining and machine learning cluster on the right in grey.

Entity matching. In the next step we consider the email thread between a *user* and a *peer*, for whom we want to generate skill endorsements. Applying the processing pipeline described before, we extract entities from the email thread, and we match those entities and terms on the skill graph. When an entity is found in the skill graph, we say that the corresponding vertex is *matched*. The matching of entities induces a subgraph on the skill graph, which represents a set of candidate skills that can be used to infer the skill endorsements. In most cases an email thread is related to a small number of topics, so the matched skills of the thread will be located in a small number of clusters of the skill graph. To improve the relevance of the extracted skills, we focus our attention on the subgraph formed by taking the union of all the clusters that contain at least one matched skill.

Skill ranking. Let G' be the subgraph formed in the previous step (either the whole skill graph or the subgraph of the matching clusters), and let M be the set of matched skills in G'. In the final step, we summarize the skills M by considering the global structure of G'. For example, if the skills *'decision trees'* and *'nearest neighbor algorithm'* have been matched, we can summarize them with the skill *'supervised learning'*, which is highly connected with the former two, even though the latter skill has not been matched.

This task is achieved by centrality-based techniques, used to find important vertices in the graph G' with respect to the matched vertices M. There is a plethora of such centrality measures. In our experiments, we surveyed a number of these, including PageRank (PR), personalized PageRank (PPR), normalized personalized PageRank (NPPR $= {}^{\text{PPR}}/_{\text{PR}}$), effective importance (EI $= {}^{\text{PPR}}/_{w(v)}$) [**?**], and harmonic centrality [**?**]. We write GSE(X), where $\mathrm{X} = \mathrm{PR}, \mathrm{PPR}, \mathrm{NPPR}, \ldots$, to refer to algorithm GSE with centrality measure X. Our results show that the variants of PageRank perform similarly, with PPR being the best variant, while harmonic centrality performs the worst. Therefore, for sake of brevity, in some cases we restrict our attention to GSE(PPR).

5.2 Index-based algorithm

The second algorithm, referred to as ISE (index-based skill endorsements), uses the tagged vector space version of the knowledge base. Given an email thread for which we want to infer skill endorsements, ISE uses the thread as a query, and considers the tags of the retrieved posts as candidate skills.

Equivalently, ISE can be described as a k-NN multi-label classification method. In more detail, given a query, the algorithm retrieves top-k closest vectors according to a BM25-like similarity function. The algorithm then assigns a score $r(t)$ to each tag as the sum of the similarity of the vectors in which it appears

$$r(t) = \sum_{v \in R} \begin{cases} \mathrm{sim}(q, v) & \text{if } t \in \ell(v) \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where R is the set of top-k nearest neighbor vectors. The ranking of the skills is the one induced by $r(t)$.

6. EXPERIMENTAL EVALUATION

We evaluate our methods on several different tasks. First, we consider a prediction task on StackOverflow users. Then we report the results of a user study involving email data. We provide further evidence by running our algorithms on Apache Spark mailing list. Finally we test our algorithms on the public emails of Hillary Clinton.

StackOverflow users. We consider predicting the tags used by a given StackOverflow user given their posts. We use the body of the posts as the input to our text-mining pipeline. The ground truth is the set of tags associated with the user, while we assume that these tags are the skills of the user. Due to sparsity in the model, it is likely that the prediction performance depends on the number of posts of a user. That is, we hypothesize that the prediction task becomes easier for users with more data. To test this hypothesis, we draw a random set of 150 StackOverflow users by performing stratified sampling according to the quantile of the number of their posts across the whole population. The posts by these 150 users are excluded from the knowledge base.

We compare two different versions of the GSE algorithm (the global skill graph vs. clustered graph), and the ISE algorithm. We use *mean average precision* (MAP). as the evaluation metric. The results shown in Figures 2 and 3 indicate that ISE outperforms GSE by a large margin. We also see that in most cases MAP does not increases significantly with the number of the posts of a user, indicating that a few posts suffice for making a good prediction.

User Study. We perform a user study on nine users, most of whom have a computer science background. We ask the users to specify three peers they communicate frequently.

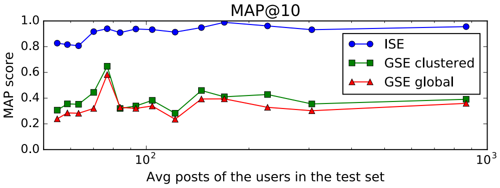

Figure 2: MAP@10 for different Algorithms

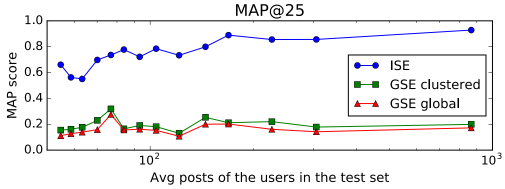

Figure 3: MAP@25 for different Algorithms

Table 1: Results from the user study on nine volunteers.

Algorithm	MAP@10	MAP@25	Average rating
ISE	0.617	0.557	3.74
GSE(PPR)	0.201	0.200	1.52
GSE(NPPR)	0.178	0.196	1.88
GSE(EI)	0.194	0.173	1.75
GSE(Harmonic)	0.238	0.198	1.71

Then, the users run our system on their email to infer skill endorsements are inferred for these peers. For each algorithm, we show a ranked list of the top 25 skills of each peer, and ask the users to indicate whether each skill is relevant. We also ask users to assess the list as a whole on a 5-point Likert scale (with 5 being more relevant). In the survey, we define a relevant skill for a peer as: *"the peer has done some work in the area, or could provide advice about it."* We compare ISE to 4 variants of GSE (the clustered version, which performs slightly better than the global one), namely, PPR, NPPR, EI, and Harmonic. For comparison we use the same MAP metric, as well as the average rating on the Likert scale.

The results, shown in Table 1, indicate again that ISE outperforms all GSE variants by a wide margin, irrespective of the centrality measure used.

Use cases. Next we apply our best-performing algorithm, ISE, on different publicly-available datasets. First, we use the Apache Spark mailing lists and we infer the skills of the most frequent users. In this case we treat the mailing list address as the user, and the people posting on the mailing list as the peers. The results are shown in Table 2. We see that one of the top skills for two of the users is 'Apache Spark,' as expected. Judging from the inferred skills, it is not surprising that Joseph Bradley is a developer of the MLlib project at Databricks.[4] Note that 'git@git.apache.org' is the email id of the bot named CodingCat which reports on merging of the code and compilation of the project.

Finally, we apply our algorithm on Hillary Clinton's emails. Her peers' skills are summarized in Table 3. Since Hillary Clinton was the Secretary of State at the time of the email collection, her discussions usually include foreign relationships, and hence different countries are reported as skills.

Table 2: Skills for users of the Apache Spark mailing list.

Email	Skills
sowen@cloudera.com	apache spark, hadoop, java, scala, spark
joseph@databricks.com	machine learning, apache spark, gradient descent, flex, random forest
git@git.apache.org	github, git, jira, infrastructure, bugzilla

Table 3: Skills inferred for Hillary Clinton.

israel, palestine, united states, nuclear weapons, politics

7. CONCLUSIONS

We proposed a novel approach for inferring an individual's skills by analyzing their personal communication data. Our approach makes use of a public knowledge base from the StackExchange Q&A forum. We found that ISE, an index-based approach, outperforms all version of GSE, a graph-based algorithm, when tested on the users of StackOverflow. The results from the user study confirmed the superiority of ISE, as its rating on a 5-points Likert scale is close to 4. We also collected open-ended feedback from the users taking the survey. We found that some of the top skills extracted by our algorithms were soft skills such as *group meetings*, *phd advisor*, and *grading*. However, the users were expecting, and suggested, hard skills such as *graph mining* or *machine learning*. This result might be a bias caused by the background of the users, and deserves further investigation. Finally we note that results on the enron email dataset were not as good. This can be attributed to the fact that stackexchange knowledge base is not well defined for all the available professions. To overcome this problem, one can use a more appropriate user-generated content for building the skill knowledge base.

Acknowledgements. This work has been supported by the Finnish Funding Agency for Innovation (project "Re:Know"), and by EU H2020 Program (project "SoBigData").

[4] http://www.cs.cmu.edu/~jkbradle